
The University Of Sheffield.

# Developing Methods and Resources for Automated Processing of the African Language Igbo

*Author:*
Ikechukwu E. Onyenwe

*Supervisor:*
Dr. Mark R. Hepple

*A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy*

Department of Computer Science
Faculty of Engineering

April 2017

# Declaration

I hereby declare that I am the sole author of this thesis. That except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. The contents are the outcome of research done under my supervisor. Part of this thesis has appeared in the following publications:

1. Onyenwe, Ikechukwu E., Chinedu Uchechukwu, and Mark Hepple. **Part-of-speech Tagset and Corpus Development for Igbo, an African**. In Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop, pages 93–98, Dublin, Ireland, August 23-24 2014. 2015 Association for Computational Linguistics.

2. Onyenwe, Ikechukwu, Mark Hepple, Chinedu Uchechukwu, and Ignatius Ezeani. **Use of Transformation-Based Learning in Annotation Pipeline of Igbo, an African Language**. In Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, pages 24–33, Hissar, Bulgaria, September 10, 2015. 2015 Association for Computational Linguistics.

3. Onyenwe, Ikechukwu E. and Mark Hepple. **Predicting Morphologically-Complex Unknown Words in Igbo**. In Proceedings of the Nineteenth International Conference on Text, Speech, Dialogue — TSD 2016, Brno, Czech Republic, September 12-16, 2016. Published by Springer-Verlag in Lecture Notes in Artificial Intelligence (LNAI), Volume 9924[1].

4. Onyenwe, Ikechukwu E. and Mark Hepple. **Predicting Morphologically-Complex Unknown Words in Igbo**. In Proceedings of the Community-based Building of Language Resources (CBBLR) workshop — TSD 2016, Brno, Czech Republic, September 12, 2016[2].

5. Onyenwe, Ikechukwu, Mark Hepple and Chinedu Uchechukwu. **Improving Accuracy of Igbo Corpus Annotation Using Morphological Reconstruction and Transformation-Based Learning**. JEP-TALN-RECITAL 2016, Workshop TALAf 2016: Traitement Automatique des Langues Africaines (TALAf 2016: African Language Processing), July, 2016, Paris, France, publisher: ATALA/AFCP, pages 1-10.

[1] The best papers which succeeded in both review processes (by the TSD 2016 Conference PC and CBBLR Workshop 2016 PC) will be published in the TSD 2016 Springer and CBBLR Proceedings

[2] See footnote 1

6. Onyenwe, Ikechukwu, Mark Hepple, and Ignatius Ezeani. **Towards An Effective Igbo Part-of-Speech Tagger**. Manuscript submitted for journal publication.

7. Onyenwe, Ikechukwu E and Hepple, Mark and Uchechukwu, Chinedu and Ignatius Ezeani. **A Basic Language Resource Kit Implementation for Igbo*NLP* Project**. Manuscript submitted for journal publication.

The above jointly authored publications are primarily the work of the first author. The role of the co-authors was editorial and supervisory.



Ikechukwu E. Onyenwe
April 2017

# Dedication

**To my family:**
my lovely wife *Obiọma* and my son *Chimdịndụ*.

**To these great ones:**
**Onyeanusi Ikechukwu and Gladys N. Onyenwe** *for being my wonderful parents.*
**Dr. Mark Hepple** *for being extremely good mentor and supervisor to me.*
**Prof. Boniface C.E. Egboka** *for believing in me.*
**Rev. Canon Prof. A.D. Nkamnebe** *for overwhelming guidance and supports to me.*

**In loving memory of my beloved sister *Lily*. Forever in my heart, my beautiful sister with a beautiful heart.**

# Acknowledgements

Firstly, I would like to express my exceptional appreciation and sincere thanks to my supervisor *Dr. Mark Hepple* (a Reader in Computer Science), who has been an immense mentor to me. I would like to thank him for supporting my research and enabling me to grow as a research scientist. His patience, motivation, immense knowledge and guidance on both research as well as on my career have been invaluable. His guidance and hard questions helped me to broaden my research from various standpoints in all the time of study and writing of this PHD thesis. I could not have envisaged having a greater supervisor and mentor for my PHD study. He is the true definition of a mentor.

I would like to thank the rest of my panel committee members, my chair, *Dr. John Barker* and my advisor, *Dr. Mark Stevenson*, for their insightful comments and suggestions. My sincere thanks also go to *Dr. Uchechukwu Chinedu*, a senior linguist who provided me with some Igbo linguistic materials and ideas. His collaboration throughout this study was very helpful. The administrative teams (com-$X$ groups) of Computer Science, University of Sheffield, are delightfully appreciated for their administrative supports throughout this PHD study.

Many thanks to Nnamdi Azikiwe University and Tertiary Eduction Trust Fund (TET-Fund) Nigeria for the funding. Special thanks to the Vice Chancellor, *Prof. Joe Ahaneku*, and his management teams for their supports.

I sincerely appreciate all who have positively impacted my life, especially these great ones: *Mr. Robin and Mrs. Joan Story, The Rt. Rev Prof. 'kelue Okoye, HRH Sir Dr. Harry Obi-Nwosu, Prof. S.O. Anigbogu, and Pastor (Dr.) and Mrs. Sam Okerenta.*

I thank my fellow colleagues at Igbo*NLP* project and Natural Language Processing Group of the Computer Science Department of Faculty of Engineering, The University of Sheffield, United Kingdom; for the sleepless nights we worked together meeting deadlines for panel meetings, research retreats, conferences , etc., and for all of the fun we had in the last four years. Special thanks to *Mark Tice*, *Ignatius Ezeani*, *Olusayo Obajemu*, *Samuel Nwagbo*, and *Joshua Gbenga Adeyemi* for their friendship and support beyond earthly norm. I am also thanking my wonderful colleagues and friends within and outside Nnamdi Azikiwe University in Nigeria, and those outside Nigeria for their calls and messages. You guys are one of the major reasons while I smile.

A special thanks to my family; words cannot express how grateful I am to my wife (seed of beauty) and son, parents, in-laws, and siblings for all of their prayers on my behalf.

To God be all the glory great things He has done. Amen!

**Ikechukwu E. Onyenwe, April 2017**.

# Developing Methods and Resources for Automated Processing of the African Language Igbo

**Ikechukwu E. Onyenwe**

## Abstract

Natural Language Processing (NLP) research is still in its infancy in Africa. Most of languages in Africa have few or zero NLP resources available, of which Igbo is among those at zero state. In this study, we develop NLP resources to support NLP-based research in the Igbo language. The springboard is the development of a new part-of-speech (POS) tagset for Igbo (IgbTS) based on a slight adaptation of the EAGLES guideline as a result of language internal features not recognized in EAGLES. The tagset consists of three granularities: fine-grain (85 tags), medium-grain (70 tags) and coarse-grain (15 tags). The medium-grained tagset is to strike a balance between the other two grains for practical purpose. Following this is the preprocessing of Igbo electronic texts through normalization and tokenization processes. The tokenizer is developed in this study using the tagset definition of a word token and the outcome is an Igbo corpus (IgbC) of about one million tokens.

This IgbTS was applied to a part of the IgbC to produce the first Igbo tagged corpus (IgbTC). To investigate the effectiveness, validity and reproducibility of the IgbTS, an inter-annotation agreement (IAA) exercise was undertaken, which led to the revision of the IgbTS where necessary. A novel automatic method was developed to bootstrap a manual annotation process through exploitation of the by-products of this IAA exercise, to improve IgbTC. To further improve the quality of the IgbTC, a committee of taggers approach was adopted to propose erroneous instances on IgbTC for correction. A novel automatic method that uses knowledge of affixes to flag and correct all morphologically-inflected words in the IgbTC whose tags violate their status as not being morphologically-inflected was also developed and used.

Experiments towards the development of an automatic POS tagging system for Igbo using IgbTC show good accuracy scores comparable to other languages that these taggers have been tested on, such as English. Accuracy on the words previously unseen during the taggers' training (also called unknown words) is considerably low, and much lower on the unknown words that are morphologically-complex, which indicates difficulty in handling morphologically-complex words in Igbo. This was improved by adopting a morphological reconstruction method (a linguistically-informed segmentation into stems and affixes) that reformatted these morphologically-complex words into patterns learnable by machines. This enables taggers to use the knowledge of stems and associated affixes of these morphologically-complex words during the tagging process to predict their appropriate tags. Interestingly, this method outperforms other methods that existing taggers use in handling unknown words, and achieves an impressive increase for the accuracy of the morphologically-inflected unknown words and overall unknown words.

These developments are the first NLP toolkit for the Igbo language and a step towards achieving the objective of Basic Language Resources Kits (BLARK) for the language. This Igbo*NLP* toolkit will be made available for the NLP community and should encourage further research and development for the language.

# Contents

# List of Figures

# List of Tables

# Part I

# Computational Linguistic Background

# Chapter 1

# Introduction

## 1.1 Introduction

The future prediction of main communication method between humans and computers will likely be in natural language, which is evident from recent advances in Natural Language Processing. Interestingly, some of the innate communication behaviour of humans is being exhibited by computers, for instance a good percentage of daily communications today is between humans and computers.

Today computers reciprocate human communication, we no longer interact with fellow humans to pay-in or withdraw money, that is now done through an automatic teller machine (ATM); we can interact with smart phone assistants to set an alarm and organize a calender, or find and connect with friends, or search for items on the Internet. We may even communicate with fellow humans in other languages through a "speak and translate" system as a mediator (e.g. Apple iTune free live voice and text translator that speaks 42 languages and hold written conversations in 100 languages (Apps, 2015)).

These are possible through the availability of language technologies today. It is foreseeable that, in the future, computers might exactly match human performance in face-to-face communication or close to human-to-human. Language technology is a multidisciplinary field comprising linguistics, psychology, engineering, and computer science. It is predominantly divided into speech technology and natural language processing (NLP). While speech technology looks at data in spoken form, NLP automatically processes written or textual data in natural languages.

In communication, there are two main coordinated processes, viz; putting ideas into words and extracting the ideas from words. These two processes rely on context to identify the possible meanings of ambiguous words in order to form the correct message. These are major challenges facing computers in communication. NLP is a field of computer science, artificial intelligence (AI), and computational linguistics which studies the interaction between computers and natural language. Processing human natural language is not a simple task as it involves extracting meanings of words which could be ambiguous. Word ambiguity arises when a word has more than one meaning and disambiguating this word requires understanding the word based on the context it is used. For instance, the word "fly" could be a verb (*-fe-* in Igbo) or a noun (*ijiji (winged insect)* in Igbo); in the sentence "time flies like an arrow", we can easily pick up "flies" to be a verb with our common sense. We can also observe the meaning of the word "flies" epiphenomenologically by

looking at the surrounding words. The task of disambiguation can be simple or complex depending on the ambiguity level of a language. NLP tools developed to solve the word ambiguity task for a language can go further in use for developing a machine translator, parser, chunker, word sense disamguator, and other tools to aid in human↔computer communication for that language.

Therefore, developing NLP resource tools for languages is a necessity in this age. NLP research has favoured a number of European languages. There are approximately 6000 languages in the world, but only a handful possess the NLP resources required for developing NLP tools (Bigi, 2011). The collection of corpus datasets for these languages (mostly under-resourced languages) is usually done by NLP researchers from scratch, with or without methods to bootstrap a manual process. The size of generated data may not be as large as European texts, which might affect the performance of NLP systems developed, compared to the technologically favoured European languages (Tachbelie et al., 2011). As a start, we develop NLP tools for an African language, Igbo, focusing mainly on developing a corpus, tagset, part-of-speech (POS) tagged corpus, automatic POS tagger, and morphological component for predicting the morphologically-inflected words and their segmentation. These NLP tools will form the basis of a basic language resource kit (BLARK), which is concept originally introduced in European Language Research Association (ELRA) newsletter in 1998 (Krauwer, 2003).

## 1.2 Motivation

Language barriers are being broken down with language technology systems, at most African languages are under-resourced and have not featured in this line of research due to a lack of NLP resources. It is likely that if nothing is done these languages will go into extinction and speakers excluded from communication in the world using their languages. Africa is the world's second largest and second most densely populated continent with approximately one billion people (Kaneda and Bietsch, 2016). Africa's languages form about 30% of the world languages and native speakers form 13% of the world population (Lewis et al., 2015), and yet it is a NLP tools dark continent. For example, on the Language Resources and Evaluation (LRE) Map (a freely accessible large database on resources for NLP) English has 663 corpora/computational tools, showing that it is the most studied language, followed by French and German languages, then Italian and Spanish. All of these languages are European, but almost no African languages appear (CorpusLinguistics, 2016; Calzolari et al., 2011). Krauwer (2003) asks of the fate of these lesser privileged languages and the place of their speakers as the global information society is gradually enlarged; he then paints two different problems and proffers a solution. The Problems are: (1) a few big languages end up overshadowing the globe, so smaller languages will gradually fade; (2) a few big languages end up overshadowing the place, and even though the smaller languages are kept, their speakers will be marginalized. The proffered solution is that language and speech technology will be used to guarantee involvement of all Europeans in the European expanse on an equal basis, irrespective of their language. This is the solution he adopted due to being a native speaker of one of the smaller languages in Europe.

Apart from exclusion, NLP research in African languages is an important aspect of

research for the following reasons:

- African languages are linguistically rich in features, and these features should be made known to the wider world in NLP for research purposes. Some of these features are described in Hyman (2003). For example, one of the Nigerian languages, Leggbo, a minority language spoken by about 60000 people, has subject–verb–object word order for affirmative sentences and subject–object–verb for negative sentences. Zulu, a language in the Bantu family, has an interesting agglutinative and conjunctive writing system; for example, the Zulu sentence *Okungumthakathi kuyangikhwifa* has two word forms, but is six separate words in English: 'The damn witch is bewitching me' (CorpusLinguistics, 2016). Also, in Igbo language spoken in Nigeria, 'must eat completely' (3 words in English) is agglutinatively written *richarịrị* where *–ri* is verb root, and *–cha* and *–rịrị* are suffixes indicating completion and compulsion respectively.

- The economic importance of studying African language as highlighted by Economist (2011) are; there are about 600 million mobile users in the African continent (more than Europe and America). One of the world's largest markets is located in the eastern part of Nigeria, where about 3 million people both local and foreign go on a daily basis to buy or invest in goods such as food materials, IT, and construction equipment.

- The increase of vernaculars on the internet is another important aspect of the study since a good percentage of these vernaculars are from African languages (Childs, 2005; De Pauw and De Schryver, 2009). For instance, there are 330,965,359 Internet users in Africa as of Nov 30 2015 (a 28.6% penetration rate) and 124,568,500 Facebook subscribers as of Nov 15 2015 (a 10.8% penetration rate). Nigeria alone has 92,699,924 Internet users as of Jun 30 2015 (51.1% of the population) and 15,000,000 Facebook users as of Nov 15 2015 (8.3% penetration rate) (InternetWorldStats, 2015).

- NLP research in the area of low-resource languages is worthwhile as not only will it provide NLP tools for the language, it will give insight on linguistic phenomena that are not found in already resource-rich languages. The NLP study results found could spur NLP researchers to further work, and native speakers of the language to participate in NLP research.

Inspired by the outlines above, the Igbo language was adopted because I am a native speaker of the language and an NLP researcher, and only a handful of linguistic literatures and texts are available for this language. This is to give Igbo people the potential benefits of NLP technology for computer use and information access, contributing to their communication within the global information society. Igbo native speakers is about 32 million (Factbook, 2016) and marginalizing this population from communication in the global village is a serious problem worthy of a solution. Recently, the UNESCO advisory committee on language pluralism and multi-language education predicted that Igbo language may be heading for extinction by 2025 if nothing is done by its speakers (Ani, 2012). This raises another serious problem worth considering: imagine about 32 million people without a language and culture (language defines people's culture). The

Igbo language, as of today, has received no attention in the area of NLP due to lack of research resources. Developing NLP resources for the Igbo language will help to reveal its linguistic richness and bootstrap its usage in this information technology age.

The NLP results achieved will be available for other African languages to follow suit. NLP in Africa is still in its infancy; of about 2000 languages, a very few have featured in NLP research and resources, which are not easily found online. Igbo language is one of the African languages with zero available NLP tools as of May 2013 (when we started this research).

## 1.3 Aims and Objectives

This research aims to develop functional NLP tools for the Igbo language. The objectives are to build:

- **A sizeable Igbo corpus (IgbC)**. Homogeneously collect Igbo electronic texts to produce a good corpus free from giving wrong word-type statistics, since Igbo has 30 dialectal variations.
- **A suitable tagset for part-of-speech (POS) annotation of the IgbC**. Design and develop a tagset that will capture the key linguistic distinctions in this IgbC.
- **A suitable tokenizer**. Design and develop a tokenizer in line with the tagset design.
- **An annotated corpus for the Igbo Language (IgbTC)**. Develop a POS tagged corpus for Igbo using this tagset and analyse the outcome using inter-annotation agreement (IAA).
- **A method for monolingual bootstrapping of the annotation process**. Develop an automatic approach that will project tagset changes made in the IAA exercise onto the IgbTC, instead of using human annotators to tag the corpus from new.
- **An automatic POS tagger that is capable of delivering good accuracy**. When an automatic POS tagger is applied on the IgbTC developed from this tagset, it should deliver a good accuracy comparable to that of tagger for other languages.
- **Handling morphologically-complex words**. Use of the knowledge of stems and associated affixes to predict appropriate tags for morphologically-inflected unknown words can improve the accuracy of unknown words that are morphologically-complex and other unknown words.

As a consequence of the stated objectives, the resources produced will be made available via the web to the NLP research community and wider society for further research and use. The Igbo language appears to have no free publicly available NLP resource tools. This research is, to the best of our knowledge, the first publicly available NLP methods/resources for Igbo.

## 1.4 Thesis Contributions

The main contributions of this thesis are as follows:

1. This study gave rise to first Igbo*NLP* tool kit, which is a step towards achieving BLARK goals for the Igbo language. The contents of this kit are:
   (a) An Igbo corpus (IgbC) of about one million words comprising two genres– the Bible[1] to represent the religious texts genre (RTG) and a novel[2] to represent the modern texts genre (MTG).
   (b) An Igbo tagsets (IgbTS) – three types of tagsets were developed: fine-grain (85 tags), medium-grain (70 tags) and coarse-grain (15 tags). Fine and medium sized grain are collapsible to coarse grain, and can be mapped to other language tagsets following the Eagles guide. The medium grained tagset captures inflected and non-inflected tokens in IgbC.
   (c) A sentence and word tokenizer.
   (d) A POS-tagged Igbo corpus (IgbTC) comprising 263856 words, part of the Bible is referred to as Igbo Tagged New Testament Bible Texts (IgbTNT) and 39960 words, the entire novel is referred to as Igbo Tagged Modern Texts (IgbTMT).
   (e) A morphological parser based on a morphological reconstruction method.
   (f) An automatic part-of-speech (POS) tagger developed based on IgbTC.
2. A monolingual-based manual annotation bootstrapping method through the exploitation of changes in the by-products of the inter-annotation agreement exercise and tag-error correction method that uses affix information to track and correct all morphologically-inflected words that are improperly tagged (their tags indicate that they are not morphologically-inflected). By using these methods, IgbTC achieved a considerably size achieved and was improved greatly in quality within the required time frame.
3. Word features suitable for prediction in Igbo: use of linguistically-informed stem and associated affixes of a morphologically-inflected word in Igbo is a better word feature for prediction than using the last (sometimes first) letters of a word that would normally serve as a proxy for actual linguistic affixes.

## 1.5 Remaining Chapters Outline

The remaining chapters are as follows:

- Chapter 2 describes the linguistic features of the Igbo language. This will reveal to the reader both the simple and complex linguistic patterns of the language.
- Chapter 3 explains tagsets and corpora, reviews the tagsets and corpora of other languages for the purpose of comparison between the Igbo tagset and tagsets for other language.

[1] obtained from jw.org
[2] obtained from its author and written in 2013

- Chapter 4 discusses NLP based on the POS tagging system– the techniques, machine learning methods to tagging, evaluation metrics and performance of POS taggers developed with these techniques.
- Chapter 5 discusses the corpus development, including the challenges and solutions. It also presents corpus data preparation methods and statistics.
- Chapter 6 discusses design and development of the Igbo tagset, method of tagset revision, the initial encoding of linguistic tags into the Igbo corpus, and the platform used.
- Chapter 7 discusses how we used the by-products of the inter-annotation agreement exercise to improve the result achieved in chapter 6.
- Chapter 8 details the POS-taggers used for tagging Igbo corpus data and the evaluation of results.
- Chapter 9 is a discussion on automatic tag-error correction and handling of unknown words in Igbo using morphological characteristics features of the language.
- Chapter 9 discusses future directions.

# Chapter 2

# Linguistic Background

Igbo, one of the most spoken languages in West Africa is a Kwa sub-group language of the Niger-Congo family (Widjaja, 2013). Igbo is the native language for a subset of Nigerians called Igbo who live in the eastern part of the country. Nigeria has three majority languages with millions of speakers collectively known by the word *wazobịa*; that is, 'wa' from Yoruba, 'zo' from Hausa, and 'bịa' from Igbo, each meaning 'to come' (Widjaja, 2013). The Igbo region forms roughly 18% of Nigeria, and there are about 32 million Igbo language speakers (Factbook, 2016). It is worth noting that Igbo people are mostly bilingual (as are most Nigerian), also speaking English as Nigeria is a multilingual country with around 510 different languages, so English serves as the official language.

## 2.1 The Igbo Language

Igbo features tones and vowel harmony characteristics (Widjaja, 2013) and, like many other languages, has multiple dialects; there are about thirty, each with different contrastive pitch. Dialectal variation is mostly lexical, phonological, and syntactic structures (Emenanjo, 1978; UCLA, 2014). The standard dialect is based on the Owerri and Umuahia dialects, the capital cities of the two eastern states, Imo and Abia (UCLA, 2014). The standard dialect claim is based on historical and literary reasons, though Emenanjo (1978) argued that dialectal variation is not based on any region, rather it is a function of selecting what appears best from various styles or ideas. This study will focus on standard Igbo, hereafter abbreviated as SI.

The first written Igbo words and phrases were found in the book of a German missionary, G.C.A. Oldendorp, *Geschichte der Mission der Evangelischen Bruder auf der Carabischen* "History of the Mission of the Evangelical Brothers in the Caribbean", published in 1777 (Pritchett, 2014; Omniglot, 2016). Following this, 79 words were found in "The Interesting Narrative of the Life of Olaudah Equiano" published 1789 in London, England by a former slave. In 1841 a Norris expedition on the Niger took two missionary linguists from the CMS (Church Missionary Society) staff in Freetown, namely, J. F. Schon and Samuel Ajayi Crowther, together with twelve interpreters (including Igbo), who came from liberated slave families settled in Freetown. Schon was interested in Igbo and Hausa, and tried to communicate with Igbo people in their own language but was disappointed that people did not understand him, probably because of his accent. He then left the Igbo language study for twenty years (Pritchett, 2014). The first Igbo textbook, *Isoama-Ibo*

*A Primer*, was written by Samuel Ajayi Crowther in 1857 (Pritchett, 2014; Omniglot, 2016) an ex-slave, African Linguist, and the first African Anglican Bishop. In 1861, J. F. Schon seemingly resumed his Igbo studies, publishing *Oku Ibo: Grammatical Elements of the Ibo Language*, the first Igbo grammar, written in the Isuama dialect using Lepsius orthography (Pritchett, 2014).

### 2.1.1 Writing System

The Igbo orthography was filled with controversies between the Roman Catholic Church and the Church Missionary Society (CMS) with their two competing orthography systems, the New and Lepsius orthographies, respectively (Oraka, 1983; Uchechukwu, 2008). This lasted for a period of about 30 years (1929–1961), and was finally resolved in 1961 by the Ọnwụ Committee (Ọnwụ Committee, 1961). The official orthography, known as Ọnwụ was adopted and standardized by the Ọnwụ Committee (1961), which uses the Latin script (see table 2.1). Igbo in the 1500s before Ọnwụ Committee had a writing system called *Nsibidi*, based on ideograms as used by some secret cults (Ekpe and Okonko) for secret communications. Representation in electronic text requires the use of unicode, mainly because of the use of diacritics on characters. The use of diacritical marks is to distinguish between "light" and "heavy" vowels (Oraka, 1983).

| Letter | A | B | Ch | D | E | F | G | Gb |
|---|---|---|---|---|---|---|---|---|
| Pronounciation | /a/ | /b/ | /ʧ/ | /d/ | /e/ | /f/ | /g/ | /g͡b/ |
| Letter | Gh | Gw | H | I | Ị | J | K | Kp |
| Pronounciation | /ɣ/ | /gʷ | /ɦ/ | /i/ | /ị/ | /dʒ/ | /k/ | /k͡p/ |
| Letter | Kw | L | M | N | Nw | Ny | Ṅ | O |
| Pronounciation | /kʷ/ | /l/ | /m/ | /n/ | /ŋʷ/ | /ɲ/ | /ŋ/ | /o/ |
| Letter | Ọ | P | R | S | Sh | T | U | Ụ |
| Pronounciation | /ɔ̣/ | /p/ | /ɹ/ | /s/ | /ʃ/ | /t/ | /u/ | /ʊ̣/ |
| Letter | V | W | Y | Z | | | | |
| Pronounciation | /v/ | /w/ | /j/ | /z/ | | | | |

Table 2.1: The standard orthographic graphemes for Igbo

The Ọnwụ standard orthography of Igbo is made up of 36 graphemes (see table 2.1). There are 28 consonants: *b gb ch d f g gh gw h j k kw kp l m n nw ny ṅ p r s sh t v w y z*, and 8 vowels divided into two harmony groups based on Advanced Tongue Root (ATR) (see table 2.2), nine of the consonants are digraphs: *ch, gb, gh, gw, kp, kw, nw, ny, sh* (Ọnwụ Committee, 1961; Agbo, 2013; Uchechukwu, 2008). The consonants *sh* and *v* are not frequently used in word formation. The vowels of the two harmony groups are combined according to vowel harmony to form Igbo words (Ọnwụ Committee, 1961; Emenanjo, 1978). Vowel harmony is a phenomenon in some languages, such as Igbo, for all of the vowels found in a word to be constituents of the same group. As examples in the figure 2.1, -ATR in table 2.2 will have *aka* 'hand', *akwụkwọ* 'book', *ọkụ* 'fire', and *ụwa*

'world' will have -ATR; and *eke* 'market', *okwu* 'word', *mmiri*, 'water' and *egbe* 'hawk' have +ATR. Also see Igbo grammar in (IgboGuide.org, 2016).

| | | | | |
|---|---|---|---|---|
| -ATR | ị [ɪ] | ụ [ʊ] | a [ɑ] | ọ [ɔ] |
| +ATR | i [i] | u [u] | e [e] | o [o] |

Table 2.2: Vowel Harmony Groups (Uchechukwu, 2008)

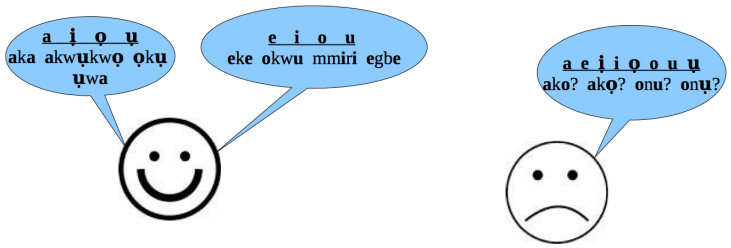



Figure 2.1: Igbo Vowel Harmony

### 2.1.2 Phonology

The majority of the words in the language take their vowels from one harmony group in table 2.2; additionally, the language is tonal. Its qualities as a tonal language were used by Goldsmith (1979) in the development of his theory of Autosegmental Phonology. Three distinct tones are recognized in the language: High, Low, and Downstep (which occurs mainly after a high tone). The tones are represented as *High* [H] = [ ´ ], *Low* [L] = [ ` ], *downstep* = [ ˉ ] (Emenanjo, 1978; Ikekeonwu, 1999) and are placed above the tone bearing units (TBU) of the language. There are two tone marking systems used for written Igbo. In one system only contrastive tones are marked (Welmers and Welmers, 1968; Nwachukwu, 1987), while in the second system all high tones are left unmarked and all low tones and downsteps are marked (Williamson, 1971; Emenanjo, 1978). The second system will be used here for illustration of the tonal feature of the language, which could be lexical or grammatical. For example, at the lexical level, the word *akwa* without tone marks could be equivalent to 'bed/bridge', 'cry', 'cloth', or 'egg'. But these equivalents can be properly distinguished when tone marked, as follows: akwa "cry", akwà "cloth", àkwà "bed or brigde", àkwa "egg". At the grammatical level, an interrogative sentence can be distinguished from a simple declarative sentence through a change in tone. For example, sentence (1) could be changed into the interrogative sentence (2) through a change in the tone of the third person pronoun, from a high to a low tone. Such tonal changes play a role in the grammar of the language.

(1) Ọ nà-àbịa
he AUX-come
'He is coming.'

(2) Ọ̀ nà-àbịa
he AUX-come
'Is he coming?'

In addition, there are syllabic nasal consonants which are also tone bearing units, and always occur before a consonant. For example: ǹdo 'Sorry', which can be explicitly tone

| Lexical tone | Word 1<br><br>Distinguishing affirmative sentence from a question | Word 2 | Word 3 | Word 4 | Word 5 |
|---|---|---|---|---|---|
| HH | *o/ọ/m* 'she/he/I' | *ha* 'they' | *isi* 'head' | *akwa* 'cry' | *oke* 'male' |
| HL | – | – | *isì* 'smell' | *akwà* 'cloth' | *okè* 'boundary' |
| LL | *ò/ò/m̀* | *hà* 'verb root'<br>*hà* 'question' | *ìsì* 'blindness' | *àkwà* 'bed' | *òkè* 'share' |
| HL | – | – | *ìsi* 'to cook' | *àkwa* 'egg' | *òke* 'rat' |

Table 2.3: How lexical tones affect the grammatical meaning of Igbo words

marked as ǹdó. For more examples on the use of tones for grammatical distinctions, see table 2.3.

There are eight vowels, thirty consonants, and two tones (high and low) in phonemic analysis of Igbo. Emenanjo (1978) represent the structure in this way:

$$(C)S^T \tag{2.1}$$

This structure represents the Igbo Syllable, where C is a consonant, ( ) indicates optionality, T is Tone, and S is syllabic. There are two restrictions for vowels, syllabic nasals, and consonants in this structure:

1. S is always a vowel and never a syllabic nasal; and
2. there can be only one consonant in C position.

We can derive three instances from this as follows:

1. a tone-bearing unit is a syllable of one vowel;
2. syllables are preceded by consonants;
3. a syllable forms a whole or part of a word.

The Igbo syllabic is either a vowel or nasal. Nasals are the consonants *m, n and ṅ*, each carrying a different tone (as does each vowel in SI).

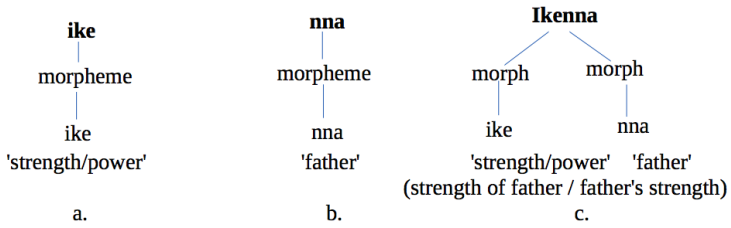
### 2.1.3 Morphological Structure

Morphology is the study of the internal makeup of words, this is done through identifying *morphemes*, often described as the smallest unit of a word with a grammatical function (Van Valin, 2001). Some languages use morphology to emphasize what other languages would stress syntactically as a lexical unit. Igbo is in this language class; its nouns and verbs are the two main grammatical classes that undergo affixation of morphemes to change or extend the grammatical meaning of the original word. The implication of this

is that morphemes are affixed either by prefixation or suffixation to form a complete morphological structure (Anderson and Petronella, 2006). This is the main causative factor of the agglutinative morphological structure of Igbo words, and is especially prevalent in verbs.

Igbo nouns and verbs are monosyllabic in their root form before morphological effects, that is, they have consonant-syllable (CS) structure. Emenanjo (1978) identifies only five monosyllabic nouns, di, chi, nwa, be, and ji. Nouns can start with either a vowel or a nasal syllabic; in each case, articulation should be harmonized with the root *i-ke, ọ-nwa, a-la*, and where it is a nasal syllabic, articulation takes place from the following consonant *m-ma, m-yọ, n-na, n-nọ* (Clark, 1990).

## Noun Morphology



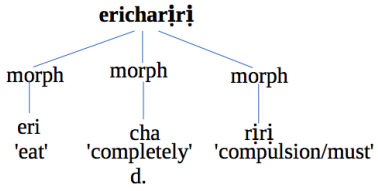




Figure 2.2: Demonstrating the use of morphemes in Igbo words

In the above examples, figure 2.2a and 2.2b show that *ike* and *nna* are two common nouns that can morphologically join to form the proper noun *Ikenna*, as shown in 2.2c. Reduplication in Igbo is seen in the nominalization process, where nouns are realized from verbs (Emenanjo, 1978). For instance, *ịgụ egwu* "to sing song" when nominalized will form *ọgụ egwu* "singer". This nominalization is a complex verb structure where the noun *egwu* "song" is complementing the verb, *-gụ*, with a prefix *i* to complete its sense. The verb *ịgụ* can be found in other structures like *ịgụ ọnụ* "to count" and *ọgụ ọnụ* "counter". The prefix marker indicates the infinitive class, which changes to "ọ" (a personal pronoun) to form a noun.

## Verb Morphology

The word form *erichaṛịṛị* contains four morphemes: a verbal vowel prefix, a verb root, and two suffixes as shown in figure 2.2d. See table 2.4.

Verbs are the only grammatical class that undergo inflection to depict tense (AYOGU et al., 2013) and aspect (UCLA, 2014) and this is achieved through morphological affixation. Ikegwuonu (2011) attests that use of $-rV$ in Igbo verbs more often indicates tense. Further

| | | |
|---|---|---|
| *Obi ga-erichariri nri ahụ* | Obi aux-eat.completely.must food DET | 'Obi must eat up that food.' |

Table 2.4: Illustrating figure 2.2d further with a sentence

| Original State | Segmented State | Prefix/Suffix/Enclitics |
|---|---|---|
| abaghị | a+ba+ghị | Prefix *a-* is a Verbal Vowel Prefix (VVP) that define a verb participle. There are two VVPs *a-, e-*. Suffix *-ghị* for negation. |
| abanye | a+ba+nye | Prefix *a-*, as above. Suffix *-nye* to denote penetration to interior |
| abanyeghị | a+ba+nye+ghị | Prefix *a-*, as above. Suffix *-ghị* negate suffix *-nye* |
| abanyekwala | a+ba+nye+kwa+la | Prefix *a-*, as above. Enclitics *-kwa* and two suffixes *-nye, -la* |
| abatabeghị | a+ba+ta+be+ghị | Prefix *a-*, as above. Suffixes *ta,be,ghị* to tell that who you've been looking for has not come back. |
| abịaghịkwa | a+bịa+ghị+kwa | Prefix *a-*, as above. Suffix and enclitics *-ghị, -kwa* to show 'did not come also'. |
| abịakwaghị | a+bịa+kwa+ghị | Prefix *a-*, as above. Suffix and enclitics *-ghị, -kwa* to show 'did not come also'. |
| abịakwa | a+bịa+kwa | Prefix *a-*, as above. Enclitics *-kwa* to show 'come also'. |
| abịakwasị | a+bịa+kwa+sị | Prefix *a-*, as above. Enclitics and suffix *-kwa,-sị* to show 'come also persistently'. |
| abịakwutebeghị | a+bịa+kwu+te+be+ghị | Prefix *a-*, as above. Enclitics *-kwu* and suffix *-te, -be, -ghị* to negate 'to begin to come towards ...in addition to something else'. |

Table 2.5: Illustrating enclitics and suffixes as found in Igbo verbs

to this, he states that the past tense marker $-rV$, which is a bound morpheme attached to the verb root, is morphologically obvious rather than the present tense, which is covertly marked. Verbs also undergo additional morphological change activated by sentence-type. Enclitics is a grammatical class of words in Igbo that stand on their own when appearing in a sentence, before or after a class that is not a verb (Emenanjo, 1978); they are suffixed to the verb if found immediately after it. Therefore, a verbal can be the formation of $AUX - P \pm V.R \pm S \pm E$, where $AUX$ is the auxiliaries *na* and *ga*, $P$ is a prefix, $V.R$ is a verb root, $S$ is a suffix (inflectional $INFL$ or extensional $EXT$) and $E$ is enclitics; though this is often disobeyed by Igbo writers. See table 2.5 for illustrations. Also see verbals in table 2.6 and Emenanjo (1978) for further details.

Observe the occurrence positions of *kwa* and *ghị* in table 2.5, this indicates that some Igbo morphemes can combine with stems in multiple different orders to form new words. For example, we can hypothesize that *kwa* does not always occur at a strict position of an Igbo word using the following examples *a+bịa+kwa+sị, a+bịa+kwa+ghị, a+ba+nye+kwa+la.*

### 2.1.4 Syntactic Structure

*"Syntax is the central component of human language"* (Van Valin, 2001). The order of words in a sentence is vital in semantic and context analysis. Word order defines how words are to be used in a sentence construction. The order of clauses in Igbo is Subject–Verb–Object (SVO)(UCLA, 2014), with a complement to the right of the head in all types of phrases.

Simple sentences in Igbo (examples from Nweke (2011) and Clark (1990)), *Okeke gburu agwọ* "Okeke killed a snake" and *mèchie anya* "Close (your) eyes" are expressed as follows: *Okeke* as the subject, *gburu* and *mèchie* as the verbs, and *agwọ* and *anya* as the objects. The verb has a separate constituent tense which is an inflection on the verb *gbu* to indicate past tense.

These examples show that at any level, the complements are always to the right of the head. Therefore, a simple Igbo sentence has $NP \pm Verbal \pm NP$ structure. Where NP (Noun Phrase), VP (Verb Phrase) and PP (Prepositional Phrase) have $\pm A + N \pm A \pm P \pm Nm \pm Q \pm D \pm RC$, $V \pm NP$, and $P \pm NP$ structures respectively. PP belongs to verb phrase though it can be found at beginning or end of a sentence having same meaning (Emenanjo, 1978). A is adjective, N is noun, P is pronominal modifier, Nm is numeral, Q is a quantifier, D is demonstrative, and RC is relative clause. Table 2.6 provides examples.

| NP | | | Verbal | | NP | | | Eng |
|---|---|---|---|---|---|---|---|---|
| | | | Verb | Stem | | | | |
| | NP | Enclitic | V.R. | Suffix | Enclitic | NP | Enclitic | |
| 1a | Ndị a | cha | bịa | –ra | | oriri | | All these people came to the feasting |
| 1b | Ndị a | | bịa | | –cha | oriri | | All these people came to the feasting |
| 1c | Ndị a | | bịa | –ra | | oriri | cha | All these people came to the feasting |
| 2a | Gịnị | | bụ | | –kwa | nke a | | What else is this? |
| 2b | Gịnị | –kwa | bụ | | | nke a | | What else is this? |
| 2c | Gịnị | | bụ | | | nke a | kwa | What else is this? |

Table 2.6: Relative positions of the verb root, suffix, and enclitic in Igbo NP from Emenanjo (1978)

Words of more than one lexical category are constituents of sentences. Traditionally, noun, verb, adposition, adjective, and adverb are the most common and important lexical classes. Igbo, as outlined in Emenanjo (1978), has its lexical classes condensed into six.

## 2.2 Conclusion

This chapter introduces the linguistic background of Igbo starting from the native speakers' region and population, the language family, and interesting features like its writing system, phonology, syntax, and morphology. Apart from morphological effect that causes multiple words in English to be agglutinatively written as one form in Igbo, whitespace is also used to denote lexical boundaries between both morphologically-inflected and non morphologically-inflected words. In Igbo, morphemes have different lengths and they can combine with stems in multiple different orders to form different variants of new words. It is important

to note that other classes, such as nouns, could also be affected by morphology. Therefore, it is a non-trivial task to find the boundaries between affixes in morphologically-inflected words.

# Chapter 3

# Part-of-Speech Tagset and Corpora

A tagset is a part-of-speech (POS) annotation scheme designed independently to suit a particular language. It is used for identifying the grammatical function of each word on a sentential level. A good tagset should capture the key grammatical features of a language, while maintaining an optimal size. Developing a tagset is an important first step for POS annotation tasks in both manual and automatic processes. Below we define tagset and corpora, state the importance of conducting this study, and look at corpus resources and the challenges of developing a POS corpus for under-resourced languages.

## 3.1 Tagset

The first step for part-of-speech (POS) annotation is a well designed, consistent, and complete tagset for the language (Bamba Dione et al., 2010). Studying and analysing the language in detail is an essential step towards developing a new tagset that will be used for POS tagging a corpus. A tagset is a set of *word categories* designed to be applied to the tokens of text (Leech, 1997), while tagset design is the *process of developing tagging guidelines* for identifying and labelling each token in the language domain under study with the appropriate grammatical class. A standard guideline, such as EAGLES, could be adopted for this process to avoid re-inventing the wheel.

In a classical POS tagging task, classification of tokens into one POS class is not achievable since some words cannot be cleanly classified into one class. The input consists of a tokenized corpus and well defined tagset, and the accuracy of a tagset is essential for delivering a good POS tagging. For high quality tagset design and annotations, a revision of the tagset is required, which is done through an inter-annotation agreement process (discussed in chapter 6). Further improvements of this tagging scheme (if need arise) is necessary until best accuracy is acquired since the majority of an automatic POS tagger's errors are due to wrong human judgements (Manning, 2011).

The grammatical classes and size of a tagset for a language are dependent on the purpose or target users. This is where Atwell (2008) advises tagset developers to be clear about the purpose of POS tagging a corpus. It might be for enriching a corpus with linguistic analysis in order to maximize the potential of the corpus' re-use in a wider range of applications, or a purpose specific for the user. For example, a corpus linguist may design a tagset that is very fine-grained with grammatical distinctions reflecting his expert interest in syntax and morphology, but such fined-grained distinctions may cause

problems for automatic POS tagging (Atwell, 2008). Therefore, in the tagset design, it is essential to be subtle and consider its size; according to De Pauwy et al. (2012) and Atwell (2008) the lesser or more fine-grained a tagset, the higher or lower the accuracy and less/more the ambiguity[1].

The main knowledge engineering required in tagset design and development can be localised in the choice of the tagset, which is subject to either external or internal criteria, or both. The external criteria focuses on the linguistic distinctions required in the output corpora. For example, the Penn treebank tagset (recently the most used tagset for English) omitted some distinctions used in the LOB (Lancaster-Oslo/Bergen) and Brown tagsets on which it is based. Thus, the auxiliary verbs "be, do, have" share the same tags as other verbs in the Penn treebank, but are separated in the LOB tagset (Elworthy, 1995). The internal criteria delve into finer grains of the language such as higher level syntactic or morphological analysis; for example, distinctions found in corpora such as Susanne which have tags indicating phrasal structure in addition to POS tags (Elworthy, 1995). For agglutinative languages, a tagset could be designed in such a way that POS tags are accompanied by a paradigm string, whose positions denote certain other grammatical aspects (Aibek et al., 2014).

## 3.2 Corpus

A corpus is a large systematic collection of electronic text in a language for linguistic analysis. It could also be a collection of written or spoken texts upon which a linguistic evaluation is based. Examples of existing corpora are Brown, Penn Treebank, Wall Street Journal (WSJ), Quranic Arabic Corpus (QAB), Icelandic Frequency Dictionary (IFD. Corpus development could be monolingual to represent a single language, or bilingual to represent two languages, or even multilingual to represent multiple languages. European Corpus Initiative (ECI) is a multilingual corpus, with 98 million words in Turkish, Japenese, Russian, Chinese, and other languages (Robin, 2009). A corpus can be created from written language, spoken language, or both; sources are mainly audio recordings for spoken language, and web texts, religious texts, educational texts, history texts, etc. for written language.

Apart from the useful information that a corpus provides to linguists, a tokenized corpus with POS meta-data provides the lexical, morphosyntactic, semantic, or pragmatic information needed for building an automatic POS tagger (an important basic tool needed for building advanced natural language processing tools (NLP), such as machine translation system, parser, etc.).

## 3.3 EAGLES Guide for Developing A Good Tagset and Corpus

EAGLES is a standard guideline for developing grammatical classes for any new language. The need for standardisation in the creation of a tagset is desirable due to the

[1]See figures 8.5 and 8.6 in chapter 8 for ambiguity rates and POS taggers' accuracy scores on Igbo corpus (IgbTC) using different granularities of Igbo tagsets (IgbTS)

interchangeability, validity, reproducibility, and re-usability of the tagset and annotated corpora it produces. Leech and Wilson (1999) advise that it is important to refrain from a "free-for-all" or "reinvention of the wheel" in new tagset design and development, and that the annotation scheme used for one language should, as far as possible, be reusable and compatible with others. When part-of-speech (POS) tags are added to a corpus, the resulting tagged corpus can be passed-on on to other users for study purposes; therefore, compatibility and re-usability require a certain level of standardisation for the purpose of enabling researchers to exchange data and resources. Re-usability in this sense means that a tagged corpus can be used for a purpose other than the original, so if standardisation were considered during development of this tagged corpus, it would greatly reduce the need for manual adaptation by new users. Compatibility means the ability to apply POS tags that are common across languages in the annotation scheme, eg. core tags, such as noun, verb, adjective, preposition, pronoun, adverb, etc.; and equally that the annotations applied to texts can easily be recovered from different languages. For example, there are some tags that are most useful for a POS tagging system across languages, irrespective of disagreements linked to the internal structure found in languages. This, among many other reasons, gave rise to the universal tagset of 12 tags, which has been successfully tested on 22 languages (Petrov et al., 2011).

A set of tagset features are outlined in EAGLES; the choice of how to apply these is entirely dependent on the use. At a morphosyntactic annotation level, EAGLES describes three structures: obligatory, recommended, and optional extensions for properties that are language-specific. In the *obligatory* category are thirteen core tags (noun, verbs, adverbs, adjective, article, etc.); the structure *recommended* and *optional* are dependent on these core tags. For example, the recommended attributes for noun in *obligatory* is of type common/proper, and degree positive/comparative/superlative for adverb. There are also number, gender, case, finiteness, tense, voice, and other miscellaneous sub-categorisation features. In the *optional* case, there are similar attributes as in *recommended*, and additional ones specific to small numbers of languages.

While EAGLES is a flexible framework that consists all basic attributes, it is mostly used as a springboard for starting tagset design. However, as a project of the European union, it only covers a fraction of the world's languages consisting of only European languages. For example, in the obligatory level, the 'article' attribute is not applicable to the Igbo language; 'the' and 'a' are not overtly used in Igbo sentences, eg. *Ọ gara ahịa* "S/he went to (the) market", it is left for communicators to know if it is "the market" (one mentioned previously) or "a market". Leech and Wilson (1999) identify around 20 users in Europe have used EAGLES guidelines for tagset design, so the guidelines on morphosyntactic annotations have been applied, tested, and evaluated in a number of national and European languages; examples discussed in Hardie (2003) are the MULTEXT, GRACE, and CRATER projects.

A corpus is much more useful when annotated, which is an enrichment of the original raw corpus. From this perspective, adding annotations (tags in the POS case) to a corpus is giving 'added value', which can be useful in many ways for research, either by the team that carried out the annotation, or others who find it useful for their own purposes. Geoffrey (2004) highlights nine important standards for corpus annotation, such as: annotations should be separable (easy to separate from the raw corpus), explicit and detailed documentation about the annotations should be provided; annotation should

be based on a 'consensual' set of categories on which people tend to agree, in order to fit with the re-usability goal for annotated corpora, there should be an annotation manual to explain the scheme to users; and evaluation of the annotations for consistency and accuracy.

## 3.4 The State of NLP in Africa

Despite the large number of languages and native speakers in Africa, it is still an NLP-dark continent; the LRE map[2] shows that while English has a good number of computational and corpora tools ( 663 reported in Corpus linguistics[3]), African languages have relatively few.

There has been a growing interest in NLP in Africa (Björn et al., 2009; Tachbelie et al., 2011; Bamba Dione et al., 2010; Trushkina, 2006). TALAf[4] (Traitement Automatique des Langues Africaines[5] (text and speech)) is a workshop (held at the JEP-TALN-RECITAL conference, 2016) with the aim of bringing together researchers in the NLP field working on African Indigenous Languages[6] (AIL) through: meetings at the workshop; extracting knowledge using open source tools, standards (ISO, Unicode), and publishing the tools developed with an open license to avoid losses when a project stops and cannot be reopened for lack of resources; developing a set of best practices based on the researchers' acquaintances; setting up simple and effective methodologies based on free, or almost free, software for the development of tools; communicating methods that can eschew the use of non-existent tools; and refraining from loss of time and energy. AFLAT[7] is an African Language Technology body interested in language technology research for AIL, aiming to catalogue resources (such as corpora, dictionaries, and NLP tools) for the majority of resource-scarce AIL (both current and extinct) for the benefit of researchers interested in African language technology.

AILs are linguistically rich and have high divergence in typology (Mariya, 2012), although some bear little relation to one another. The typological difference could be the effect of many ethnicities in Africa. There are four language classes in Africa: Afro-Asiatic, Nilo-Saharan, Niger-Congo, and Khoisan (Alejandro and Beatriz, 2013). The Niger-Congo family is a large language phylum that contains approximately 1500 of the languages of Africa (Gordon, 2005; Demuth et al., 1986). The Igbo language is among the 40-60 languages in the Kwa sub-group of the Niger-Congo family. Common shared features among the languages of Africa are in their phonology, syntax, morphology and lexicon (Alejandro and Beatriz, 2013), implying that NLP methods, experiments, and experiences obtained for one language could be extended to others.

There are major challenges facing under-resourced languages; most outstanding are the text to be tagged, orthography of the text, tokenization (dealing with morphology since affixations are prevalent features in most AILs), and the size of tagset. In POS

[2] `https://en.wikipedia.org/wiki/LRE_Map`[Accessed: 31/07/2016]

[3] `https://corplinguistics.wordpress.com/tag/swahili/`[Accessed: 31/07/2016]

[4] http://talaf.imag.fr/2016/

[5] Automatic processing of African languages.

[6] Indigenous languages that historically belong to the continent, rather than being brought from another country. Under this definition, languages like English and Arabic are not African.

[7] http://aflat.org/node/1

tagset development and tagging for the Niger-Congo family, the first and most common issue is in their morphology and orthography-languages in this family are morphologically agglutinative in structure (Bosch et al., 2008; Poulos and Louwrens, 1994); words are formed by concatenation of morphemes that would syntactically stand as a lexical unit in other non-AIL languages, and this is mostly the case for noun and verb classes. See tables 3.1 and 3.2.

POS tagset design and tagging in the above case is a non-trivial task. Firstly, which units should be classified as tokens, since words in these language types are highly inflected with morphemes, and tokenizing purely on whitespace would be linguistically misrepresenting. Additionally, it is difficult to define the boundaries of these morphemes in inflected words[8], and choosing the tagging types for each. It is also difficult to determine the best level of morphology decomposition, since these languages are so morphologically rich, and the order of occurrence of morphemes is not fixed (*abịaghịkwa* and *abịakwaghị* are valid words with the same sense in Igbo; also see table 2.5 of chapter 2). Analysing and determining the best tokenization algorithm for morphologically complex languages is a lengthy and laborious process- ZulMorph, the UNISA prototype morphological analyser for Zulu, took a decade to develop (Bosch et al., 2008).

Another major problem that is very common in AILs is multiword units. They may appear as separate tokens with the same tag[9], or combined tokens with different tags. For example, in Igbo noun classes, agentive nouns are formed through the nominalization of verbs (eg. *ọgụ egwu* 'singer'), and instrumental nouns are words used to refer to, or describe, instruments, which are also formed through the nominalisation of verbs (eg. *ngwu ji* 'digger'). In Wolof (Senegalese language), Bamba Dione et al. (2010) found that the pronominals or focus markers and associated inflection often appear as separated words in the Wolof text. A different case of the above is found in Northern Sotho, where a single word form will receive more than one tag (Taljard et al., 2008). Another prominent feature in AIL is the use of ideophones or words that evoke vivid sensations.

AIL are classified as under-resourced languages because of lacking the linguistic resources such as electronic texts, word lists, dictionaries, grammars, spell-checkers, etc. (De Pauw and De Schryver, 2009). Though some languages have one or more linguistics materials available that can help kick-off natural language processing (NLP) research. But lack of the processed linguistic items is the major cause of NLP stagnant growth in African language technology and reason can be attributed to the social and political past of Africa that did not promote the use of native languages in education and commerce, until recently (Mariya, 2012).

In AIL, a verb may comprise subject, concord, a verb stem (bears the basic meaning) and inflectional ending (Heid et al., 2006). Morphemes prefixed to the verb root may include lexical class such as object concords, potential and progressive, negative, and participle morphemes. There might be derivational or extensional suffixes appearing between a verb stem and inflectional ending as in table 3.2.

According to Heid et al. (2006), all verbal derivatives can be blindly tagged as verbs or can be morphologically analysed. If the latter, then tagging will be based on the verbal suffix's lexical functions. Morphological ambiguity is resolved using contextual information. For example, verbal derivation like that one in table 3.3 can be blindly tagged as verbs, or

[8]We refer to inflected words as words formed morphologically by either inflection or extensional suffixes.

[9]This means part of speech (POS) class or tag.

| Class | Northern Sotho | | Zulu | | Igbo | |
|---|---|---|---|---|---|---|
| | Plural marker | Example | Plural marker | Example | Plural marker | Example |
| sg | mo– | *motho* 'person' | *umu–* | *umuntu* 'person' | nwa | *nwa mmadụ* or *mmadụ* 'person' |
| pl | ba– | *batho* 'persons' | *aba–* | *abantu* 'persons' | *ụmụ* | *ụmụ mmadụ* 'persons' |

Table 3.1: Noun classes for Northen Sotho, Zulu and Igbo. Part of table taken from (Heid et al., 2006))

| | "they do not sell" | Negative morpheme | Subject | Verb root | Suffix | Prefix | Inflectional ending |
|---|---|---|---|---|---|---|---|
| Northern Sotho | ga ba rekiše | ga | ba | rek– | –iš– | – | –e |
| Zulu | abathengisi | a– | –ba– | –theng– | –is– | – | –i |
| Igbo | ha anaghị ere | –ghị | ha | –na | –ghị | a– | – |

Table 3.2: Verbal morphology Northen Sotho, Zulu and Igbo. Part of table taken from (Heid et al., 2006))

alternatively, first be morphologically analysed and then tagged the verbal suffixes based on their lexical functions. Compare table 3.3 with table 3.4. The tables 3.3 and 3.4 are illustrative excerpts from Heid et al. (2006) of Northern Sotho morphological analysis in their tagset designs.

| Module Composition | Abbreviations morpheme | Stems and Derivations |
|---|---|---|
| root + reciprocal + standard modifications | VRRec | rekana |
| | VRRecPer | rekane |
| | VRRecPas | rekanwa |
| | VRRecPerPas | rekanwe |

Table 3.3: Derivations of the verb reka

| | |
|---|---|
| rekana 'V' | *rek–* 'Vroot' *–an–* 'Rec' *–a* |
| rekane 'V' | *rek–* 'Vroot' *–an–* 'Rec' *–e* 'Per' |
| rekanwa 'V' | *rek–* 'Vroot' *–an–* 'Rec' *–w–* 'Pas' *–a* |
| rekanwe 'V' | *rek–* 'Vroot' *–an–* 'Rec' *–w–* 'Pas' *–e* 'Per' |

Table 3.4: Alternative in POS tagging of the verb reka

## 3.5 Low-Resourced Languages Corpora and Tagsets

This section reviews low-resourced languages tagsets and corpora that have been developed for African Indigenous languages[10] (AIL) and non-African. Finally, we discuss English

[10]Indigenous languages that historically belong to the continent, rather than being brought from another country. Under this definition, languages like English and Arabic are not African.

since it is one of the most spoken languages of the world and some African countries use it as their official language (e.g. Nigeria).

### 3.5.1 AIL

| | Swahili | Northern Sotho | Zulu | Cilubà |
|---|---|---|---|---|
| Number of sentences | 152,877 | 9,214 | 3,026 | 422 |
| Number of tokens | 3,293,955 | 72,206 | 21,416 | 5,805 |
| POS Tagset size | 71 | 64 | 16 | 40 |
| % of ambiguous words | 22.41 | 45.27 | 1.50 | 6.70 |
| Average% of unknown words | 3.20 | 7.50 | 28.63 | 26.93 |

Table 3.5: Corpus and tagset information for Swahili, Northern Sotho, Zulu and Cilubà. The percentage ratios are computed on 10-fold cross validation. Unknown words are previously unseen words in the training data. **Source:** Table taken from De Pauwy et al. (2012)

Table 3.5 presents four AIL corpora and tagsets statistics by De Pauwy et al. (2012). The Swahili corpus is part of Helsinki Corpus of Swahili (HCS) tagged in Standard Swahili text using SALAMA[11] (Swahili Language Manager) by Arvi (2004). In addition to tags, HCS contains other information, such as the word base form (lemma), morphology, noun class affiliation, and verbal morphology. HCS consists different text styles, such as texts from Deutsche Welle newswire to represent Swahili news and excerpts from a number of textbooks (eg. prose, fiction, education, and sciences). The size of HCS is 12.5 million words[12] and tagset used contains about 302 tags (Hurskainen, 2004).

Northern Sotho corpus annotation by De Pauw and De Schryver (2009) contains 10000 tokens and 56 tags. Microsoft Excel environment was used for the annotation based on the following reasons: computer-literate users in Northern Sotho are familiar with Microsoft Office suite and POS tagging in Excel could speed up annotation. Taljard et al. (2008) designed Nothern Sotho tagset based on the lexical and morphological criteria. The structure of the tagset are into two annotation levels of EAGLES, namely; *obligatory* and *recommended*. The authors used the *obligatory* level to distinguish the Nothern Sotho tagset into nine different classes: *concords, pronouns, nouns, adjectives, verbals, morphemes, particles, questions, and others*. From the obligatory classes, a fine-grained tagset which has 141 tags was developed. Following this was additional morphosyntactic distinctions, which led to 262 different types of morphemes. There are five features considered by Taljard et al. (2008) in Nothern Sotho tagset, namely; (1) the class membership feature, which is a classification of tags based on different classes; (2) the personal attribute features- a classification based on first and second persons (e.g., PERS); (3) the feature set of morphemes- morphemes are classified based on their lexical functions; (4) the feature set of particles- all the possible values of particles are considered (hortative, copulative, locative, etc.). For example, in Heid et al. (2006) work, all verb forms are tagged "V" except copulative verb "VCOP" and participle-like words are tagged

[11] A multi-purpose language management environment developed at the University of Helsinki.

[12] `www.csc.fi`

| Authors | Tagset size | ±Noun class | Tool? |
|---|---|---|---|
| (Van Rooy and Pretorius, 2003) | 106 | – noun class | No |
| (De Schryver and De Pauw, 2009) | 56 | – noun class | Yes |
| (Kotze, 2008) | Partial | N.R. | Yes |
| (Taljard et al., 2008) | 141/262 | + noun class | No |
| (Gertrud Faaß et al., 2009) | 25/141 | + noun class | yes |

Table 3.6: Various tagsets sizes for Nothern Sotho from Gertrud et al. (2009)

| tokens | Tags associated to each ambiguous token | Frequency |
|---|---|---|
| a | CDEM6:CO6:CS1:CS6:CPOSS1:CPOSS6:PAHORT:PAQUE:PRES | 2304 |
| go | CO2psg:CO15:COLOC:CP15:CS15:CSLOC:Csindef:PALOC | 2201 |
| ka | CS1psg:PAINS:PATEMP:PALOC:POSSPRO1psg:POT | 1979 |
| le | CDEM5:CO2ppl:CO5:CS2ppl:CS5:PACON:VCOP | 1690 |
| ba | AUX:CDEM2:CO2:CS2:CPOSS2:VCOP | 1509 |

Table 3.7: Most frequent and ambiguous words in the Northern Sotho corpus, taken from Heid et al. (2006). Compare with table 8.4 of chapter 8

each with its grammatical function; (5) a further step to indicate whether a copulative is negated, and some features (eg. locative) of the top-level tagset.
Table 3.6 shows various tagsets by different authors. The last row of the table, Gertrud et al. (2009) disregards the morphosyntactic distinctions in the tagset of Taljard et al. (2008) to reduce 141 tags to 25 top-level tags. Their aim was geared towards building a standard and structured tagset for Nothern Sotho. The high lexical ambiguity of Nothern Sotho as shown in table 3.5 and 3.7 is an evidence that languages with disjunctive writing system[13] apparently possess a high level of words with more than one tag. Possible solutions used in Nothern Sotho's multiword problems as proffered by Taljard et al. (2008) are: (1) to run tokens together with their tags without intervening spaces. (2) to use portmanteau tags[14], that is, keeping the combined tokens together, accompanied by relevant tags, which could be segregated by means of some symbol or punctuation marks. (3) to separate the fused words during lexicon-based pre-tagging using a unique lexicon as a stoplist.

The POS tagged Zulu's corpus is called Ukwabelana, which came from the Zulu's fiction and Bible translation texts (Spiegler et al., 2010; Mariya, 2012). According to Heid et al. (2006), Zulu and Nothern-Sotho corpora were prepared in the department of African Language of the University of Pretoria, South Africa. The sources of the corpora are the newspaper reports, academic texts, and internet, but those sources that are not electronically available were OCR-scanned and hand-cleaned. The followings are the description of Ukwabelana corpus: there are about 100,000 common Zulu word types and 30,000 Zulu sentences, of which 10,000 words are morphologically tagged and 3000

[13]This example shows the disjunctive writing system in Northern Sotho that mostly occur in verb prefixes (Louwrens and Poulos, 2006). For example, *ba-ka-se-sa-ngwal-el-an-a* "they shall no longer write to one another". When this is rendered in the practical orthography of Nothern Sotho, it is written as: *ba-ka-se-sa-ngwalelana*.

[14]It is a word-level tag that consists two or more alternative tags linked by "or" operator usually represented by "-". For example, in British National Corpus (C5) tagset, the portmanteau tag VVD-VVN means "either a past tense or past participle".

| Noun | Verbs | Pronouns | Particles |
|---|---|---|---|
| Adjectival | Proper | Absolute | associative |
| Deverbative | Auxiliary | demonstrative | instrumental |
| Locative | copulative | Quantitative | locative |
| | | Possessive | Possessive |
| | | | Qulificative |

Table 3.8: Sub-categorization of the main word classes of Tswana

sentences are POS tagged (Spiegler et al., 2010). Heid et al. (2006) also report that untagged corpora of Northern Sotho and Zulu comprise of 6.5 million tokens and 5.2 million tokens compiled by African Language Department in the University of Pretoria. Table 3.5 shows that about 98% of words in Zulu corpus do not need disambiguation because it is rich in morphology and has conjunctive[15] writing systems.

De Pauwy et al. (2012) present the Cilubà small POS labelled corpus of 6,000 tokens (see table 3.5). However, I couldn't find any description about the tagset they used.

In Tswana, word classes are divided on the basis of similarities between certain words. The major types found in Tswana are nouns, verbs, pronouns, particles, adverbs, idiophones and interjection. Nouns and verbs are open classes on the basis of their morphological productivity while pronouns, particles, adverbs, interjections, and idiophones are in the close classes group since they are morphologically unproductive. Fine-grained categories of the above core word classes are based on the grounds of similarities between words within a specific word category (Berg et al., 2012). In table 3.8, noun is sub-categorized into adjectival, deverbative and locative.

According to Bamba Dione et al. (2010), Wolof is a well documented language better than other West Atlantic languages (Sub-family of Niger-Congo). There are two main aspects of the language's grammar: Wolof is rich in morphology derivation for nouns and verbs, and inflectional elements, pronouns or clitics are treated as separate tokens or as verbal suffixes. Though in the tagset design, the authors remained neutral regarding how to tokenize these elements since their main goal is to design a reliable and informative tagset with respect to the syntactic function of the linguistic elements. Therefore, the internal criteria design is less important. Wolof tagset design started from scratch since no previous tagset had been designed for the language. The sources used by the authors for Wolof corpus and tagset developments are the Wolof Bible, dictionaries, and grammars books. Table 3.9 lists Wolof different tagset sizes by Bamba Dione et al. (2010). Coarse-grained tagset in Wolof contains *adverbs, prepositions, articles, comparatives, conjunctions, determiners, inflectional markers, nouns, pronouns, particles, verbs, reflexives, foreign language material, and punctuation.* One of the difficulties encountered during the tagset design for verbs was its finiteness, and the possible step adopted by the authors to find a solution was to follow a particular work of a linguist who proposed three categories for verb finiteness. These categories are POS tagged in their tagset as VVFIN, VVNFN,

[15]The practical orthography rendering of "I will work for them" in Zulu is written as one word, namely *ngizobasebenzela* instead of *ngi-zo-ba-sebenzela* in Nothern-Sotho (Louwrens and Poulos, 2006). Igbo also has a form of disjunctive writing system compared to Nothern-Sotho. This exists between auxiliary and participle verbs joined together by hyphenation (-) (marked bold in the following example). "I will work for them" in Igbo is *M* ***ga-arụrụ*** *ha ọrụ.*

| **Tagset Name** | Detailed | Medium | General | Standard |
|---|---|---|---|---|
| **Tagset size** | 200 | 44 | 14 | 80 |
| **Tags name** | ATDs.b.P | ATDs | AT | ARTD |
| | ATDp.y.R | ATDp | AT | ARTD |
| | ATDs.b.SF | ATDSF | AT | ARTF |
| | ATDs.w.SF | ATDSF | AT | ARTF |
| | ATDs.ñ.SF | ATDSF | AT | ARTF |
| | I.1p.CF.PF | ICF | I | ICF |
| | I.1p.DiFut.IMPF | IFUT | I | IFUT |
| | I.3p.NF.PF | INF | I | INF |
| | I.1p.VF.PF | IVF | I | IVF |
| | I.1s.SuF.IMPF | ISUF | I | ISuF |
| | I.3p.SF | ISF | I | ISF |
| | ... | ... | ... | ... |

Table 3.9: Different granularities found in Wolof tagset Bamba Dione et al. (2010)

VVINF corresponding to finite, deficiently finite and infinite verbs respectively. Also, there was an issue of multiword units. In this case, they used the standard tokenization format where tags are assigned to each token separated by lexical space at the first level. For example, 'inflectional sentence focus marker' followed by 'sentence focus particle'. Thus, the multiword 'maa ngi' is POS tagged as 'maa/ISF ngi/UPSF', where ISF means sentence focus inflection marker and UPSF is a sentence focus particle. Their tagset granularity is into four types: a fine-grained of 200 different classes, which they used to annotate the entire gold standard corpus; a medium coarse tagset of 44 tags; more coarse tagset using the 14 common grammatical classes; and a standard tagset of 80 tags which is define as useful for morphosyntactic studies of Wolof (Bamba Dione et al., 2010).

Yoruba is one of the major languages used in South western and North central of Nigeria. Its annotated corpus was developed from the Yoruba-English and English-Yoruba dictionaries, YLP lexical database containing 450,000 words and Yoruba lexical analyser. An output of 312,562 annotated corpus with tags was achieved (Adedjouma et al., 2013). The lexical database is the work of Awoyele released to Linguistic Data Consortium (LDC) in 2008.

Amharic is a language in the Semitic family (Gebre, 2010; Björn et al., 2009). It is spoken in Ethiopia by about 30 million speakers as first or second language (Björn et al., 2009). During the Amharic tagset development, Gebre (2010) identify some orthographic system issues, such as: allowing words to be delimited by space, words are formed by joining two or more words together to form a lexical unit, non existence of capital letters in the writing system, and the use of only consonants and long vowels. The short vowels are left for the readers to fill the gaps. Björn and Lars (2009) adopt different steps in developing Amharic corpora, and the annotation steps are corpora collection and manual tagging, automatic POS tagging, morphological analysis, and further refinement and application of the resources. Sources of their untagged corpora are Ethiopian News Headlines (having approximately 3.5 million words in Amharic text), Walta Information Center (consisting of 8715 Amharic news articles)– partly annotated with appropriate tags by human annotators (Björn and Lars, 2009), and two bilingual corpora of Amharic-English consisting of government policy files which are collected from the Ethiopian Ministry of

| htp | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | um– | aba– | um– | imi– | ili– | ama– | isi– | izi– | in– | izin– | ulu– | ubu– | uku– |
| **Class** | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n9 | n10 | n11 | n14 | n15 |

Table 3.10: Xhosa noun class prefixes developed from Allwood et al. (2003)

Information web. There have been three different tagsets developed for Amharic corpus POS tagging. The first two came from linguists in the Ethiopian languages Research Center (ELRC) at Addis Ababa University (AAU) (Björn and Lars, 2009; Björn et al., 2009). The basic tagset has 10 common grammatical classes, and one other tag (UNC) for problematic words. The 10 basic types were further subdivided into 30 types (describe in the work of Girma and Mesfin (2006)) to accommodate extended lexical functions attached to conjunction, pronoun, preposition, numerals and verbs. The third tagset was made by Sisay in 2005 (Björn and Lars, 2009). This tagset (Sisay) was used in POS tagging experiments based on Conditional Random Fields. The manually POS tagged corpus[16] of Amharic originally contains 210,000 words from 1,065 Amharic news articles tagged using 30 grammatical classes (Björn and Lars, 2009). In the POS tagging experiment of Björn et al. (2009), three tagsets were used: the largest 30 tagset developed by ELRC, the 11 basic tagset that contains 10 grammatical classes, and the tagset by Sisay (2005)[17]. In order to retain the core tags, the full tagset was mapped to 10 tags such that UNC is mapped to residual, CONJ and PREP are mapped to adposition, and N and PRON mapped to noun (Björn et al., 2009).

Allwood et al. (2003) proposes corpus-based approach for developing tagset and training data for Xhosa language of South Africa. The authors chose this method because of the challenges of linguistic phenomena most AIL are facing, such as agglutinative or morphemic merging languages. The corpus-based approach enables information retrieval from enriched corpus, which is achieved through annotating linguistic facts. The annotations are used to derive specific linguistic, grammatical and lexical patterns from the corpus. Instead of manual tagging of Xhosa, the authors proposes a computer-based-drag-and-drop tagger, and the training corpus data developed will be used to train a POS tagger for the language. Xhosa tagset design goes a bit further than the two normal tagset create levels: core POS tags and syntagmatic morphological categories. There is also paradigmatic distinctions, which tries to identify the paradigmatic inflections within a particular syntagmatic morphological class. For example, the word *abantwana* "children" in the first level will be tagged "N". In the second level, the degree of granularity is increased through POS tagging each of the prefixal, stem and suffixal morphemes based on their lexical functions. Here, *abantwana* will be tagged as "a/PREF+ba/PREF+ntw/NSTEM+ana/SUF"[18]. While in the third level, instead of the prefix PREF in the second level, they find a distinct POS tag for each of the noun class prefixes from the predefined list in the table 3.10.

[16]It is freely accessed in XML format in both Ethiopic scripts and a transliterated form from nlp.amharic.org.

[17]Developed for comparison reasons

[18]Compare with tables 9.3 and 9.11 in chapter 9.

### 3.5.2 Non-AIL

Sherpa is spoken in Nepal (South Asia) and Sikkim communities. There are about 200,000 speakers who live in Nepal, 20,000 in Sikkim and 800 in Tibet[19]. In the past, Sherpa language is spoken without letters. But in the recent years, Sherpa language scripts are based on the Sambota scripts, which is Tibetan orthography (Sang, 2005; Gelu, 2010). According to Gelu (2010), there are limited written text available for Sherpa language, therefore, the tagset developed is for the written texts available in the language. The tagset was prepared for tagging Sherpa texts in Sambota scripts following Tibetan orthography, which led to the use of tokenization that is based on Tibetan orthography. Sherpa language does not have any inflection in regard to gender, person, and a number due to its agglutinative form. It is rich in derivational morphology, and word order is subject-object-verb. In Sherpa noun phrases, modifiers follow the head noun, and there is no morphological marker to show tense. Tenses are expressed by the interaction of adverbs, aspect, and evidential marking. Sherpa tagset was developed in (Gelu, 2010). It contains 86 tags, which includes minor and major categories in the Sherpa written texts. Priority was given to the morphological and syntactic aspect during the design phase rather than the semantic aspect. The tagset is hierarchical morphosyntactic based-features. For compatibility and interoperability, general labels (NN, NP, JJ, CC, etc.) were used for grammatical types that are common across languages. Though, prominent lexical features attached to these types were further divided into subcategories in decomposable form. The written texts used lack some features like suffixes, the number, and case markers in the nominal categories. Uniform lexical markers with no morphophonemic changes are separated from the nominals and given a separate tag as suffix, while all others with morphophonemic changes are given separate tag apart from suffix. The verbal forms, aspect, mood, and evidential markers are treated as suffix and given separate tags. The auxiliary verb, copular verb, and nominalizers are treated as separate tags. The Sherpa verbal categories take negative markers as prefix. Though, at times it comes in between the verb root and causative marker to cause the negative form of the verb. In the tagset design, the negative affix is separated and given a tag. If a negative marker occurs in any adjective as prefix, it is separated from the adjective and given a tag as done in the verb. The onomatopoeic and echo-words[20] were given separate tags. There is no well defined Sambota scripts as regard to syntactic punctuation marks for off words, clauses, and enumeration. Gelu (2010) sub-categorized Sherpa's punctuation mark into three; syllable, word, and sentence boundary markers in the text and proposed a separate for them. Symbols such as brackets, mathematical operators are given separate tags.

Kurdish is a Northwestern Iranian language spoken in Eastern Turkey. Girma and Mesfin (2010) develop a medium-scale morphological lexicon for Kurmanji Kurdish using freely available lexical resources. The list of lexical categories used was developed from Kurdish reference grammar, which contains nouns, verbs, pronouns, numerals, adjectives, pre-, post- and circumposition, complementizers and several particles. Kurdish morphological lexicon called Kurlex was developed through morphological description within Alexina framework. This is achieved through converting their lexical resources into Alexina[21]

[19] `http://en.wikipedia.org/wiki/Sherpa_language`[March 2014]

[20] Words that imitate the sound they denote (Ideophones in Igbo).

[21] Alexina is an existing lexicon for Kurmanji Kurdish by Girma and Mesfin (2010)

format and using them to extract as much information as possible. A tagset consisting of 36 tags was designed and developed.

Sornlertlamvanich et al. (1999) present an initiative project by Open Linguistic Resources Channelled towards InterDiscipline research (ORCHID) geared towards developing linguistic resources for Thai and Japanese languages to support NLP research. The ORCHID corpus for Thai contains about 400k words of the National Electronics and Computer Technology Centre (NECTEC) proceedings in Thailand. ORCHID Thai corpus was developed from limited resources with most of the text entered into the system through keyboard. Apart from automatic POS tagging, all other processes were manually executed with limited software support. The Thai original POS class has 13 grammatical classes with 45 subcategories. Following NECTEC research aim, the POS classes were redefined, some tags were added to clarify ambiguity, and this led to a new 14 word classes with 47 subcategories. The redefinition of the original tags affected the classifier (CLAS) and prefix (FIXP) classes. As a measure to alleviate POS tagging difficulties in manual process, problematic cases were illustrated in their tagging scheme to act as a guidelines in determining the correct POS tagging type in the cases of potential ambiguity. An example of such guideline between verb and preposition is given based on these two classes having the same lexical forms, and making distinctions between them is difficult in POS tagging. In order to clarify how they will be tagged if encountered, the authors made the following intuitive guidelines (1) preposition cannot be negated, while verb can. (2) preposition status can be tested by moving the preposition phrase around within the same sentential context. Preposition always accompanies the proceeding noun under movement, but verb does not. ORCHID is the first project to build Thai tagged corpus.

Kumar and Josan (2012) describe the development of the Punjabi tagset for the purpose of POS tagging using machine learning techniques. Before their work, only a tagset of 630 fine-grained tags was in existence. This tagset consists of all the tags for the various word categories, word specific tags, and tags for punctuations. Sapna et al. (2011) use 630 fine-grained tags to implement HHM tagger for Punjabi, in which 503 tags out of the proposed 630 tags were found in 8 million words of Punjabi corpus[22]. Sapna et al. (2011) design a different tagset for the purpose of their work. The tagset was developed by using coarse-grained granularity for representing morphosyntactic features of Punjabi, which led to a tagset size of 40 tags. The tagset developed was compared with the existing tagsets for Indian languages.

Hardie (2003) describes the development of automatic POS tagging of Urdu texts from scratch. He started with tagset design and guidelines for manual POS and post-editing tagging. The tagset design complied with the EAGLES standard on morphosyntactic annotation where necessary. The Urdu grammar used as a model for the tagset design is based on Schmidt (1999). The tagset size developed is 400 tags, and manual POS tagging was undertaken to obtain POS tagged corpus for Urdu which serves as a training data for the implementation of POS tagger for Urdu language.

Nepali Language Resources and Localisation for Education and Communication (Nelralec) is designed to develop corpus and computational linguistics in Nepal language. Nelralec is possible via the implementation of new corpus-based lexicography methods in a new and empirical Nepali dictionary. Justifications of POS tagging a new Nepali National

[22]Texts source was online collection.

Corpus (NNC) with tags are: (1) to ensure that it is a state-of-the-art language resource, (2) it helps in corpus-based lexicography, (3) it provides an upgraded resource for language engineering implementations, and (4) to widen the range of survey available to future researchers exploiting the corpus in the analysis of the grammatical and textual structures of Nepali (Nelralec, 2006). Tagset for Nepali was developed by a team of linguists from Tribhuvan University (Hardie et al., 2005). The initial set of categories was based on the Nepali grammar of Acharya (1991). Iteratively, the tagset was implemented by using a small data samples, discussion, re-evaluation, and then re-testing it for several weeks. The authors of the tagset adopted a hierarchical architecture design, for example in VVYN1F, $V-$ indicates verbs, $VV-$ indicate finite verbs, $VVY-$ indicate third person finite verbs, etc. There are two structural features in the tagset: (1) the Nepali postpositions, which are specially written as affixes on the nouns or other words that they control, are treated as discrete tokens. (2) The tense, aspect, and modality are not marked up on the finite verbs, which are categorized solely depending on their agreement marking. This is a needful simplification for handling the very complex verbal inflections of Nepali. (Nelralec, 2006). Nepali tagged corpus for training and testing automated system was created by a team of analysts. They undertook the POS tagging of Nepali texts by hands. The process involves tokenization, assigning a tag, assembling lists of morphological rules and exceptions, etc. However, as the size of linguistic knowledge in the manually annotated dataset grew, it became possible to include that knowledge into a preparatory version of an automatic tagger, which was then run on the texts prior to manual investigation. Manual annotation of 350,000 words, which is a subsection of 1 million words of NNC took several months (Nelralec, 2006).

The Kazakh (spoken by Republic of Kazakhstan) tagset was designed based on the internal criterion principle where a tag is followed by a paradigm string whose locations mean certain grammatical aspects. There are respective paradigms along with generative scopes for POS that take inflectional suffixes. That is, the upper bound limit on a number of possible tags that can be generated from a given POS, and the different compositions of the corresponding paradigms. The maximum size of the tagset (36 tags) is equivalent to the total generative capacity (3844 tags). Depending on the extent of granularity needed for an application, some or even all grammatical aspects may be deleted or included back, providing additional adjustability. For example, *Mektepke bardym.* "I went to school". KLC tagset will represent this sentence in POS tags and its phrasal structure as follows: *Mektepke/ZEP_A0N0S0P3C3 (ZEP - impersonal noun; A0 - inanimate; N0 - singular; S0 - no possessor; P3 - 3rd person; C3 - dative case) bardym/ET_G0T3M1V0P1 (ET - regular verb; G0 - not negated; T3 - past tense; M1 - indicative mood; V0 - active voice; P1 - 1st person) ./.* (Aibek et al., 2014).

### 3.5.3 English Language

Though there are many POS lists in English, much recent language processing uses Penn Treebank tagset of size 45. The tagset has been used to label a lot of varieties of English corpora like Brown corpus, Wall Street Journal (WSJ) corpus, and Switchboard corpus. The Brown corpus consists a million words of samples from 500 written texts of different genres, WSJ contains a million words published on the Wall Street Journal, and Switchboard consists of two million words collected from telephone conversations between

1990-1991 (Jurafsky and Martin, 2014). The 45-tag tagset of Penn Treebank was collapsed from 87-tag tagset originally designed for Brown corpus. Since there have been initial works done in English POS tagging system, these tagged corpora were created by simply running POS tagger on the texts, and then human experts were used to hand-correct errors. The unambiguous and ambiguous word types rates are 86% and 14% for WSJ; 85% and 15% for Brown, while tokens are 45% and 55% for WSJ; 33% and 67% for Brown.

## 3.6 Bible As a Corpus for NLP Research

Bible have been considered in Resnik et al. (1999) as the most available, widely accessed, carefully translated because most translators believe it is the word of God, and well structured (books, chapters, verses) material for building parallel corpora. Resnik and co-authors annotated biblical texts for the purpose of creating an aligned multilingual Bible corpus for computational linguistic research such as automatically creating and evaluating translation lexicons, and semantically annotated texts for parallel translations over a wider number of languages. The Bible format enable them to easily tagged elements as b (book), c (chapters), and id attributes make it possible to identify verses independent of the context. *E.g. Mat:1:1 for Matthew, chapter 1, verse 1.*. Tapas and Philip (1999) use the Bible as a corpus for OCR evaluation across languages.

## 3.7 Conclusion

We looked at various sizes of tagset and corpus data for different languages, challenges associated with the design and development (especially in African Indigenous languages (AIL)) and how best to resolve it, and the guideline necessary for start-up design of a tagset. For the course of developing Igbo tagset and corpus, we studied existing tagsets and corpora design and developments for various languages. The strength and limitations of each tagset and/or corpus development were taken into account as guides to ensure standardization in creating our tagset and corpus. The transferring of tagset tags onto a corpus through the tagging process, and the ambiguous assumptions underlying the various operations are made clear, as in the case of how best to undergo the morphological analysis of verbs or what should be the best size of a tagset. Finally, we discussed the Bible as the most available and good start-up corpus for new language.

# Chapter 4

# Background

This chapter discusses the relevant Natural Language Processing (NLP) background. NLP involves making computers to perform useful tasks with natural languages of humans. Its strength is in its ability to draw conclusions from a pool of texts through some analysis. There are various analytical stages involve in NLP application, but we are going to restrict our discussions on the ones relevant for this thesis.

## 4.1 NLP Preprocessing Pipeline

Preprocessing is an important task and critical step in Text Mining, NLP, and Information Extraction. Text preprocessing is integral to virtually all NLP tasks. It reformats the original texts into meaningful units that contain important linguistic features before performing subsequent NLP tasks. Generally, text preprocessing steps are, but not limited to throwing away unwanted elements (e.g. HTML tags), determining word/sentence boundaries, stemming/lemmatization (optional), stopword removal, and capitalization. Preprocessing in NLP may appear simple but most often it leads to non trivial task. For example, finding lexical boundaries for words such as "Ph.D." and "can't" and multiword expressions are problematic in tokenization. This is because spaces and punctuation marks such as colon, semi-colon or periods (.) can serve as end of word or sentence marker and other purposes. Therefore, it is not sufficient to base tokenization only on the standard end of word or sentence characters. Poor text preprocessing performance will have a detrimental effect on downstream processing.

When a clean text has been created from the above process, which usually depends on research purpose, it is called a corpus. There are two concepts from the corpus that are necessary, they are token, which is occurrences of a word, and word type, an identical word as a dictionary entry.

## 4.2 Part-of-Speech (POS) Tagging

Part-of-speech (POS) or morphosyntactic tagging, lemmatization, and semantic field annotations are common corpus-based annotation. But the basic and most widely applied annotation is POS annotation, which is important in the language technology. It provides supportive information, and acts as the backbone for evaluation of both rule-based and

supervised approaches for statistical NLP. For example, POS taggers use statistical information derived from POS tagged corpus to determine the tag (a grammatical class) of a word. This prediction ability of taggers is very useful in advanced NLP tasks, such as parser and Machine Translation [MT] (e.g. *POS-tagging* $\rightarrow$ *Syntactic-parsing* $\rightarrow$ *MT*) (Jurafsky and Martin, 2007).

Generally, the resources that are necessary for POS tagging task are a corpus, tagset, and POS tagger. A corpus is the computer-readable texts prepared through texts preprocessing methods, while tagset provides tags for annotating words of a corpus. Both of them provide experimental data for POS tagging experiments. See chapter 3 for more details on corpus and tagset.

## 4.3 POS Tagger and Tagging Techniques

POS tagger could be built manually by humans or automatically trained using machine classifiers methods.

### 4.3.1 Manual POS Tagging

Manual POS tagging involves transferring of tags[1] outlined in a tagset design to the tokens of a corpus. These tags are apply to the tokens based on the context in which tokens are used, and to adequately capture the grammatical distinctions of the language under study. Tagset are mainly formulated through lexical grammar database of a language rather than direct looking at the tokens of a corpus and making decisions. Applying tags of a tagging scheme (tagset) may involve the use of

1. core POS tags, the obligatory level [Noun, Verb, Pronoun, etc.] of EAGLES Leech and Wilson (1999),

2. high level of granularity that involve tags developed based on the *recommended* and *optional* level of EAGLES. These tags are dependent on the core tags,

3. a more fine-grained tags that involve decomposing tags of stage 2, such as POS tagging each of the prefixal, stem and suffixal morphemes of a token based on their lexical functions.

From the above illustration, the higher the granularity level the more complex is the tagging scheme. It is necessary to be conscientious in considering tagset size while designing it since the smaller the granularity of a tagset, the higher the accuracy and less the ambiguity (De Pauwy et al., 2012). Also, it is important that all outstanding grammatical classes are handled while maintaining the economy-size of a tagset.

Information regarding dealing with challenging phenomena in a language is expressed in the tagging guideline of a tagset. However, without testing how effective it is through manual implementation, it would be a difficult task to identify and design such important guidelines in the first place. Hardie (2003) states

[1] Also known as part of speech (POS) tag.

> "In theory, the discovery of areas of problematic classification, and the creation of tagging guidelines, could be done in the process of developing an automated tagger. However, it does not seem conceivable that this could be an easier way to produce the guidelines than via the process of manual tagging."

In POS tagging experiments, manual POS tagged corpora are required as training data for taggers. Even in some cases, such as unsupervised methods, some manually annotated corpora are still required as a benchmark in evaluating taggers performance.

Manual POS tagging in the modern corpus linguistic is infeasible since computational linguistic research now dealt with millions of tokens at a time. Semi-automatic methods that normally involve human experts in the loop are used to bootstrap manual annotation process. This approach could be *monolingual-based* focusing only on the target language, *bilingual-based* focusing on two languages, and *multilingual-based* focusing on more than two languages or combination of them. In both bilingual-based and multilingual-based approaches, at least there is a resource-rich language and thus other languages will have numerous borrowings from it. An obvious example of monolingual-based is to manually annotate a small part of a text, use the annotated-text to train a POS tagger. Select a part of the same text that is not annotated (recommended selecting a larger size), use the tagger to annotate the selected part. Hand correct those annotations, retrain a tagger on that corrected part plus the initial one, and so on. Do and continue on this process until a large amount of good quality annotated data is produced or you become weary of the process. Apart from this example, there are others like Bamba Dione et al. (2010) that use heuristics for semi-automatic annotation to develop a gold standard for training automatic POS taggers. The authors used GATE (Cunningham et al., 2002) to tag 26,846 tokens of the Matthew gospel taken from the Wolof Bible. First, they automatically tagged the Matthew corpus with guessed tags, and then meticulously hand-checked and corrected all the automatic tagging steps. Girma and Mesfin (2010) use the lexical information obtained from Kurdish freely available language resources to automatically generate annotated POS corpus. Adedjouma et al. (2013) use an eight-step-automatic method to POS tag Yoruba corpus, and then manually correct the final outcome.

### 4.3.2 Automatic POS Tagging

Basically, rule-based and probabilistic-based are techniques in NLP for assigning POS tags to words in a sentence. The former one assigns tags based on rules; rules can be hand-crafted-based (Karlsson, 1995) or corpus-based (Brill, 1995a). While the latter assigns tags based on probability models (Ratnaparkhi et al., 1996). Both techniques learning model can be supervised or unsupervised. While supervised taggers use manually POS tagged corpus-data to automatically generate statistics or rules for tagging operations, unsupervised method automatically generate POS taggers without use of manually POS tagged corpus (Brill, 1995b). The followings are widely used and well evaluated supervised POS taggers that have been used on most European, and a few other world languages.

- *Baseline Tagger*: Unigram in computational linguistics and probability refers to a single token. Therefore, a unigram tagger assign tag based on most common tag of a single word. For example, a unigram tagger will classify "race" as "NNC" since it derives from training corpus that "race" is more often tagged as "NNC".

Unigram-based tagger finds the *most probable tag* for each word by computing the frequency of tags assigned to each word in a training corpus. While common noun "NNC" is mostly use as a default tag for classifying unseen words in the training corpus. Although unigram tagger is a context-independent type of tagger, it can achieves an acceptable results on a large training corpus-data. The results it achieves are normally use as a baseline for more sophisticated taggers.

- *Hidden Markov Model (HMM) Tagger*: A HMM tagger generally chooses a tag sequence for a given sentence rather than for a single word. For instance, given a sentence $w_1 \ . \ . \ . \ w_n$ , a HMM based tagger chooses a tag sequence $t_1...t_n$ that maximizes the following joint probability:

$$P(t_1...t_n \ , \ w_1...w_n) = P(t_1...t_n \ )P(w_1...w_n \ |t_1...t_n)$$

  In practice, it is often impractical to compute $\mathrm{P}(t_1 \ . \ . \ . \ t_n)$. Therefore many different taggers have been proposed to simplify this probability computation. Example of such tagger is TnT (Brants, 2000b), which is one of the most commonly used HMM based tagger. It uses second order Markov models to simplify the computation; it assumes that the tag of a word is determined by the POS tags of the previous two words.

- *Maximum Entropy (ME) Based Tagger*: Unigram and HMM taggers computes probability based on $P(tag|tag)$ and $P(tag|word)$. The addition of knowledge source, such as word features, to improve tagger's performance will require some conditioning, and each time new feature is added, the conditional probability gets harder leading to computational complications. According to Ratnaparkhi et al. (1996), ME-based tagger is introduced to provide a principled way of incorporating complex features into probability models. For example, given a sentence $S$ made of $w_1...w_n$ words, an ME-based tagger computes the conditional probability of a tag sequence $t_1...t_n$ as:

$$P(t_1...t_n|w_1...w_n) \approx \prod_{i=1}^{n} P(t_i|C_i)$$

  where $C_1...C_n$ are the corresponding contexts of each $w$ in $S$. The context $C$ of a $w$ also includes $t_{i-1}$ (previous tag before the current $w$). An ME-based taggers use this feature set to compute $P(t_i|C_i)$. The idea is to learn the weights of the features with the highest entropy from distributions that satisfy a certain set of constraints using the training corpus. Example of ME-based tagger is Stanford Log-Linear POS tagger implemented in Java by Toutanova et al. (2003).

- *Transformation-based Error-Driven Tagger*: This method utilizes rules generated from the training corpus commonly called transformations. These transformations are used to extract readable grammar directly from the training data without human linguist intervention. The training data is manually and correctly tagged corpus. The corpus size is usually small, and it serves as input to the initial annotator. Transformation-Based-error-driven Learning (TBL) works by automatically detecting and remedying errors in a pre-tagged corpus, and incrementally improving its learnt

model. It initially assigns unigram tagger's tag to each token in an untagged corpus resulting in a temporary tagged corpus. The unigram tagger derives information for choosing the most probable tag for each token from the tagged corpus called the truth. Iteratively, the temporary tagged corpus is compared to the truth corpus through the TBL learner module, and a new rule with a positive impact is added to the rule list each time. The process is repeatedly executed until a given threshold is reached, and temporary tagged corpus resembles or close to the truth. At the end, this process produces an ordered list of transformations to applied on the test data. This was originally developed by Brill (1995a) and subsequently improved both in speed and performance by Ngai and Florian (2001) and Hepple (2000).

- *Similarity-Based Reasoning Tagger*: Similarity-based reasoning is a method in intelligent system that draws conclusions by finding similarity between entities. Daelemans et al. (1996) introduce a memory-based supervised learning techniques to POS tagging based on similarity reasoning. The tag of a word in a particular context is generalised from the most similar cases held in memory.

## 4.4 Methods to Improve POS Tagging Performance

Here we describe various techniques that can improve automatic POS tagger's performance.

### 4.4.1 Strings Extraction Methods For POS Taggers

One of the problems facing POS tagging systems is the aspect of unknown words; words that are seen in the sentences, but are not found within the lexicon of a tagger. This problem gets worse as tagger gets more texts because new words are constantly added to the language, and people are likely to use words that are outside a tagger's lexicon. Basically, there are two methods to dealing with the problem of new or previously unseen words (also called unknown words). First is to build a complete lexicon and handle unknown words in a basic way, that is, either block the word or ask for information about the word from the user. Second is to perform word analysis, which would allow the tagger to analyse features in the sentences that contain unknown words in a robust way. Unknown words could be learned by looking at the word itself, surrounding words and tags and these features are stored as part of information in the lexicon. Thus, if the a similar word is encountered again later, tagger would guess its tag using information in the lexicon. Features analysis by tagger in a sentence could be definition of closed and open classes, characters extractions or morphological analysis.

Various works have been done in extracting strings from a word, which in effect serves as a proxy for actual linguistic suffixes. Extracted strings are used as features by taggers for prediction. The feature generation methods used by these taggers are based on last/first letters of a word. We are going to look at ways taggers have achieved this focusing only on taggers used in this work. We discuss the following taggers:

- Stanford Part-of-Speech tagger *SLLT* (Toutanova et al., 2003) uses its *-arch* module to determine what features are used to build a tagger. This module contains *ExtractorFramesRare* use to extract features like wordshape, suffix or prefix for rare

words. Rare words are determine by setting a threshold, and all training words whose frequency that fall below this threshold are rare. Therefore, for rare word *well-dressed*, the module would generate non-zero valued features like *prefix($w_i$) = w, prefix($w_i$) = we, prefix($w_i$) = wel, prefix($w_i$) = well; suffix($w_i$) = ssed, suffix($w_i$) = sed, suffix($w_i$) = ed, suffix($w_i$) = d; has-hyphen($w_i$); wordshape($w_i$)=xxx-xxxxx; short-wordshape($w_i$) = x-x* (Jurafsky and Martin, 2014).

- Trigrams'n'Tags *TnT* (Brants, 2000b) uses *-endings* analysis called *suffix tries* which is based on probability distribution. The *-endings* are generated from words in the training corpus that have a set of fixed length. This module assumes that rare words' *-endings* are better approximation for predicting unknown words rather than *-endings* found in the known words.

- Transformation-Based Learning *TBL* (Brill, 1995a) uses cues learned from training data in predicting the most likely tag for the unknown words. For example, an unknown word is labelled proper noun if capitalized else common noun. Also, the transformation templates include the use of prefix and suffix length (if specified) in predicting to change initial tag X to Y.

- Memory-Based Tagger *MBT* (Daelemans et al., 1996) uses feature patterns defined to add extra information to the tagger concerning the contextual information and the formation of the words to be tagged. This is done by the parameters *-p*– feature pattern for known words, and *-P*– feature pattern for unknown words.

- Hungarian Part-of-Speech *HunPOS* (Halácsy et al., 2007) is a re-implementation of TnT. HunPOS uses tag transition (*-t N*) and emission probability (*-e N*) the same way TnT does, but it is flexible because for some languages and applications, "*-t3 -e2*" (previous three tags and previous two words) may be favourable for unknown instances instead. *-f N* estimates an unseen word's tag distribution based on the tag distribution of rare words. 'Rare' is defined as seen less than N times in the train corpus. It also uses *-endings* as in TnT.

### 4.4.2 Morphological Analysis

Thede and Harper (1997) define three levels of morphological analysis: reconstruction, generation and recognition; *morphological reconstruction* processes an unknown word by using information regarding the stem and affixes of that word; *morphological generation* studies the ability of morphological affixation rules to build new words from a lexicon of stems; *morphological recognition* uses knowledge concerning affixes to deduce the possible POS and other features of a word, without using any direct knowledge about the word's stem. Most of unknown word research revolves round the morphological reconstruction and generation. Morphological analyser in Morphy tool performs inflectional analysis by determining the stems using a dictionary of stem and their corresponding inflection types (Lezius, 2000).

IceMorphy is an unknown word guesser in IceTagger. Its morphological analyser performs analysis as: for any given word, the morphological class is guessed based on *-endings* of the given word. The stem of this word is extracted, and all possible

morphological -*endings* are generated. This is a string search, where $s_i = (i = 1, ..., k)$, such that $s_i = r + ending_i$, where r is possible morphological -*endings*. It uses a dictionary to look-up $s_i$ until a word is formed having same morphological class or no word match found (Loftsson, 2007).

Arabic is a highly inflectional language with many morphological and grammatical features (Sawalha and Atwell, 2010). Daoud (2010) develops algorithm that given Arabic word, it decomposes into its root and affixes based on the affix analysis that takes advantage of the statistical studies of the diacritical Arabic morphological features. Sawalha and Atwell (2009) develop morphological analyzer that uses linguistic knowledge of Arabic language as well as corpora to verify the linguistic information. Sawalha and Atwell (2010) develop fine-grained feature tagset and fine-grained morphological analyser for Arabic for the purpose of improving correct analysis of Arabic words. Their morphological analyser uses linguistic lists of functional words, named entities and broken plural lists in its lexicon.

### 4.4.3 Combination of POS Taggers

This is combination methods that involves more than one tagger for the purpose of correcting the biases of individual taggers. Use of combination of taggers in POS tagging task has often shown a higher tagging accuracy than achieved by individual taggers. The reason is that different taggers yield different errors. These differences, provided they are complementary and systematic, can be used to improve results. For a tagger combination pool, it is thus necessary to use taggers that are developed based on different language models (Loftsson, 2007).

### 4.4.4 POS Taggers Integration Method

This means integrating a functional module of one tagger into another such that the outcome runs like a single tagger. In order to achieve this, it is often important to have access to the source code of taggers involved. This fact, indeed, is probably the reason why integration methods are not frequently discussed as stated by Loftsson (2007).

## 4.5 Measures for Evaluation

The goal of evaluation in POS tagging system is to understand how well a tagger performs on a specific language texts, either for comparison with other taggers or for understanding whether a new POS tagging system is needed for the language. The standard and generally used evaluation methods are outlined below.

The holdout technique is the easiest kind of cross validation. The corpus data is separated into training and testing sets, and the training set is then used to train taggers. Then the taggers predict the tags for the tokens in the testing set (usually unseen similar texts). The errors taggers make during testing are gathered which give the mean absolute test set error, this is used to evaluate taggers' performance. K-fold cross validation is an improved version of holdout method. The corpus data is separated into k subsets, and the holdout method is iterated k number of times. In each iteration, one of the k subsets is used as the test data and the remainder (k-1) subsets are combined to form a

training data. Then the average error over all k testing is computed. The advantage of this technique is that how the data gets divided doesn't matter much, and every data point gets to be in a test data exactly once, and gets to be in a training data k-1 times. The following formulas are metrics used for this calculation

$$accuracy = \frac{\text{number of tokens correctly tagged by tagger}}{\text{total number of tokens}} \tag{4.1}$$

$$errorrate = 1.0 - accurracy \tag{4.2}$$

To find how many tags per token assigned by the taggers

$$\text{ambiguity rate} = \frac{\text{total number of unique tags per word type}}{\text{total number of word types}} \tag{4.3}$$

We also measure the quality and quantity of taggers output by calculating the precision and recall. Precision finds the answer for "tokens given this tag, do they suppose to get it?", while recall "all tokens that are suppose to get this tag, did they get it?". Thus, we calculate them as follows

$$precision = \frac{\text{True Positive}}{\text{True Positive + False Positive}} \tag{4.4}$$

$$Recall = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \tag{4.5}$$

$f$-measure now finds the harmonic-mean of precision and recall by

$$fmeasure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.6}$$

## 4.6 Conclusion

This chapter discusses the introductory part of Natural Language Processing (NLP) focussing on the resources used in this study. Preprocessing of texts before performing subsequent NLP tasks is a vital stage required to remove unwanted stuffs that could cause noise in the corpus data. Part-of-speech (POS) tagging is a type of supervised learning techniques, and as such requires a well POS tagged training data (usually manually annotated) to learn from, and previously unseen but similar test data for testing. Therefore, it is the utmost importance that a well annotated clean corpus with suitable tagset be used for developing the training data usually called the gold standard. There are different application methods to POS tagging system, and each adopts different machine learning (ML) approach(es). We have discussed the methods and ML approaches in this chapter. For a tagger to be considered robust, it must efficiently handles words that are not yet seen during training phase, the existing techniques on how most taggers handle previously unseen words in training data are also discussed. Finally, we discuss the necessary metrics for measuring taggers performance in order to ascertain how well they will do on a particular language.

# Part II

# Data Development

# Chapter 5

# Igbo Corpus Development

This chapter discusses the NLP pipeline processes involved in developing Igbo corpus, which started with data collection and preparation.

## 5.1 Corpus Data Collection

We discuss Igbo corpus collection and challenges in this section.

### 5.1.1 Electronic Text

Electronic text itself is not necessarily a corpus, rather it is an unstructured mass of textual data. It must go through stages of processing to be redefined as a corpus. The development of a corpus is based on its scientific purpose giving the features required for the research. Corpus is a systematic collection of pieces of language texts in electronic form, that will represent as far as possible language features relevant for computational linguistic research. The principle of selecting contents of a corpus should be based on the examination of communicative function of the text in the community in which it arise (Wynne et al., 2005). For example, in building a contemporary corpus for Arabic, Al-Sulaiti and Atwell (2006) consider collection of texts that will largely reflect the reality of the language.

For the purpose of this research, we considered homogeneous collection of Igbo texts for our corpus development. We collected two different types of electronic texts: the New World Translation (NWT) Bible[1](represent religious texts) was collected via online source and a novel written by a native speaker. This novel was handed over to us (in softcopy) by an experienced and senior Igbo linguist who certified that the author used modern and orthographic standard of Igbo. The purpose of using Bible and novel texts in our research is to enable the testing of a Part of Speech (POS) tagger on different text styles. The Bible represent religious texts, while novel represent modern Igbo texts[2]. The following section discusses the reasons why we imposed restrictions on texts collection.

[1]Obtained from jw.org

[2]It was written in 2013.

### 5.1.2 Possible Sources and Issues

There are possible sources of Igbo electronic text collections that are representative of the language's communicative function as used in the Igbo community. These can be sourced through the web and resorting to old-fashioned methods such as keyboarding or Optical Character Recognition (OCR) scanning of existing textbooks. The web sourcing for corpus development has been widely recognised in the literature by several authors (Resnik, 1999; Baroni and Kilgarriff, 2006; Scannell, 2007) as valuable. We collected some Igbo texts through web by using *wget* ( a free utility for non-interactive download of files from the web), but we avoided Scannell (2007) type of web crawling because of the following issues. Igbo language has about thirty dialects, each with its own writing conventions, and therefore sourcing useful Igbo texts is a non-trivial task. Obtaining such collections through text mining from the web will result in a heterogeneous collection of text. Therefore, the task goes beyond text collecting and extends to collecting a relatively large homogeneous dataset based on consistent conventions. Heterogeneous collection of texts could introduce errors such as wrong word type statistics. In 2013, at one of the University of Sheffield NLP group seminar, I had the opportunity to access Sketch Engine[3] for Igbo texts collection, but was discouraged when I observed some errors due to different Igbo dialects and typography. Errors such as "nine" and "niile" are different writings of the English word "all" in different Igbo dialect, and "nile" instead of "niile" is writer's typographical error. Crawling the web without restriction to a consistent convention could possibly add these words which when word type is calculated would see each as a different type. The aim of our project, which is to develop POS annotated corpus data for automatic POS, requires consistency in the language orthography.

Next is the issue of tonal markings. The first major surprise is that Igbo texts 'by native speakers' written 'for native speakers' are usually not tone-marked. Indeed, the tone marking conventions described and illustrated in the sections above are usually found in journal or especially academic articles. The Igbo Bible introduced by Church Missionary Society (CMS) in 1931 took more than seventy years to produce (Rowbory, 2009). It is the oldest and biggest composite text of the language, but it is not tone marked and only available in hard-copy. As stated earlier, the 36 graphemes of the Standard Igbo (SI) orthography consists a set of conventions for writing the language, which includes letters in lower and upper cases, digraphs, writing diacritics such as dot below (ọ,Ọ) and pronunciation. But it contains no diacritic symbols "*High* [H] = [ ´ ], *Low* [L] = [ ` ], *downstep* = [ ⁻ ]" for tonal representation (see Omniglot (2016) and Ọnwụ Committee (1961)). Some of the existing Igbo texts (both in hard, soft or web copies) written in SI followed this orthography with or without tones. Writers of these texts mainly use tones in cases where semantic disambiguation are needed.

Following this circumstances, we found the NWT Bible of Jehovah's Witness useful. It is partly tone-marked, and writers consistently used writing diacritics and conventions of SI orthography in the Bible. Therefore, it becomes necessary to use the Bible as a benchmark for comparison to check other electronic texts that are based on SI orthography for this research. Religious texts have been stated in the literature as being orthographically consistent, and have been shown to be the most available and widely accessed electronic

[3]It is a corpus tool use to create and search text corpora in many languages via online sources. https://www.sketchengine.co.uk/.

book (Resnik et al., 1999; Tapas and Philip, 1999; Alrabiah et al., 2014). This is because they are found mostly in the public domain and is believe that they are carefully written or translated from one language to another. To large extent, religious texts use words written clearly in standard form of the target language for easy readability and assimilation by the native speakers. Therefore, it is a good starting point (only if it is available) for creation of language processing technology for under-resourced languages, such as Igbo. We chose NWT version downloaded from the web[4]. It is the *only* religious text in Igbo that adopted SI orthography, and is available in soft-copy as of the time of this research. Igbo people are predominantly Christians with pagans, Islam and other religions sharing the remainder. The NWT Bible generally does not adopt a particular tone marking system, neither is there a consistent use of tone marks for all the sentences in the Bible. Instead, there is a narrow use of tone marks in specific and restricted circumstances throughout the book. An example is when there is a need to disambiguate a particular word. For instance, *ihe* without tone mark could mean 'thing' or 'light'. These two are always tone marked in the Bible to avoid confusion; hence *ìhè* 'light' and *íhé* 'thing'. The same applies to many other lexical items. Another instance is the placement of a low tone on the third person pronoun to indicate the onset of an interrogative sentence, which otherwise would be read as a declarative sentence. This particular example has already been cited as one of the uses of tone marks in the language. Apart from such instances, the sentences in the Bible are not tone marked. As such, one cannot rely on such narrow use of tone marks to generalize conclusions on the use of tonal diacritics in the language. But words of the Bible texts play the same grammatical roles as in the non-tone marked, not fully tone-marked, and fully tone marked in Igbo. Although the writers narrowly used tonal diacritics in the Bible, they are consistent in the use of writing diacritics such as dot below (ọ,Ọ).

With regard to corpus gold standard design and development in general, we are in a somewhat special situation as the author combines (i) the expertise in NLP and (ii) native speaker knowledge of Igbo in one person, and an expert in Igbo linguistic that collaborated with us has described the collected texts as almost consistent in the use of Igbo SI. Therefore, to a large degree, we can generalize that the collected texts represent the reality of Igbo language. To the best of our knowledge these texts are among the very few texts using tone marking system in a particular pattern that has already been cited as one of the uses of tone mark in the language.

## 5.2 Character Encoding

The major set back that hindered availability of most languages' texts (in effect labelling them as "low-resourced") is lack of appropriate character encoding processing programs, which only is available for European languages. It is only recently that most generic software platforms have adopted an international encoding standard (usually called unicode like UTF8) for use with different languages and scripts. This development has favoured the font (alphabets) and software problems of most languages like Arabic (Atwell et al., 2004) and Igbo (Uchechukwu, 2005, 2006). Consequently, the period of search for appropriate font and text processing programs for writing Igbo texts is now in the

[4] `jw.org`

past. Preserving Igbo character encoding in text preparation is very necessary to avoid the issue of wrong separate tokens due to wrong encoding format. For example, wrong Igbo character encoding will cause this phrase in the Bible after tokenization to have the form "*Jesi amụọ Devid bu ´ . eze* instead of "*Jesi amụọ Devid bụ́ eze*" . This issue was resolved in the tokenization section.

## 5.3 Data Preparation

**Preprocessing**: The NLP processing pipeline of the Igbo electronic texts involves downloading web pages of the NWT Bible for the languages, stripping the HTML tags and trimming to get desired content like starting each verse on a new line; this is then normalized and tokenized.

### 5.3.1 Trimming and Normalization

NWT Bible was downloaded webpage by webpage using *wget* command. To make neat the downloaded Igbo electronic[5] texts, all web associated tags like HTML were removed. All the downloaded files were compiled into a single file having books, chapters and verses arrangements. The books were separated into new and old testaments and verses of each chapter were formatted to start new lines.

In the normalization, tokens in this form Mid'i·an, Ca'naan·ites, Am'or·ites, etc., in English were normalized to Midian, Canaanites, Amorites and in Igbo, some tokens like *bú* instead of *bụ́*, *m* instead *m̀* , combining grave accent (̀), combining accute accent (́ ) and combining dot below (̣) that were seen as separate tokens were all corrected. Some of the Hebrew symbols א, ב, ך associated with the book of Psalms "*Abụ Ọma*" were removed. Also, some words' characters in the texts were found written separately. For example, *k ọ ọ k w a* instead of *kọọkwa* and *n ụ r ụ k w a* instead of *nụrụkwa*. Samples in this form were corrected to avoid calculating wrong tokens and word type totals. The writing system NWT Bible adopted conform to the Standard Igbo orthorgraphy, word like *niile* which can come in forms like *nine, nile, etc.* is correctly written in all instances. As stated earlier, the NWT Bible generally does not adopt a particular tone marking system, instead, there is narrow use of tone marks in specific circumstances, like when there is a need to disambiguate a particular word, throughout the book. To the best of our knowledge this NWT Bible is one of the very few texts using tone marking system in a particular pattern that has already been cited (see Phonology section in chapter 2) as one of the uses of tone mark in the language.

During cleaning-up exercise of the initial part-of-speech tagged Bible texts in chapter 6, some words found wrongly written were normalized. For example, the words originally written as *ọ bụla* "any" in the Bible texts, and *ọzọ dị mgba* "chimpanzee" were normalized to *ọbụla* and *ọzọdịmgba*. This led to addition of normalization component in the tokenization methods of section 5.3.2.

[5]jw.org

| | | | |
|---|---|---|---|
| 1 | Obi richara nri ahụ | Obi eat.completely.PAST food DET | 'Obi ate up the whole food.' |
| 2 | Obi ga-ericha nri ahụ | Obi aux-eat.completely food DET | 'Obi will eat up that food.' |
| 3 | Obi ga-erichariri nri ahụ | Obi aux-eat.completely.must food DET | 'Obi must eat up that food.' |

Table 5.1: Igbo morphological structure

| | |
|---|---|
| batabeghịkwa | ba+ta+be+ghị+kwa |
| batabekwaghị | ba+ta+be+kwa+ghị |
| bịaghịkwa | bịa+ghị+kwa |
| bịakwaghị | bịa+kwa+ghị |

Table 5.2: Morphemes attachment to Igbo verbs

### 5.3.2 Tokenization

Tokenization is an important preprocessing step required to adequately separate words in sentences into units of information. Tokenization for alphabetic, whitespace, and punctuation segmented languages such as English is considered a relatively simple process compared to morphologically rich languages. Errors made at early stage of development are likely to introduce more errors at later stages of text processing (Pretorius et al., 2009). Therefore, tokenization and normalization for good text preprocessing is a necessary step.

#### Word Segmentation Issues in Igbo

Every language has a level of peculiarity that will create difficulty for NLP tasks. The problem is further aggravated if the language under study is rich in morphology. A focus on the challenges of Igbo verb tokenization shows that Igbo is no exception as it is an agglutinative language. In agglutinative languages, morphemes are suffixed mainly to a verb or to other POS class to form a complete morphological structure (Anderson and Petronella, 2006). This means that what would be expressed as several lexical units in English can equally be expressed in Igbo as a single unit through affixation. There are three examples in table 5.1. The morphemes involved are: the verbal vowel prefix *e-*, verb root *-ri* "eat", extensional suffix *-cha* indicating "completion", inflection (-rV) *-ra* indicating "past tense", *-rịrị* "morphemes showing compulsion". While *ga-* is an auxiliary usually hyphenated to the verb participle.

In example 1 of table 5.1 annotating *richara* as a verb will be misrepresenting a fact in the language, because only *ri-* is a verb; *-cha* is the "completive suffix", while *-ra* is the past tense marker. This is more complex in example 2 where we have the modal suffix "-cha" which itself is not a verb. For example 3 *erichariri* can not be called a verb - the best solution is to segment and annotate these morphemes separately, according to their grammatical functions or to have special tags indicating occurrence of morphemes (affixes) in the verbs or any other class. We cannot hide them without giving the wrong picture of what constitutes a verb in the Igbo language.

Apart from the above challenging factors, the attachment of some morphemes to Igbo verbs has inconsistent pattern of occurrence. The table 5.2 shows the random sequence of occurrence between *-ghị* and *-kwa*. Both words grammatically have the same meaning but can be written orthographically as in table 5.2 depending on the writer.

A good tokenizer algorithm is a prerequisite for understanding the verb structure

in any language. Improper morphological analysis in tokenization may distort corpus linguistic conclusions and statistics. The above examples emphasise the importance of a robust morphological analyser for Igbo language, particularly in light of the increased exploitation of electronic corpora for linguistic research. Apart from Igbo verbs, there are other POS classes that are inflected with affixes such as conjunction *nakwa (na+kwa)*, demonstratives *ahụkwa (ahụ+kwa)*, adjectives *ọzọkwa (ọzọ+kwa)*, etc.

**Tokenization Method**

For the sake of start-up, we tokenized based on the whitespaces– Igbo language uses whitespace to represent horizontal or vertical space in typography and punctuation. In addition to tokenization, we used regex to perform the followings:

*Separate tokens if the string matches:*

- "ga-" or " n' " or " N' " or "na-" or "Na-" or "ana-" or "ịna-"; for example, the following samples *n'elu, na–erugharị, ịna-akwa, ana-egbu* in the Bible will be separated into *n', elu, na–, erugharị, ịna- , akwa, ana-, egbu* tokens. As in the UPenn scheme where verb contractions (won't) and Anglo-Saxon genitive of nouns (children's or parents') are split into their component morphemes (wo n't, children 's and parents '), and separate tags are assigned to each component (Atwell, 2008). " n' " or " N' " (if " n' " begins a sentence) are prepositions orthographically written in full as "na or Na" (*na mbido* "in beginning") but which lose "a" whenever it follows a word that starts with a vowel (*a,ị,ụ,ọ,e,i,u,o*), for example, *n'elu* "on top". Using regex, any occurrences of such or related cases are split, so that tags can be assigned to each part separately according their grammatical behaviour (n'/PREP elu/NNC).
- any non-zero length sequence consisting of a–z, A–Z, 0–9, combining grave accent ( ̀ ), combining acute accent ( ́ ), combining dot below ( ̣); for example, these words *bụ́, m̀, ìhè, ahụ́, ájà* in the corpus will be separated as tokens with their diacritics. Tokenization without considering these diacritics will classify each diacritic symbol as a token thereby misrepresenting the Igbo words. See figure 5.1 for use of diacritics in the Igbo words.
- any single character from: left double-quotation mark ("), right double-quotation mark (”), comma (,), colon (:), semicolon (;), exclamation (!), question (?), dot (.); here, *... otú a, sị: "Nwoke ..."* will have the form *... otú a , sị : " Nwoke ... " .*
- any single non-whitespace character.

In addition to this, we added component for normalizing wrongly written words. In place of sentence splitting, we use (i) verses for the Bible since all the Bible 66 books are written in verse level) and (ii) sentences for other texts (novel, essay, etc.), but we maintain a sentence length not $> 100$ for all the texts. Sentence or verse lengths that

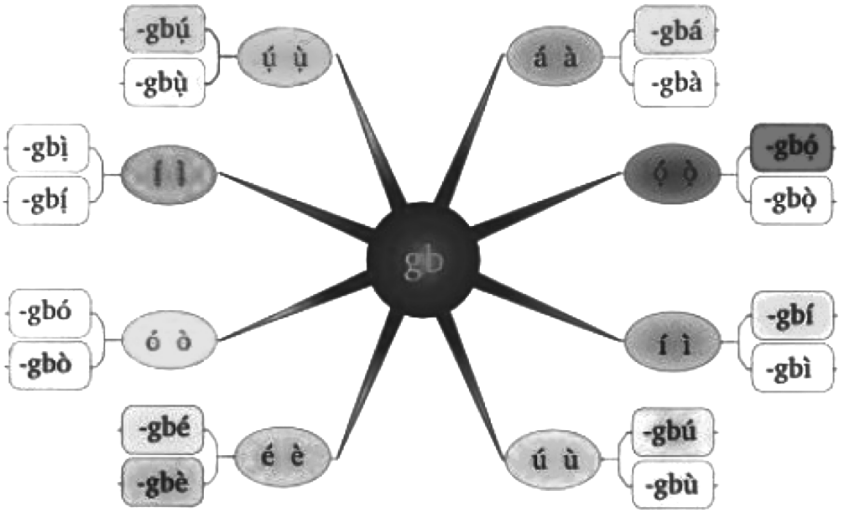



Figure 5.1: Diacritization in Igbo vowels to represent accents from Uchechukwu (2006)

are more than 100 are split into newlines and identity numbers are assigned to them as links to the original sentences or verse. Verse at line 1780 (last chapter of the book Mak) of the New testament Bible was 145 word length, it was split into two verses (1780a and 1780b) increasing sentence size by one. The major aim here is to use the output of this tokenization algorithm to implement the new Igbo tagset, which will capture all the inflected and non-inflected tokens in the Igbo corpus for further analysis.

### 5.3.3 Corpus Statistics

In this section we discuss the statistics of Igbo corpus based on the tokenization method above.

#### Before and After Tokenization

Word segmentation of the Igbo corpora based on the *whitespace* and *whitespace and punctuations* are given in table 5.3.

| | Whitespace | | Whitespace and Punctuation | |
|---|---|---|---|---|
| Name | Sentence | Tokens | Sentence | Tokens |
| Bible | 32416 | 849524 | 32416 | 1165944 |
| Novel | 351 | 351 | 351 | 36552 |

Table 5.3: Corpus statistics after using whitespace and punctuation tokenization on the Igbo texts

| Old testament books | # of Sentence | # of token | New testament books | # of Sentence | # of token |
|---|---|---|---|---|---|
| Jenesis | 1583 | 50800 | Matiu | 1099 | 33395 |
| Ọpụpụ | 1253 | 42559 | Mak | 695 | 21287 |
| Levitikụs | 886 | 31768 | Luk | 1175 | 36101 |
| Ọnụ Ọgụgụ | 1324 | 42099 | Jọn | 901 | 27638 |

| | | | | | |
|---|---|---|---|---|---|
| Diuteronomi | 993 | 35772 | Ọrụ Ndị Ozi | 1035 | 33808 |
| Jọshụa | 682 | 22265 | Ndị Rom | 449 | 14886 |
| Ndị Ikpe | 639 | 23452 | 1 Ndị Korint | 453 | 14728 |
| Rut | 89 | 3338 | 2 Ndị Korint | 270 | 9905 |
| 1 Samuel | 841 | 32033 | Galeshia | 155 | 4991 |
| 2 Samuel | 719 | 25937 | Ndị Efesos | 161 | 4818 |
| 1 Ndị Eze | 838 | 31296 | Ndị Filipaị | 108 | 3548 |
| 2 Ndị Eze | 744 | 29056 | Ndị Kolosị | 99 | 3353 |
| 1 Ihe E mere | 971 | 25950 | 1 Ndị Tesalonaịka | 94 | 2974 |
| 2 Ihe E mere | 858 | 32579 | 2 Ndị Tesalonaịka | 50 | 1631 |
| Ezra | 290 | 8954 | 1 Timoti | 119 | 4003 |
| Nehemaya | 419 | 13618 | 2 Timoti | 87 | 2696 |
| Esta | 177 | 7044 | Taịtọs | 49 | 1675 |
| Job | 1112 | 26154 | Failimon | 26 | 738 |
| Abụ Ọma | 2731 | 63520 | Ndị Hibru | 316 | 10664 |
| Ilu | 946 | 22083 | James | 113 | 3495 |
| Ekliziastis | 234 | 7543 | 1 Pita | 110 | 3967 |
| Abụ Solomon | 125 | 4153 | 2 Pita | 64 | 2392 |
| Aịzaya | 1358 | 51720 | 1 Jọn | 110 | 3700 |
| Jeremaya | 1416 | 56532 | 2 Jọn | 14 | 422 |
| Abụ Akwa | 163 | 5007 | 3 Jọn | 15 | 479 |
| Ezikiel | 1321 | 53862 | Jud | 26 | 995 |
| Daniel | 369 | 15264 | Mkpughe | 426 | 16506 |
| Hosịa | 211 | 7101 | | | |
| Joel | 76 | 2751 | | | |
| Emos | 155 | 5928 | | | |
| Obedaya | 22 | 843 | | | |
| Jona | 52 | 1775 | | | |
| Maịka | 112 | 4233 | | | |
| Nehum | 50 | 1805 | | | |
| Habakok | 59 | 2202 | | | |
| Zefanaya | 56 | 2114 | | | |
| Hakai | 40 | 1591 | | | |
| Zakaraya | 225 | 8581 | | | |
| Malakai | 59 | 2631 | | | |
| **Total** | 24198 | 805899 | **Total** | 8219 | 264795 |
| Bible size | Sentences:32417 | | | | Tokens:1070694 |
| Novel Size | Sentences:2032 | | | | Tokens:40039 |
| **Overall Total** | Sentences:34451 | | | | Tokens:1110733 |

Table 5.4: Corpus statistics after above tokenization method

The two tokenization approaches (whitespace and punctuations) are not proper for Igbo given the issues stated in section 5.3.2. This led to the tokenization method in the same section. Table 5.4 listed the number of tokens in each book; New and Old testaments Bible and the novel as produced by this tokenization algorithm. The statistics from this table (5.4) reveals that the tokenization algorithm for Igbo gave a total of 1070694 tokens, which is 221184 tokens greater in number compared to *whitespace* tokenization and 95236 tokens less than the number of tokens produced by the *whitespace and punctiation* methods. This implies that the 95236 tokens are wrong tokens due to tokenization of diacritics (combining grave accent ( ̀ ), combining acute accent ( ́ ), combining dot below ( ̣)) as separate tokens, which suppose to be part of a whole token. Also, the following word types were calculated for Old testament books: 12747, new testament books: 6424

and novel: 3122.

The tokenization process in section 5.3.2 gave overall sizes of 1110733[6] tokens and 34451 sentences for Igbo corpus (IgbC). IgbC has two major parts; the Bible comprising 1070694[7] tokens and 32417 sentences, and the novel comprising of 40039[8] tokens and 2032 sentences. The Bible represents the religious Igbo texts (IgbBT) while the novel represents the modern Igbo texts (IgbMT). The IgbBT is made up of Old (IgbOT) and New testaments (IgbNT): IgbOT is 805899[9] tokens and 24198 sentences, and IgbNT is 264795[10] tokens and 8219 sentences in size. This is the first available corpus developed for Igbo language to the best of our knowledge.

## 5.4 Analysis of Corpora Used for Experiment

Considering time as constraints, we used only the new testament and the novel in the subsequent study. The followings are some statistics from the both corpora. Each corpora is split into 10 folds and the average statistics are given in table 5.5. Table 5.6 displays the most frequent and less frequent tokens in the both genres. The less frequent tokens are all inflected tokens whose meanings are more than one lexical units in English, and they are what form the majority of unknown words[11].

| | Overall | UnKnown | known | Sentences |
|---|---|---|---|---|
| IgbNT[a] | 263856 | 313 | 26073 | 8219 |
| IgbMT[b] | 39960 | 196 | 3800 | 2032 |

[a]New Testament texts of Igbo corpus from NWT Bible
[b]Modern texts Igbo corpus from the Novel

Table 5.5: Average known, unknown and overall tokens/sentences in Igbo corpora

## 5.5 Conclusion

This chapter discusses how we developed the Igbo corpus (IgbC) from the selected electronic texts. We considered homogeneous collection of texts as the right approach than rather crawling the web in order to reduce the amount of noise through dialectal variation. It is necessary at this stage because we want to avoid as much as possible

[6]Due to issues stated in footnotes 8 and 10, this size now is 1109715.

[7]Due to issues stated in footnotes 10, this size now is 1069755.

[8] During initial tagging and analysis phase in chapter 6, this size was reduced to 39960 through normalization of wrongly written words. Mere looking at these words, they appear to be in the right forms of Igbo words but in practical orthography they are to be written as one. Major issue they posed during tagging was deciding the right class they are to belong, they make no sense standing as one token in any context they appeared. Example is *ọbụla* "any" instead of *ọ bụla* in the text. See other examples in section 6.2.1 of chapter 6.

[9] Issue stated in footnote 8 resulted in its size reduced to 803527.

[10] Issue stated in footnote 8 resulted in its size reduced to 263856.

[11]Unknown words arise from the previously unseen words in the data for for training a system but are in the held-out data set apart for testing, this is constructed using 10-fold cross validation over corpus size.

| IgbNT | | IgbMT | |
|---|---|---|---|
| More Frequent | | | |
| Token | Frequency | Token | Frequency |
| , | 16920 | . | 2002 |
| n’ | 10313 | , | 1629 |
| . | 7404 | ha | 1460 |
| na- | 7228 | na | 1352 |
| na | 6941 | n’ | 1336 |
| ya | 6453 | na- | 1059 |
| ndị | 5404 | ya | 972 |
| ka | 4963 | a | 880 |
| ha | 4621 | bụ | 676 |
| ọ | 4431 | ọ | 631 |
| Less Frequent | | | |
| emesokwa | 1 | nwudere | 1 |
| Ònyekwa | 1 | rịgo | 1 |
| gbazie | 1 | gwupụtara | 1 |
| guzogidenụ | 1 | edoghị | 1 |
| kpọghị | 1 | chịlịchara | 1 |
| dọlitere | 1 | gaghariịrị | 1 |
| enwela | 1 | gbagburu | 1 |
| eleruru | 1 | tachaa | 1 |
| esichabeghị | 1 | emere | 1 |
| atụnyere | 1 | chetaghị | 1 |

Table 5.6: Top 10 most frequent tokens and top 10 less frequent tokens in Igbo corpus (IgbC)

cases such as mixing up text styles and dialects (30 dialects in Igbo) that will introduce errors into the system. Through preprocessing stages of NLP discussed in chapter 4, we created a corpus for the language. Challenges encountered and proffered solutions were also discussed in this chapter. There are two different corpora used based on different text styles: one represents religious texts and the other represents modern texts. The general statistics of both is also presented. The modern texts is recently written compared to the Bible we used to represent the religious texts.

# Chapter 6

# Linguistic Materials

We outline this chapter based on the flow of data development work. Firstly, Part-of-speech (POS) annotation scheme usually called tagset was designed and developed. And this initial tagset was used to perform initial linguistic annotation on the Igbo corpus. Finally, as a result of errors found on this initial annotated corpus through analysis, a revision of the tagset was initiated by performing Inter-Annotation Agreement (IAA) exercise[1].

## 6.1 Creating Linguistic Class: Tagset

Evidently, a well-designed tagset is the first requirement for manual or automatic POS annotation of any language under consideration. We discuss in this section a POS annotation scheme (tagset) designed and developed for Igbo, its transfer onto Igbo corpus, and a revision of it through performing IAA. The tagset design followed 5 higher-level goals:

1. Encode key linguistic distinctions, taking into account typological peculiarities of the language.
2. Allowing linguistic hypotheses to be evaluated by search over POS patterns. For example, evaluating whether all verbs in Igbo normally go with inherent complements.
3. Automatic tagging based on the tagset should deliver high-accuracy performance.
4. Key lexico-grammatical distinctions should be good for advanced NLP processing tasks like parsing.
5. Capture all the inflected and non-inflected tokens in Igbo texts for further analysis like computational morphology. About inflected and non-inflected tokens, a stem/root in Igbo can produce many variants of words through affixation. For example, the stem *bịa* "come" has can form these variants *bịara, bịakwa, bịakwaghị, bịaghịkwa, bịaghachikwara* with affixes.

We adopted the Leech (1997) definition of a POS tagset as a set of word categories to be applied to the tokens of a text. The tagset was designed following the standard

[1]We acknowledge the contribution of Dr. Chinedu Uchechukwu, a senior Igbo linguist in Nnamdi Azikiwe University, Nigeria, who provided valuable inputs throughout the development of Igbo Tagset.

EAGLES guidelines, diverging where necessary (e.g. EAGLES, which favours the European languages, specifies articles at the obligatory level, but this category does not apply for Igbo). A crucial question in tagset design is the extent of fine-grained distinctions to encode within the tagset. Too course-grained a tagset may fail to capture distinctions that would be valuable for subsequent analysis, e.g. syntactic parsing. Too fine-grained a tagset may make automatic (and manual) POS tagging difficult, resulting in errors that lead to different problems for later processing. In what follows, we introduce three granularities of tagset, of which the medium-grained tagset is intended to provide a basis for practical POS tagging, along with both coarse and fine-grained tagsets, that provide views of the data at alternative levels of detail. In an example from Atwell (2008), a tagset in English might try to divide adjective into attributive and predictive adjective, which implies that taggers will on English adjective have more than one tag to choose from depending on context. This makes the task of disambiguating adjective non-trivial. As a guide for developers of taggers, avoiding any distinction that will cause computational difficulties against taggers' performance is necessary. Therefore, considering external and internal criteria necessary for this tagset design, more attention was given to capturing key linguistic features of the language that will be computational helpful for practical purposes.

The medium-grained tagset is intended to strike an appropriate balance for practical purposes, in regard to granularity, capturing what we believe will be the key lexico-grammatical distinctions that will be of value for subsequent processing, such as parsing. The tagset schemes includes 70[2] tags, which apply to entire tokens (as produced by the tokenization algorithm in chapter 5), and is the tagset used in the manual tagging work described below, fine-grained is sized 85 tags, and coarse-grained is sized 15 tags. The fine-grained tagset comprises the 70 tags of medium grained and 30 tags when we went further to find paradigmatic tags of Igbo extensional affixes. See figures A.1 and A.2 on page 149 and on page 150 for details.

### 6.1.1 Design Stages

The tagset design began with Emenanjo (1978) 7 descriptive grammatical classes, viz; verbal [V], nominal [N], nominal modifier [NM], conjunction [CJN], preposition [PREP], suffixes [SUF] and enclitics [ENC], which he found these categories to be convenient and economical to set up POS in Igbo. Therefore, he defined them as follows:

1. **Verbal [V]** takes affixes, especially inflectional suffixes and it is only POS that requires a complement or bound cognate noun to be complete and meaningful.

2. **Nominal [N]**: The following functions are used to identified. (1) it can be used as minimal noun phrase, (2) it can be used as the head of two-word noun phrase, and (3) it can be used as the word immediately following a verb.

[2] We started with 59 tags of the initial tagset called IgbTS0 (stage 4 of figure A.1 in the appendix). It was used to perform initial annotation of New testament Bible texts (IgbNT), which is part of Igbo corpus (IgbC). This exercise gave the first tagged Igbo corpus, which we called IgbTNT0. We performed this exercise before Inter-Annotation Agreement (IAA) process, and method used is discussed in section 6.2.1. Analysis of IgbTNT0 led to revision of the 59 tags of IgbTS0 tagset that gave rise to 70-tag tagset. This 70-tag tagset is also used to tag modern texts genre (IgbMT). Reasons for tagset revision is also in section 6.2.1.

3. **Nominal modifier** [**NM**] always occur in a noun phrase and cannot be used alone or as the head of a two-word noun phrase.

4. **Conjunction** [**CJN**] only links words or sentences together in the language.

5. **Preposition** [**PREP**] is found preceding nominals and verbals and cannot be found in isolation.

6. **Suffixes** [**SUF**] are only bound elements in the language and primarily affixed to only verbals. Suffixes are found in the verb phrase slots. **Enclitics** [**ENC**] are used with both verbals and other word classes. It can be found in both verb phrase and noun phrase slots. They are joined to the verbs, if found immediately after a verb without any intervening words, otherwise they stand alone in noun phrases. See examples in table 2.6 of chapter 2.

At the recommendation level of EAGLES, the major classes are listed with their attributes, which on standard requirement, EAGLES states,

> "if they (attributes) occur in a particular language, then it is advisable that the tagset of that language should encode them."

We adopted values of the major classes that are found in compliance with Igbo, and included ones that are not found. For example, in EAGLES noun class, there are four values, viz; type, gender, number, and case. We adapted only type, that is, proper and common nouns. The rest are not applicable to the language. The language has plural modifiers that if following a noun will indicate that noun to be plural but not changing the form of the noun as in English (e.g. *nwoke* "boy", *ụmụ nwoke* "boys"). Also, we split pronouns and determiners 'super-category' of EAGLES into different classes since they have distinctive role they play in the language. Two words in Igbo are found to have characters of a determiner and they are classified under nominal modifier in Emenanjo (1978) as demonstratives. So, we chose demonstrative over determiner since determiner can have various functions such as subsuming articles and there is no article in Igbo.

The *seven classes* (7 core tags) defined above were used for a start-up at the initial stage. Emenanjo (1978) classifies the nominal class in Igbo into nouns, numerals, pronouns and interrogatives, and further simplify nouns into proper, common, qualificative, adverbial and ideophones. Consequently, the *7 core tags* were decomposed into *15 simple tags*, and from 15 to further *25 simpler tags*. In practice, changes to capture key grammatical categories in a tagset is advisable while maintaining an optimal size for machine learning purposes. For example, in preliminary stage of automatic POS tagging of LOB corpus using Tagged Brown Corpus, Leech et al. (1983) made some important tag changes that resulted in producing 134 tags against Brown's 87 tags. An example of such change is Brown's single proper noun (NP) was decomposed into NPL, NPT, NNP, JNP. Similarly, some of the *25 tags* were further decomposed into more simpler tags by studying and analyzing about 23% of Igbo corpus developed in chapter 5 and using the attributes of the major classes of the EAGLES (diverging where necessary). Some examples are decomposing common nouns into multiwords nouns which comprises agentive and instrumental nouns, and conjunction into correlation conjunctions. Thus, we have common nouns (NNC) and link-pair common nouns that have two multiwords units (NNAV ... NNAC and NNTV ...

NNTC), and conjunctions (CJN) and link-pair conjunctions that show correlations (CNJ1 ... CJN2). This process led to the *medium tagset of 59*[3] tags (see stage four in appendix A.1), which consists of 10 noun classes, 10 verbals, 2 inflectional classes, 15 POS classes with any affixation (XS), 1 enclitic, and 21 other POS classes.

*XS* is an extensional suffix marker attached to some tags for identification of words that are inflected by prefixes and/or suffixes and/or enclitics. In Igbo, suffixes and enclitics have grammatical roles they play (Emenanjo, 1978), which are subsumed into *XS*. Through the identification of the grammatical roles, *XS* was simplified into 30 morph-tags leading to *fine-grained tagset* (stage 5 of appendix A.1). The *fine-grained tagset* includes the 30 morph-tags, 1 ENC and all other tags in the *medium-grained tagset* except the 15 POS classes with *XS*.

Table 6.2 tagset shows a *coarse-grained* tagset of just 15 labels, onto which the 59 tags of the medium tagset can be mapped down. The *coarse-grained* tagset principally preserve just the core distinction between word classes, such as nouns, verb, adjective, etc, although the distinction between proper nouns (NNP) and the other 9 noun categories was preserved (which reduce to a single common noun tag NNC). The coarse-grained tagset is intende to be for the benefit of cross-lingual training and other NLP tasks such as unsupervised induction of syntactic structure and multilingual POS tags projection. Compared with the universal tagset of Petrov et al. (2011), our tagset does not have *article* which is one of the universal tagset tag. Demonstrative class *DEM* that plays major role at any level in Igbo is not among the tags of the universal tagset. See figures A.1 and A.2 on page 149 and on page 150 for details.

| | | | |
|---|---|---|---|
| NNM | Number marking nouns | BPRN | Bound Pronoun |
| NNQ | Qualificative nouns | VrV | $-rV$ implies suffix for inflectional class |
| NND | Adverbial nouns | VCJ | Conjunctional verbs |
| NNH | Inherent complement nouns | $\alpha$_XS | any POS tag with affixes |
| NNCV | Verb part of multiword noun | NNCC | Noun complement part of multiword noun |
| EX$N$ | All extensional suffixes, where $N$ is given name based on grammatical functions | EN$X$ | All enclitics, where $X$ is given name based on grammatical functions |

Table 6.1: A selection of distinctive tags of the medium size and fine-grained tagset

| | | |
|---|---|---|
| ADJ adjective | FW foreign word | QTF quantifier |
| ADV adverb | INTJ interjection | SYM symbol |
| CJN conjunction | NNC common noun | WH interrogative |
| PRN pronoun | NNP proper noun | V verb |
| CD number | PREP preposition | DEM demonstration |

Table 6.2: Coarse-grain Tagset

There are decisions taken and challenges encountered during these stages of tagset design and development. Each was handled following Igbo linguistic literatures and discussions with Igbo linguistic expert where necessary. For instance, when we moved

[3]See footnote 2.

from stage 2 to 3, we combined pronouns and pronominal modifiers to PRN since both possess the category of persons (1st, 2nd and 3rd), and share identical forms. We dropped suffix and took enclitic to stage 3 because, according to Emenanjo (1978), suffixes are principally attached to verbals, and can be found only on verbs in the verb's slot, while enclitics can be attached to verbal and other part-of-speech (POS), and can be found after Nominals in NP slots. Also, enclitics are attached to a verb if found immediately after a verb but stand on their own if found before a verb. This implies that we can find enclitics that are not attached to verbs or other nominals since we are interested on whitespace lexically separated words at this level. This intuition was proven when we found enclitic words like *kwa, nnọọ* in some sentences, which on their on are meaningless.

We identified OVS tag as open vowel suffix for only vowel inflected words like *lee, laa, abọọ*. Introduction of extensional affix tag *XS* to represent any inflected tag at stage 4 cancelled the use of OVS. Any *inflected tag* (*tag_XS*) can be caused by the presence of either suffixes, enclitics, open vowel suffix or tense inflection. Therefore, a token assigned *OVS* will conflict with *OVS_XS* since *OVS* is part of the elements that determine the presence of *XS* in a word form. For instance, if we want to find if a word has affix(es) *XS*, we look at the word analytically to find its stem and other forms (e.g.: *lee* is inflected by *OVS* "e").

For the case of adverbs [ADV], Emenanjo (1978) identified only adverbial nouns (NND), which he stated "they may be found elsewhere in the sentence, ... and always function as *emphasizers* of verbals and translate as adverbs in English." E.g. the Igbo sentence, *O ji nwayọọ eri nri ya* will be directly translated as

O ji nwayọọ eri nri ya.
He holds slowly eat food his.
"He eats his food slowly."

The adverb *nwayọọ* is found in a noun slot. The verb *ji* always precedes a noun in Igbo grammer, likewise the verbs *dụ* and *bụ* (Emenanjo, 1978). Here, one can classify *nwayọọ* as NND. But there are cases where they are found outside noun slots, therefore, we decided to classify them as adverbs (in stage 3 of figure A.1).

Another challenge is the case of *ideophones*. Athough Emenanjo (1978) classified them as a form of noun, we have assigned them a separate tag **IDEO**, as these items can be found performing many grammatical functions. For instance, the **ideophone** *kọị*, "to say that someone walks *kọị kọị*" has no nominal meaning, rather its function here is adverbial.

An important challenge comes from the complex morphological behaviour of Igbo. Thus, a verb such as *bịa*, which we assign tag VSI (for a verb in its simple or base form), can combine with extensional suffixes, such as *ghị* and *kwa*, to produce variants such as *bịaghị*, *bịakwa* and *bịaghịkwa*, which exhibit similar grammatical behaviour to the base form. As such, we might have assigned these variants the VSI tag also, but have instead chosen to assign tag VSI_XS, which serves to indicate both the core grammatical behaviour and the presence of extensional suffixes. In *abịakwa*, we find the same base form *bịa*, plus a verbal vowel prefix *a*, which results in the verb being a participle, which we assign tag VPP_XS.

The fine-grained tagset goes beyond assigning tags only to full tokens, and instead assigns tags to the individual morphemes within words, to characterise their lexico-grammatical behaviour. For example, abịakwa would be analysed as *a/VVP + bịa/VSI +*

*kwa/ENADV*, with VVP identifying the verbal vowel prefix, VSI the simple verb root, and ENADV an enclitic with additive function. The word *bịaghịkwa* would be analysed as *bịa/VSI + ghị/EXNEG + kwa/ENADV*, where EXNEG marks an extensional suffix denoting negation (see figure A.1). Clearly, practical use of this scheme requires automated morphological segmentation of Igbo, which requires further investigation, but we believe the specification of this scheme is a valuable step in this direction. A selection of the tags of this scheme, that are not typically found in other tagsets, are shown in table 6.1. A full enumeration of the scheme is given in the appendix A.

## 6.2 Linguistic Annotation

Geoffrey (2004) highlights that *rules or guidelines for assigning particular annotation devices to particular stretches of text* as the most important in specification of annotation practices. This document, which originate from sets of guidelines which evolve in the process of annotating a corpus, is needed to explain the annotation scheme to the users of an annotated corpus. Tagset we have created will be used in this section for the purpose of enriching the tagset associated guideline, and developing initial tagged Igbo corpus. We will refer this tagset as IgbTS0.

### 6.2.1 Tagset and Associated Guideline

Developing a new tagset for any language usually presents the problem of how to express information regarding challenging phenomena in the tagging guideline, especially as regards to the language internal criteria. Thus, we embarked on an initial annotation task, which is a preliminary investigation on how to use IgbTS0, in order to identify and design the IgbTS0 tagging guideline that will include information regarding challenging phenomena of the language (Hardie, 2003; Bamba Dione et al., 2010). This annotation process was a shared task among six human annotators, which led to the first ever POS tagged Igbo corpus. Henceforth, we will be referring to Igbo corpus as IgbC, Igbo initial tagset (IgbTS0), New testament part of IgbC as IgbNT and tagged New testament part of IgbC as (IgbTNT).

We used *Microsoft Excel office worksheet* as the tagging workbench. This is an alternative annotation platform (to more sophisticated ones, such as Gate[4]) recommended in areas where there is poor access to Internet. Internet access in the Eastern Nigeria is poor, and the cost of maintaining one is very high. There are other factors that motivated the choice of Microsoft Excel: It is available, accessible, and proficient. The number of *Microsoft Office* application users in the Eastern Nigeria is high compared to other software applications. In addition, Microsoft Excel has good features for data analysis, and the output can be formatted in various forms such as XML or text.

Illustrating figure 6.1, the Excel environment for POS tagging task is designed such that the selections of suitable tags for tokens are done sententially. Each row in the Excel worksheet contains a token $w$ (cells on column **A**), five most common tags for $w$ (cells on columns **C** through **G**), and a combo box of all tags in the Igbo tagset (cells on column

[4]We tried Gate architecture (Cunningham et al., 2002), but our collaborators in the eastern Nigeria were unable to download it due to poor Internet access.

| | |
|---|---|
| Group 1 | *Matthew, Phelimon, 2 Peter, 1 Timothy, 1 Peter* |
| Group 2 | *Acts, 2 Corinthians* |
| Group 3 | *Mark, Revelation, Galatians, 3 John, 2 John* |
| Group 4 | *John, Philipians, James, Colossians, 1 John, 1 Thessalonians* |
| Group 5 | *Luke, Ephessians, 2 Thessalonians, Titus* |
| Group 6 | *Romans, Hebrew, 1 Corinthians, 2 Timothy, Jude* |

Table 6.3: IgbNT Bible Book Selections by Group for POS annotation

**H**). For each $w$ in cell **A**, the user scans cells **C** through **G** and select a suitable tag for $w$ by clicking the cell. If there is no suitable tag for $w$ in the five most common tags, the user uses the combo box. The tag selected for $w$ will then appear on cell **B** immediately adjacent to $w$'s cell **A**.

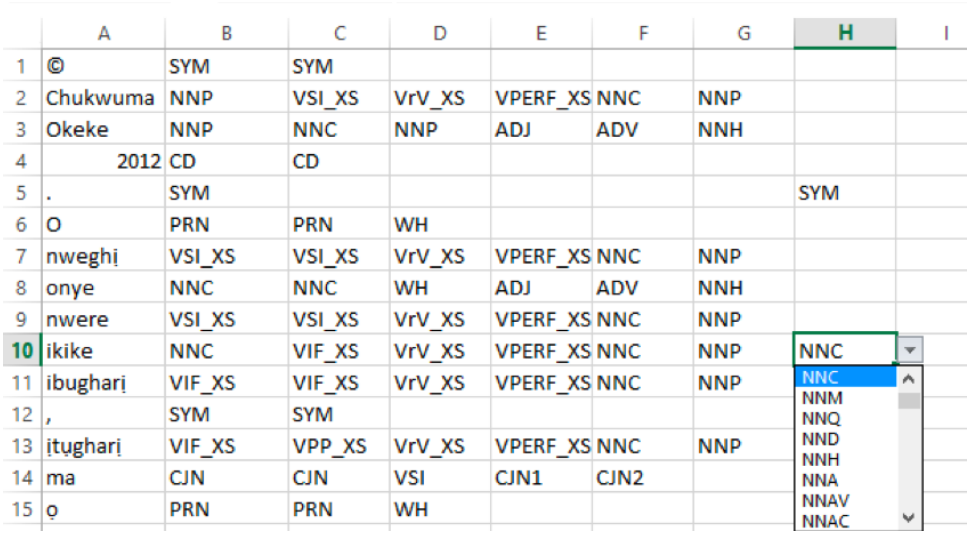



Figure 6.1: The Excel worksheet panel for Igbo POS annotation (Column B is for selected tags)

The Igbo language resources used are the New World Translation Bible[5] (NWT) of Igbo corpus and IgbTS0. We collected the New Testament portion (Henceforth IgbNT), which is ≈260000 tokens and 8000 sentences. For the purpose of rapid POS tagging, chapters in the IgbNT were allocated randomly to six groups, producing six corpora portions of approximately 43,000 tokens each (See table 6.3). To ensure quality, annotators are graduates of the Department of Linguistics at Nnamdi Azikiwe University, Nigeria, and a supervisor who is a senior lecturer in the same department; giving a total of seven human annotators. Their work is to use the IgbTS0 and IgbNT annotation materials we provided to produce POS tagged corpus.

Our plan was for each human annotator to tag at least 1000 tokens per day, resulting in complete POS tagging in ≤43 days. The overall corpus size allocated is 264795[6] tokens

[5]Obtained from `jw.org`.

[6]This is the New testament texts (IgbNT) produced by tokenization in section 5.3.2 of chapter 6 before the clean up exercise in section 6.2.1 of chapter 6 that led to 263856 tokens size.

| Token id | Token | Error | Resolved | Total types |
|---|---|---|---|---|
| 12291 | ahụkwa | DEM/DEMXS | DEM_XS | 138 |
| 4 | nke | CJN/* | CJN | |
| 26189 | mkpịrịkpị | QTF/XXXX | NNQ | |
| 59639 | mpịakọta | NOTAG | NNC | 156 |
| 1717 | wit | XXXX | NNC | |
| 58325 | bụla | NOTAG | ọbụla/QTF | 941 |
| 194197 | àghọwọ | NOTAG | à/PRNYNQ ghọwọ/VPERF | 1 |
| 11790 | ee | INT | INTJ | 3827 |
| 815 | chọọ | vSI_OVS | VSI_XS | |
| 1073 | fụọ | VSI_OVS | VSI_XS | |
| 3537 | nwee | OVS | VSI_XS | |
| 7 | banyere | VRV_XS | VrV_XS | |

Table 6.4: Different error types encountered during cleaning of initial annotation, and corrections provided

of the IgbNT. Each annotator annotates one group separately. This annotation process produced IgbTNT0– the first ever tagged Igbo texts.

During this initial annotation, relationships that exists between tags and difficult expressions identified by annotators were discussed, and the outcomes of the discussions were added to the tagset as tagging guidelines. For example, verbal complex structure of the language contains two or more parts, namely; verbal and noun inherent complement(s) constituents. These parts can occur adjacent to each other without intervening words between them in a sentence. Also, they can occur with one or more words between them in a sentence. Example of practical illustration of the verbal complex is in figure A.3 of appendix A.2.2. Illustrative examples concerning this verbal complex using the sentences: *ọ bụ ihe ịtụ n'anya* "it is a surprising thing" and *ọ tụrụ mụ n'anya* "it surprised me", "anya" in both cases is a *noun inherent complement* to the verbs "ịtụ" (infinitive) and "tụrụ" (past tense). The process of manual implementation of tags enables us to establish suitable interpretations to the relationships that exists between the tags associated with this verbal complex in our tagset.

### Cleaning Up the Initial Annotation

Given the six POS tagged sub-corpora, we collected the best samples and eliminated errors found in the process. In most cases, this process is indistinguishable from "editing". The types of errors found are *unspecific tag* where annotators could not apply a specific tag to a particular token (1st row of table 6.4), *no tag* where tokens are not classified by annotators (2nd and 3rd rows of table 6.4), and *wrong form* where valid tags are wrongly represented (4th row of table 6.4). Total number of tags and tokens affected by these errors are 39 and 5062, which is 1.92% of IgbTNT0. Proper consultations were made to resolve errors in the *unspecific tag* and *no tag* sets. In solving the remainder, we built a tag replacement dictionary of the errors in the *wrong form* class, and pass the IgbTNT0 through it to produce IgbTNT1. The tag replacement dictionary is represented as tag_replacement = {'INT':'INTJ', 'VSI_OVS': 'VSI_XS', ...}.

One of the main issues that caused *no tag* error was improper word form. For example, the token *bụla* is incomplete without *ọ*; in the Bible, both were separated by a lexical space *ọ bụla* 'any'. If annotators had assigned *ọ* with a tag 'PRN' (since it has pronoun form), identifying the right tag for *bụla* became challenging since its meaning is incomplete. This was fixed by removing the lexical space between them. The IgbNT size which was originally 264,795, after initial tokenization, reduced to 263,856 after fixing all the errors. Table 6.4 shows a few examples of tokens affected and solutions provided.

## 6.3 Tagset and Annotation Improvement Process

We started this section with a preliminary experiment to determine the extent of learnability of taggers on the corpus annotation done in the previous section. We performed automatic POS tagging on IgbTNT0 and IgbTNT1 using Transformation-Based Learning in a Fast Lane (FnTBL) by Ngai and Florian (2001). This tagging was done using 10 fold cross validation on a crossed vocabulary. The average results are in table 6.5. Obviously, we expected poor performance because of the way annotation task was done.

| IgbTNT0 | IgbTNT1 |
|---|---|
| 88.22 | 90.17 |

Table 6.5: Average results of simple accuracy on 10-fold evaluation on IgbTNT0 and IgbTNT1

### 6.3.1 Tagset Evaluation

Evaluation results on IgbTNT0 and IgbTNT1 showed that the cleaning up exercise increased the accuracy by 1.95%. The accuracy score of 90.17% indicate poor performance of FnTBL on the tagged corpus, IgbTNT1. This led to further investigation on IgbTNT1 and IgbTS0, and the findings of our investigation are hereby submitted below.

#### Tagset Revision

In this section, we evaluated the 59-tag tagset (and the associated guideline) in order to measure its validity and reproducibility. When texts and human judgements are stored in computer-readable form, the result is called annotation. Annotation is developed mostly through hand-coded means by human speakers, so it is important to measure the reliability of the tagset (and the associated guideline) that produced it. Since annotations correspond to human-coders' judgements, there is no objective way of establishing the validity of an annotation. Instead, reliability is measured by verifying if human annotators are consistently making the same decisions using the same guideline. High reliability is a prerequisite for validity. Since several human annotators use the same texts with the same guidelines provided, then their IAA is calculated (Fernández, 2011).

Despite the use of human annotators with good knowledge of Igbo linguistics in the previous section to perform annotation, our investigation revealed that there are factors that motivated the revision of IgbTS0 in order to maximize human annotators agreement,

and to ensure Igbo tagset is valid and reproducible. The confusing factors we found among human annotators were related to the status of what to call participles, agentive/instrumental nouns, preposition, etc. For example, annotators had issue classifying some verbs when they change their structures as they precede or follow a pronoun. Mostly they chose to tag them participle (VPP) because the changed structure is prefixed *a/e*, which makes them look like participles. The worst case we found was the handling of the nominal class formed through verb nominalization. There are agentive and instrumental nouns represented in tag as NNAV NNAC and NNTV NNTC respectively, where V and C are the verbal and inherent noun components of the structure which should always appear as a linked pair. For example, *ọgụ/NNAV egwu/NNAC* "singer" and *ngwu/NNTV ji/NNTC* "digger", but link pairs like *ntachi obi* "steadfastness", *nnwere onwe* "freedom", etc are neither agentive nor instrumental nouns.

The main objective we assigned to ourselves while revising the tagged corpus and tagset was to get high quality tagged corpus, get a specific tagset appropriate for Igbo, and to maximize agreement among human annotators in order to ensure high consistency of the tagged corpus. However, agreement among human annotators is not a guarantee for tagset quality, otherwise the trivial and uninformative tagset of one tag "WORD" that will *onyl* identify words would be optimal. In our task, most *meaning-bearing* words were assigned POS tags based on the grammatical role they play in a sentence. Nevertheless, the more informative a tagset is, the less the taggers' (human and automatic) accuracy tends to be (Atwell, 2008). Therefore, one has to know where to strike a balance between the tagset informativeness and tagger performance. These and many other reasons led to evaluation and revision of IgbTS0 through IAA exercise.

### Inter-Annotation Agreement (IAA)

We used five human annotators that are linguists and Igbo native speakers, the excel platform in figure 6.1 for annotation, the New Testament Bible corpus and tagset (and associated guideline) discussed in the above sections. The tagset serves as a model (M) for human annotators to use on the corpus. It is formatted into M = *[T,R,I]*, where T = POS tags, R = relationships between T, and I = tags interpretations on usage. We adopted Pustejovsky and Stubbs (2012) NLP annotation development cycle methodology. It involves *Model* → *Annotate* → *Evaluate* → *Revise* (M-A-E-R) cycle. We iteratively applied this *M-A-E-R* cycle, until all tags contributing huge disagreements in the annotations are corrected resulting in a higher consistency level among annotators. In each phase, the annotations *-A-* by annotators were done independently using our *M-*. At the end of each phase, we collect all annotations and apply *-ER* (Evaluate and Revise).

The IAA process took three iterative phases. In each phase, a subset (about 4.5k tokens) of New Testament Bible corpus was randomly selected (see table 6.6). The tagging guideline used was evaluated and revised at each phase. Since there are 5 human annotators ($l1, l2, l3, l4, l5$, where $l$ = linguist), each phase produced 5 annotations of the selected texts, and from these annotated texts we collate standard outputs through voting. That is, for each token, we consider tags with the highest agreement among annotators. For example, "IDEO" tag was chosen for the token *gbaa* since *gbaa* was assigned "IDEO" by 4 annotators and "VSI_XS" by one annotator. We ignored any instance where there was total disagreement between annotators including some special cases where two annotators

| | first IAA | second IAA | third IAA |
|---|---|---|---|
| # of sentences | 150 | 150 | 150 |
| # of tokens | 4977 | 4963 | 4851 |

Table 6.6: IAA texts statistics selected from the New testament Bible corpus (IgbNT)

agreed on a tag "A", two others agreed on another tag "B" and remainder chose a different tag "C". This is to ensure fairness in our judgement and to make certain that tags with high confidence rate are chosen. Though agreement among annotators is not a guarantee for quality assurance, we based our confidence on the high profiled human annotators we used. We take the collated outputs as our presumed truths, which serves as "silver standard" against which individual annotators are compared. The quality of the silver standard is determined by the annotators' tagging consistency calculated using IAA metrics as discussed in the next paragraph. Performance was evaluated using $f$-measure, simple accuracy method ($SAM$) and Cohn's kappas ($CK$). Our experiment assumed that each token is fully disambiguated, that is, one tag for one token.

In computing agreement, we used $f$-measure metric to provide a more detailed picture of IAA between annotators on individual parts-of-speech (POS) tag. The $f$-measure relates to precision and recall in the usual way. For each phase, we find the micro-averaged precision and recall, then calculate $f$-measure. In more detail, for each of the five annotators, we calculate each tag's precision and recall relative to the collated silver standard. While $CK$ helped us to evaluate human annotators' consistency by computing their overall agreement scores (observed and expected chance agreements), $SAM$ helped us to detect "bad" annotators or annotations by computing annotators' observed agreement. For $CK$, we compute observed agreement $A_o$ among annotators, expected chance agreement $A_e$ among annotators, how much agreement beyond chance was found $A_o$–$A_e$, and how much agreement beyond chance is attainable $1$–$A_e$ (Fernández, 2011). $A_o$ measures the number of tags on which annotators agree divided by total number of tags, but does not take into account agreement that is due to chance. $A_e$ measures how often annotators are expected to agree if they make random choices according to their individual tag distributions. Since the decisions of the annotators are independent, we multiply the marginals. For example, the chance of two annotators ($l1$ and $l2$) agreeing on a tag $t$ is $P(l1|t) \times P(l2|t)$. Therefore, the chance of the annotators agreeing on any tag is computed as $A_e = \sum_{t \in T} P(l1|t) \times P(l2|t)$, where $T$ is set of tags. Therefore, we compute $CK$ as follows:

$$CK = \frac{A_o - A_e}{1 - A_e} \tag{6.1}$$

$CK$ is the proportion of the possible agreement beyond chance that was actually achieved. See figure 6.3 for $CK$ performance scores.

$SAM$ is a measure that compares the IAA annotations' tags to judge whether or not the tags are identical (i.e. finding where annotators disagree or agree on tags). Annotations were done by *l1, l2, l3, l4*, and *l5* annotators using the same text at different IAA phases, which implies that there are five different annotations produced at each phase. Since annotations correspond to human annotators' judgements, we used $SAM$ to observe the annotators that are not consistently making the same decisions using the IAA

exercise guideline. High consistency is needed at the IAA exercise to ensure reliability and validity, therefore annotations that are not consistent with others are targeted here as 'bad' annotation/annotator. *SAM* can be regarded as $A_o$ that measures the number of times on which annotators agree on a tag . Therefore, we compute *SAM* as follows:

$$SAM = \frac{tp}{N} \tag{6.2}$$

where *tp* is the number of times on which annotators agree on a tag (i.e. where they chose the same *tag*), and $N$ is total number of tokens in one annotation. For example, *tp* for a token *tok* means that the tags assigned to *tok* by human annotators are identical when their annotation are compared following the steps: Firstly, we grouped the annotations in different combinations (*2, 3, and 4*) such that $C(5,2)$[7] $= 10$ different combinations (e.g. l1+l2, l5+l2, l5+l1,etc.), $C(5,3) = 10$ different combinations (l3+l4+l5, l3+l4+l1,l3+l5+l2, etc.), and $C(5,4) = 4$ combinations (e.g. l3+l4+l5+l2, l3+l4+l5+l1, l3+l4+l1+l2, and l3+l5+l1+l2). Secondly, for each group of combinations, we take a combination (*say* l3+l4+l5+l2) and compute *SAM* by counting the number of locations in *l3, l4, l5* and *l2* annotations on which annotators agree on a tag. The outcome is divided by the total number of tokens in either of the annotations (e.g. *l1*) since the annotators used the same text[8].

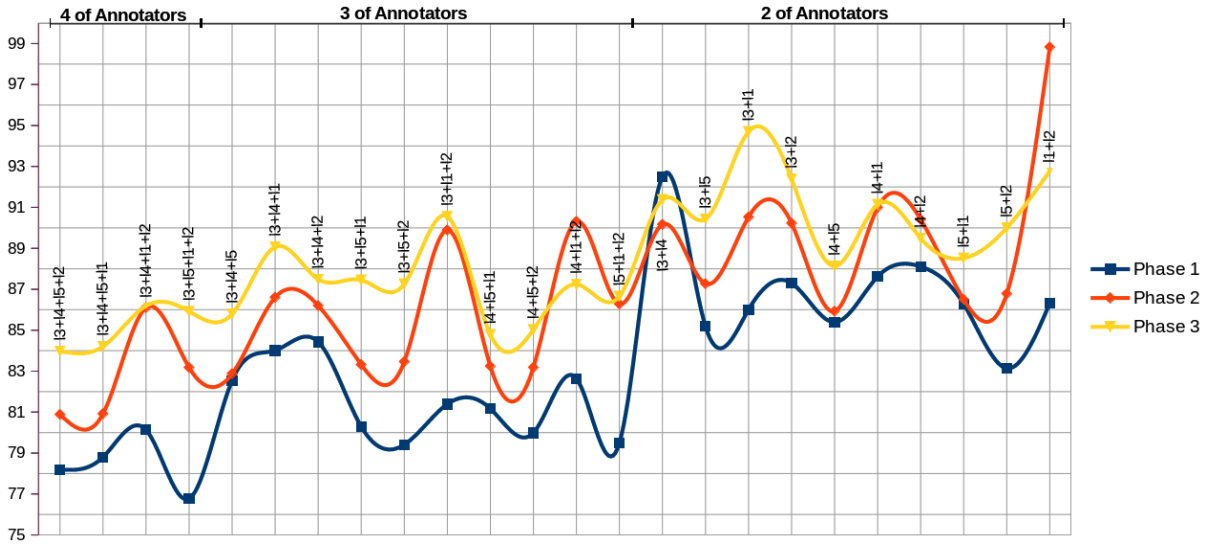



Figure 6.2: Using simple accuracy method (SAM) scores for detecting "bad" annotator or annotation. 4, 3 and 2 annotators on top means different combinations of annotators used. We used annotators that are linguists and Igbo native speakers, hence the symbol *li*, where *i* represent annotator's identity number. The nodes (*l3+l4+l5+l2, l3+l4+l2, l1+l2, and so on*) represent various accuracy scores of different combinations of annotators.

Thus, an annotator that is under-performing can be detected by comparing her annotated texts with others. For example, in figure 6.2, the graph nodes (*l3+l4+l5+l2, l3+l4+l2,*

[7]5 combination 2.

[8]There are three IAA exercise phases (see previous sections), on each phase, the annotators used the same text for their annotations.

*l1+l2, and so on*) represent various accuracy scores of different combinations of annotators. This reveals that tags assigned by annotator *l5* are most likely to disagree with the others. The low points on the graph are mostly when *l5* is combined with other annotators (e.g., *l3+l5+l1+l2*).

### 6.3.2 Discussion

The fundamental assumption of this exercise, as discuss in Artstein and Massimo (2008) and Fernández (2011), is to check if the output of human taggers through the use of the tagset and its tagging guideline is considered reliable. This we evaluated by computing whether annotators are consistent, and the consistency is measured using metrics from the study of Landis and Koch (1977), Krippendorff (1980), and Green (1977). Table 6.7 and figure 6.3 show a cumulative improvement in human annotators' consistency as the IAA exercise was progressed from one phase to another.

| Tag | Precision | | | Recall | | | $f$-measure | | |
|---|---|---|---|---|---|---|---|---|---|
| Tag | 1st | 2nd | 3rd | 1st | 2nd | 3rd | 1st | 2nd | 3rd |
| NNC | 95.40 | 96.16 | 96.65 | 90.62 | 90.04 | 95.11 | 92.31 | 92.45 | 95.36 |
| PRN | 99.03 | 99.70 | 98.10 | 98.22 | 99.52 | 99.06 | 98.12 | 99.11 | 98.07 |
| PREP | 92.89 | 97.07 | 99.00 | 94.39 | 98.60 | 99.06 | 93.09 | 97.32 | 98.53 |
| VPP | 88.47 | 89.17 | 96.62 | 89.51 | 93.13 | 95.24 | 88.04 | 90.13 | 95.33 |
| VSI | 90.01 | 93.10 | 93.11 | 89.43 | 90.02 | 97.49 | 88.39 | 90.90 | 94.71 |
| VIF_XS | 88.96 | 68.43 | 95.49 | 58.46 | 84.38 | 85.00 | 61.13 | 70.84 | 87.41 |
| VPERF | 52.86 | 62.10 | 78.65 | 52.50 | 75.00 | 76.00 | 45.05 | 59.36 | 71.59 |

Table 6.7: Some POS tags precision, recall and $f$-measure of first, second and third phases of Inter-annotation agreement (IAA) exercise

Table 6.7 shows improvements on individual tags by computing precision and recall for each assigned tag. This reveals whether all the tokens that suppose to be assigned tag "NNC" get it and whether all the tokens assigned "NNC" are correct. Figure 6.3 displays Kappa's agreement scores in order to see the consistency level among the human annotators. The scores between annotators are consistently and substantially high, which indicates that the tagging guideline is reliable, therefore, it is valid. Also, the scores indicates that human annotators have internalized a similar understanding of the tagset (and the associated guideline). The outcome of this IAA exercise is high consistency tagged sub-corpora of the Bible corpus containing tags described in the revised tagset. During th IAA exercise, the tagset (and the associated guideline) was revised through evaluation and adjudication of the disagreements found either by tag simplification, removal or addition. This is discuss in the below paragraphs.

There are steps we took to to solve the issues highlighted in the tagset revision (section 6.3.1). First is tag simplification. In the nominal class case, we redefined agentive and instrumental nouns into multiword nouns (NNCV NNCC), so that all tokens in this forms can easily fit into this class (see results in table 6.8). Multiword nouns occur as link pairs, where one (NNCV) is common noun formed through verb nominalization and the other (NNCC) is the inherent complement. For example, nominalization of multiword nouns *ịgụ egwu* "to sing", the verb *ịgụ* changes to *ọgụ*, and carries along its inherent complement

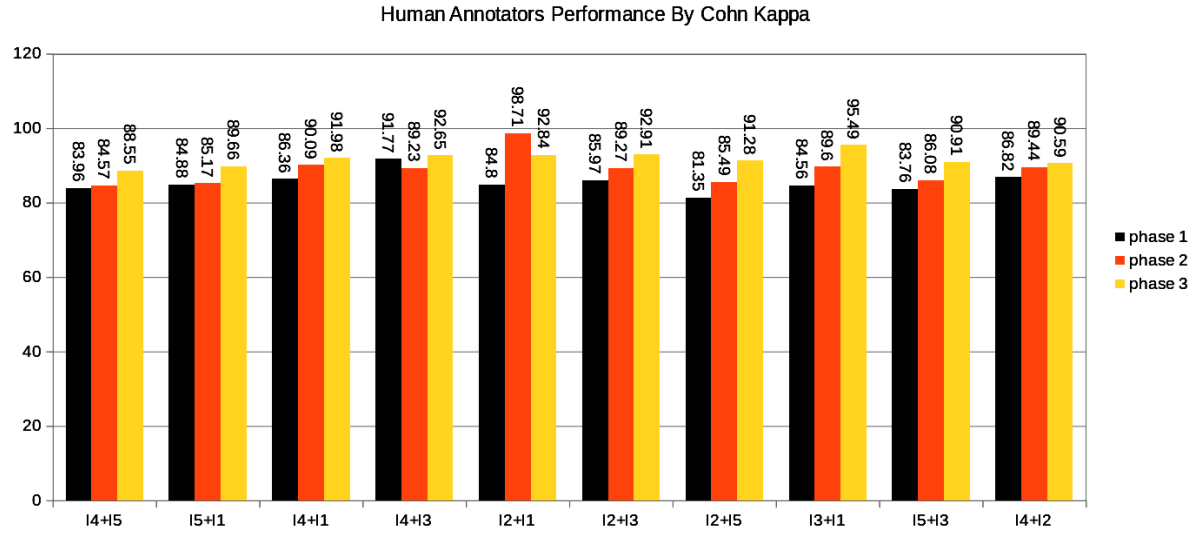



Figure 6.3: Annotators performance improvement (in pairs) in each inter-annotation phase computed using Cohn's Kappa *CK*. Vertical and horizontal coordinates represent annotators in pairs and *CK* scores. Annotators are Igbo linguists and five in number, hence the symbol *li*, where *i* represent annotator's identity number

| Tag | Precision | | | Recall | | | $f$-measure | | |
|---|---|---|---|---|---|---|---|---|---|
| Before collapsing tags to NNCV and NNCC | | | | | | | | | |
| Phases of IAA | 1st | 2nd | 3rd | 1st | 2nd | 3rd | 1st | 2nd | 3rd |
| NNAV | 51.33 | 0.0 | | 80.0 | 0.0 | | 55.52 | 0.0 | |
| NNAC | 0.0 | 0.0 | | 0.0 | 0.0 | | 0.0 | 0.0 | |
| NNTV | 0.0 | 0.0 | | 0.0 | 0.0 | | 0.0 | 0.0 | |
| NNTC | 0.0 | 0.0 | | 0.0 | 0.0 | | 0.0 | 0.0 | |
| After collapsing tags to NNCV and NNCC | | | | | | | | | |
| NNCV | | | 77.81 | | | 73.33 | | | 74.27 |
| NNCC | | | 81.14 | | | 73.33 | | | 75.79 |

Table 6.8: Some WORST POS tags precision, recall, and $f$-measure and solution proffered during IAA exercise

noun *egwu* to become *ọgụ egwu* "singer". Decomposing multiword nouns into agentive (e.g. *ọgụ egwu* "singer") and instrumental nouns (e.g. *oti ịgba* "drummer") creates four extra noun tags increasing noun class size to 10. This is an example of internal criterion design that is linguistically-motivated distinctions. Human taggers struggled with POS tagging multiwords nouns, which is a non trivial tasks. Linguistically-motivated distinctions of multiwords nouns class would have made sense if the purpose of developing this POS tagged corpus is for detailed linguistic study of Igbo language. But we aimed to develop statistical language processing model usually called taggers. Any distinction that will create computational difficulty may inhibit tagger's performance.

Next is *tag addition*. We also introduced $\alpha$_BPRN tags to differentiate between verbs (e.g. participle or simple verbs) that start with a vowel *a/e*. We observe that prefix in a verb can be caused the location of a pronoun on a sentence or if the verb is preceded

by an auxiliary verb. For instance, the word *esi* in *Ọ na- esi nri* "He is cooking " and *esi m Sheffield abịa* "I am coming from Sheffield" functions differently. The first is verb participle (VPP) because of the auxiliary verb *na-*, while the second is a simple verb inflected by a vowel prefix *e* as a result of the position of pronoun *m* in the sentence. Therefore, we introduced *VSI_BPRN* tag to indicate that *e* in *esi* is *m*-bound and *BPRN* tag for *m*-bound. It is assigned *VSI* if sentence pattern changes to *m si Sheffield abịa* "I am coming from Sheffield", while *m* is assign *PRN*. This prevented annotators assigning participle (VPP) to *esi* in the second example. Also, modal verb (VMO) can be found in two parts that requires two different tagging styles. VMO can make sense on its own or requires a noun complement to complete its sense, therefore we introduced VMOV and VMOC for verbal and noun complement parts making modal verb tags three (VMO, VMOV...VMOC). For example, *ị kwesịrị/VMO ikele ya* "you should have greeted him", *o nwere/VMOV ike/VMOC ịgụ akwọkwụ ya* "s/he can read his book". Compare the latter example with *o nwere/V ike/NNC o ji ạgụ akwọkwụ* "S/he has strength to read book" and *o nwere/V akwọkwụ ahụ* "S/he has that book".

The case of multiword unit tags is a problem area in tagset development, Atwell (2008) argued that there is not always a one-to-one mapping between word and tag. It is possible that a word may contain several tags or several words may be assigned one tag. For example, Brown and UPenn tagging schemes treated multiword items (e.g. "as well") as sequence of adverb/qualifier + adverb, Polytechnic of Wales (POW) Corpus tagset chose to provide one tag for such expression, while other tagging schemes (Lancaster-Oslo/Bergen (LOB) and British National Corpus (BNC)) decides to include special tags like ditto tags. Ditto tags are used on words that change their normal roles when in certain combinations. The first word of the combination is tagged as normal and all subsequent ones are assign the first's word tag followed by ditto symbol (Atwell, 2008). Ditto tags and linked pair tags (in Igb tagset) have similar tagging style for multiword lexical items. Igbo tagset used linked pair tags in noun class, verb class and conjunction to show relationships that exists between multiword units.

| Class | Sentence | Solution |
|---|---|---|
| Participle VPP | ọ na- esi nri "S/he is cooking" | VPP once is preceded by na- (auxiliary verb) |
| | esi m Sheffield abịa "I am coming from Sheffield" | Added VSI_BPRN to indicate that "e" in "esi" is m-bound. M = BPRN instead PRN. |
| Multiword nouns: agentive (NNAV NNAC) and instrumental (NNTV NNTC) nouns | If ọgụ/NNAV egwu/NNAC "singer", ngwu/NNTV ji/NNTC "digger", nnwere onwe is ?? "freedom" | We collapsed all to NNCV and NNCC since they are all common nouns. |
| Verb inherent complement VIC | -gba/VIC egbe/NNH "shoot", -gba/VIC egwu/NNH "dance", -gba/VIC ọsọ/NNH "run" | Removed VIC and Identified verbs based on the role they play in a context. Eg. ịgba/VIF egbe/NNH "to shoot", agba/VPP egbe/NNH "shooting", ... |

Figure 6.4: Sample problems and solutions during Inter-Annotation Agreement exercise

The tagset revision process affected IgbTS0 size because tags were simplified, removed, and added: the size moved from 59 tags to 70 tags. The effects of some IgbTS0 revisions are seen in the table 6.7. Some examples of tags simplification, removal and addition

exercise during IAA phases are shown in figure 6.4. Comparing our IAA results with Brants (2000a) IAA for POS tagging German Newspaper Corpus, Inter-annotator agreement was calculated between two coders and an accuracy of 98.80% was achieved, which is close to 98.71% highest score achieved by *l1+l2* in figure 6.3

Extensional suffix and pronoun bound markers (XS and BPRN) were added on selected base tags[9] of the tagset where they are likely to occur. After tagging entire texts, we found four tags of the tagset not used. They are tags with XS and BPRN markers viz; VGD_XS, ADJ_XS, VCO_BPRN and VCO_BPRN_XS. The base form of these tags are used in the tagged corpus, but words that represents these tags with markers happened to not appear the available corpus.

## 6.4 Conclusion

We started this chapter by creating the first ever part-of-speech (POS) annotation scheme for Igbo. We started this task by designing basic POS classes for Igbo looking at various linguistic materials available for the language. We found Emenanjo (1978) textbook on Igbo grammar valuable for this work. From it we took 7 core POS tags and expanded it to 25 POS tags. Then using about 23% tokens of the main corpus, and consulting EAGLES and Igbo linguistics materials where necessary, this 25-tag 'tagset' was further expanded to 59[10], which serves as the initial tagset of Igbo. We manually implement the tags of this tagset on the Igbo corpus in order to identify challenging phenomena in the language. The solutions on how to handle the identified challenging phenomena were included in the tagset as the associated tagging guideline. Also, implementing the tags of the 59-tag tagset on the Igbo corpus produced a 'POS tagged corpus' for Igbo. Evaluation of this 'POS tagged corpus' to ascertain how taggers will perform on it led to further investigation on the 59-tag tagset (and the associated guideline) that produce it. The tagset was revised by simplifying some its tags through Inter-Annotation Agreement (IAA) exercise. The results of IAA show high level of consistency among the IAA's annotators, which confirm the validity and reproducibility of the revised tagset.

The tagset we developed can be used in all kinds of Igbo texts, since tokens (whether in tone or non tone marked texts; diacritically or non diacritically marked texts; or any dialect texts) play the same grammatical roles. This is different from Sherpa tagset that was designed for only available texts (Gelu, 2010).

The exercises we performed in this chapter resulted in producing 'POS tagged corpus' (we shall call it IgbTNT1) developed from the 59-tag tagset, and tagged sub-corpora developed through IAA exercise. They will be used in the next chapter.

[9]Tags without any attached marker.
[10]See footnote 1.

# Part III

# Data Improvement, POS System and Morphological Features

# Chapter 7

# Data Improvement

There are different application modes to consider when planning to label tokens in a corpus: develop a part-of-speech (POS) tagging scheme, manually annotate all or a significant amount of the corpus, or opt for a mixed method, such as manually annotating a part, and the remainder semi-automatically. The output of semi-automatic annotation will be hand checked. Automatic annotation is less error-free but can produce many more POS tagged corpora than humans can reasonably achieve. Manual is more error-free, but very labour-intensive and costly, and outcome of the process is often used to train a machine to perform automatic annotation. A good annotation-based system combines both processes to form "semi-automatic" annotation. In semi-automatic annotation, manual steps can come in several stages of the overall process.

In this chapter, we will discuss how we exploited the by-products of Inter-Annotation Agreement (IAA) exercise, developed in the previous chapter, in a semi-automatic way to improve the quality of *IgbTNT1*[1]. Most of the abbreviated names are brought forward from the previous chapters.

## 7.1 Related Work

There have been works done in monolingual bootstrapping of manual POS annotation, and automatically correcting errors found in a tagged corpus. By monolingual bootstrapping, it means that the resources applied only focuses on the target language. Instead of going through tagged texts[2] word by word or sentence by sentence by human expert to find and correct errors, an efficient means can be developed that uses the human expert in its process loop to correct errors found or make suggestions to improve method's efficiency. Brill and Marcus (1992) use a three-step semi-automatic technique for tagging an unfamiliar text, which will enable somebody to annotate a large text he does not know with little help from a native speaker. First, they uncovered a set of tags through observing distributional behaviour of words in the text under study, then built a lexicon that identified most likely tags for each word and finally, learned rules to both correct

[1] This is the cleaned up version of the initial tagged corpus *IgbTNT0* developed in section 6.2.1 of chapter 6 before tagset revision. This tagged corpus is the New Testament Bible texts referred to as *IgbNT* in the previous chapters (5 and 6).

[2] May be tagged in a fashion to avoid manual tagging from starting to the finishing points or want to improve existing tagged corpus.

errors and find where contextual information can repair tagging mistakes. Taljard et al. (2008) and Heid et al. (2006) use lexicon-based pre-tagging system that contains 7000 known words and their annotations to tag 40000 tokens of Northern Sotho's texts. The lexicon consists of manually tagged list of all closed class words, list of 3700 high frequency verb stems extracted from 6.2 million University of Pretoria Sepedi Corpus (PSC), a list of manually tagged 1000 most frequent words from PSC and 335 personal and place names. After using this lexicon in pre-tagging, the output is a partially and ambiguously tagged corpus and some tokens left untagged. The latter output is assumed to be nouns or verbs since they are open class words not covered by the lexicon. They designed a noun and verb guesser to tag these tokens left untagged. The output of these processes are reviewed manually and correct guesses are added to the lexicon. Thus the size of the lexicon grows continuously.

Finding and correcting errors to make more accurate annotated data as experimented in Loftsson (2009) and Helgadóttir et al. (2012) is method of correcting errors found automatically in a tagged corpus. Loftsson (2009) and Helgadóttir et al. (2012) apply trained POS taggers singly and collectively, then the outputs were compared with the gold standard, and differences found were marked as error candidates for verification. Leech et al. (1983) use three stages to perform overall process of automatic tagging of LOB by using Tagged Brown Corpus. First, a human inspector manually prepares the raw corpus for automatic tagging input with the help of computer-aided pre-editing, then the output of the automatic tagging (tagged corpus) is subjected to manual computer-aided post-editing where human inspector corrects any errors made during automatic tagging.

In section 6.3.1 of chapter 6, six sub-corpora are produced through IAA exercise. In our experiment, we apply a semi-automatic method that learns and propagates changes found in these six sub-corpora into *IgbTNT1*. The essence of this process is to improve the quality of *IgbTNT1*[3] in order to keep it up to date with the tokens and tags affected by the IAA decisions. All positions where these changes occurred are marked and inspected further for quality assurance.

## 7.2 Improvement Methods

This section discusses the use of a machine learning method and human annotator expert in the loop of our system to improve the POS tagging efficiency of an already tagged corpus. The system is a three-variety error detecting and correcting approach. Later in this section is the use of Loftsson (2009)'s committee of taggers approach.

### 7.2.1 Method1: Transformation-Based Learning (TBL) As a Propagation Agent

We have created a satisfactory tagset (and associated guideline) through the revision of initial tagset (*IgbTS0*[4]) developed for Igbo. To create a gold standard of the Igbo corpus, which is what to use in training and testing machine learning classifiers, it is expected that those human annotators involved in the tagset revision cycle in section 6.3.1 of chapter 6

[3]See footnote 1.

[4] This is the initial tagset before revision. It has been discussed in chapter 6.

are used. It is believe that these human annotators have internalized best understanding of the revised tagset (IgbTS1) to annotate a fresh copy of untagged Igbo corpus, or to identify and correct errors found in *IgbTNT1* as a result of the tagset revision. Using human annotators plus a well designed tagset to start a fresh tagging process is preferable but not necessarily important, and of course this approach will consume time and money. The IAA exercise concluded in chapter 6 has outputs, which are tagged sub-corpora (selected texts from *IgbNT*[5]). These sub-corpora contain information (tags and tokens), which some of them are changes due to IAA decisions. This information is part of what the human annotators have internalized, and the same they will apply onto the untagged Igbo corpus if to start afresh.

*IgbTNT1* was tagged with *IgbTS0*[6] before the revision exercise, and thus contains most of information in the revised tagset (IgbTS1). Therefore, instead of using the tedious and uneconomical 100% manual method to tag a fresh copy of untagged Igbo corpus, or to identify and correct errors in *IgbTNT1*, we devise an automatic method that will inductively learn from the six IAA's sub-corpora, and then improve *IgbTNT1*. This is done by propagating changes learned from the sub-corpora to the *IgbTNT1*, and flagging locations where these changes occurred for inspection by a human annotator expert. Through this largely automated process, we expect to reduce the amount of human annotator time and effort, by only requiring the attention of a human annotator (the expert) on the marked positions instead of the entire text. Thus the quality of the corpus is increased with a minimum of expense. The approach of requiring that all revisions should be inspected by an expert annotator is needed to ensure a good quality end-product, with an accuracy that could not be achieved through a purely automated process.

**Experimental Data and Setup:** Resources used in this experiment are the by-products of IAA exercise, a machine learning technique called TBL, and a human annotator expert. By-products refer to the sub-corpora[7] tagged in IAA exercise in section 6.3.1 of chapter 6. There are five human annotators used, and IAA took three iterative steps, hence, making a total of 15 sub-corpora outputs with five sub-corpora at each step. These sub-corpora were grouped based on the steps they were used. There are three experimental phases in this method, each phase used sub-corpora of the corresponding phase in IAA.

There are different decisions made at each IAA phase that affected the use of tags. Hence using the entire IAA's sub-corpora in a single phase experiment will cause some of the changes with less impact to be subsumed into the ones with larger impact, thereby preventing the less impact changes from reflecting in the final output of the experiment. At this level, we are interested in getting all necessary information that will help improve the quality of Igbo corpus.

Transformation-Based Learning (TBL) by Brill (1995a) is preferable for this experiment since it is a machine learning algorithm that uses two states, initial (erroneously tagged text) and truth (correctly tagged text) states. It starts with these states and iteratively learns

[5]New testament Bible texts, an untagged version of *IgbTNT1*

[6]See footnote 4.

[7]They are selected texts from IgbNT. In each IAA phase, a selected text was given to five human annotators resulting to five tagged sub-corpora with each represents an annotator's judgements. See section 6.3.1 of chapter 6.

rules that correct errors in the initial state, until it resembles the truth to an acceptable degree. The TBL deployed in this experiment is Transformation-Based Learning on the Fast-Lane (fnTBL) by Ngai and Florian (2001), with the provided 40 rule templates at a threshold of 2.

There are different experiments in this section (methods 1.1 and 1.2), in each case TBL proposes additional changes, from which new rules can be formed in the next experiment. Human annotators used in the tagset revision were not used beyond this point, except for the human annotator expert who inspects the TBL changes on the original tagged corpus. The corpus is automatically updated according to the accepted changes after the human expert's adjudication (see table 7.3). The TBL model is retrained on the newly corrected corpus, and is thus updated after each iteration. The output template for inspection is of the form P A B C, where P is the marked position ($i$), A is TBL changed tag ($w_i/t^1$), B is the current tag ($w_i/t$), and C is $i$'s contextual information ($w_{i-2}/t$, $w_{i-1}/t$, $w_i/t$, $w_{i+1}/t$, $w_{i+2}/t$). The general algorithm of this propagation method experiment is

1. Get a sub-corpus from the IAA's sub-corpora to serve as TBL truth state, TS.
2. Take "the corresponding portion" of *IgbTNT1* to serve as TBL's initial state, IS.
3. Train TBL model on both TS and IS.
4. Apply TBL generated rules to entire *IgbTNT1*.
5. Inspect locations where rules 'fire'.
6. Repeat from step 1 for TS from each phase of IAA.

**Method1.1: Collated Silver Standard As TBL Truth State** We used 'silver standards'[8] developed from each group of five sub-corpora to serve as TBL truth state and take "the corresponding subset" from *IgbTNT1* as TBL initial state. We trained a TBL learner on both states and applied the TBL model to the entire *IgbTNT1* to find errors and flag affected positions for inspections. The idea here is that the materials from *IgbTNT1* is in erroneous state. TBL will learn rules from the IAA's sub-corpora to correct these errors. When the same rules are applied elsewhere in the corpus, the location where any rule 'fire' can be seen as candidate instances of the similar errors. All these locations are inspected by a human annotator expert. Since the TBL rule that fires at a location will propose a specific tag change, the human annotator expert can either *accept the TBL proposed change*, *retain the existing tag at the location where the current tag is deemed correct*, or *impose an alternative change* according to his knowledge of revised tagset when neither TBL proposed tag or current tag are correct. For efficient inspections, the human annotator expert used contextual information of the marked positions, which helps in facilitating corrections. Note that all locations inspected by human annotator expert are marked *never to be inspected again* because we believe that human annotator expert judgement supersedes any other one. This method took three iterative steps because the entire IAA process took 3 iterative steps, each step has a 'silver standard'. We shall refer

[8] Develop by collating tags with highest number of voting- where all annotators agreed on one tag. Method has been discussed in section 6.3.1 in chapter 6.

the output of this *Method1.1* as "*IgbTNT1*$^I$" to differentiate it from "*IgbTNT1*". The results of this experiment is in table 7.2.

**Method1.2: Each sub-Corpus As TBL Truth State** Among the human annotators used for IAA exercise in chapter 6, there are some that have better understanding of a particular tag than the others. Therefore, some tags that were voted out in silver standard[9] collation might be correct if found and inspected. In this phase, we went further to evaluate this intuition by finding in each of the sub-corpora tags that were not captured in the silver standard. Individual sub-corpus of the IAA's sub-corpora was used as TBL truth state with "the corresponding subset" of *IgbTNT1*$^I$ as TBL initial state. This method took 15 iterative steps because there are 15 sub-corpora produced in the entire IAA phases, and each was used as TBL truth state. See table 7.2 for results.

The aim here is to find and inspect on *IgbTNT1*$^I$ where one annotator's judgement is different from others and vice versa. Each IAA's sub-corpora represents an annotator's judgements. TBL was trained separately on each *IAA phase* sub-corpora[10], giving a total of five trained TBL models per IAA phase. We then inspected word-tag pairs from *IgbTNT1*$^I$ that matched certain patterns, such as one annotator's TBL model disagreeing with the four others. In total, there are three patterns to identify:

- **Method1.2A** This is pattern 1 where one TBL model's rule fired and four others did not. That is, TBL model trained pn one annotator's sub-corpus disagreeing with the four others.
- **Method1.2B** This is pattern 2 where four TBL models' rule fired and one did not.
- **Method1.2C** This is pattern 3 where both (one vs four combined) fired , suggesting different tags from each other and from main corpus.

We shall call the output of this **Method1.2** "*IgbTNT2*" to differentiate from "*IgbTNT1*$^I$".

**Method1.3: Tag Error Check** This error check on *IgbTNT2* was inspired by methods 1.1 and 1.2. Firstly, all tokens in *IgbTNT2* with tags that are not in the revised tagset were checked and changed. This is done through building a tagset dictionary and passing *IgbTNT2* through it. Secondly, the TBL propagation process discussed in the above methods correctly reclassified some tokens, introducing new tags from the revised tagset. However, because of the small amount of corpus size used for TBL training, the model lacked the capacity to apply learned rules widely on the entire corpus missing some tokens that should get new/changed tags. For example, *ntachi obi* is an example of a multiword expression in Igbo meaning "steadfastness". They occur as a "link-pair" adjacent to each other without any intervening word. The second pair is complementing the meaning of the first. After TBL propagation method, as shown in *IgbTNT2* column of table 7.1, "ntachi" got a new tag (NNCV) in 35 locations and its pair "obi" also got NNCC in 35 locations. The latter occurred 798 times in the entire corpus. It can occur by itself as a noun or adjacent to a verb as a noun completing verb's meaning. *freq* column of the same

[9]See footnote 8.

[10]There are 3 iterative phases of IAA exercise, in each phase, a sub-corpus was given to five annotators resulting into five sub-corpora. See section 6.3.1 of chapter 6.

table shows that there are other 3 *ntachi obi* yet to get the new link-pair tags (NNCV and NNCC). We tracked all other locations in *IgbTNT2* where this link-pair occurred and inspect them to see whether they are suppose to get this tag or not. Outcome of our inspection is shown on the *IgbTNT3* column of table 7.1. We shall refer this **Method1.3** output as "*IgbTNT3*" to differentiate it from "*IgbTNT2*". This process corrected 4994 errors in *IgbTNT2*.

| Token | Freq | *IgbTNT1* | *IgbTNT2* | *IgbTNT3* |
|---|---|---|---|---|
| ntachi | 38 | NNC=35 | NNCV=35 | NNCV=38 |
| | | VCO=1 | NNC=1 | |
| | | NNAV=2 | NNAV=2 | |
| obi | 38 | NNC=37 | NNCC=35 | NNCC=38 |
| | | | NNC=2 | |
| | | PRN = 1 | PRN=1 | |
| ntụkwasị | 67 | VSI_XS=5 | NNCV=26 | NNCV=67 |
| | | NNAV=1 | NNC=40 | |
| | | VCO=6 | NNAV=1 | |
| | | NNC=55 | | |
| obi | 67 | NNC=67 | NNCC=27 | NNCC=67 |
| | | | NNC=40 | |

Table 7.1: Some examples of tag error check and corrections

### 7.2.2 Method2: Use of Committee of POS Taggers

We adopted Loftsson (2009) and Helgadóttir et al. (2012) method for finding and correcting errors on gold standard corpus using combination of different taggers. Taggers were trained on 90% portion of *IgbTNT3* and tested on the 10% portion of same corpus. Then locations where all taggers agreed on a tag but disagreed on *IgbTNT3*'s tag are marked as potential candidates for inspection. We used Stanford Log-linear Part-of-Speech (POS) Tagger (Toutanova and Manning, 2000; Toutanova et al., 2003), MBT– A memory-based POS tagger-generator by Daelemans et al. (1996), and FnTBL– Transformation-based learning in the fast lane by Ngai and Florian (2001) for this experiment. See table 7.2 for results. We shall refer the output of this method as "*IgbTNT4*" to differentiate it from "*IgbTNT3*".

### 7.2.3 Corpus Improvement Results

We trained TBL, a machine learning technique, on the IAA's tagged sub-corpora of the Igbo corpus with the assumption that errors flagged by the generated rule-based model will be the type of errors that occur in the main tagged Igbo corpus. The flagged positions are considered error candidates for inspections. This process is to improve the quality of this main corpus.

Table 7.2 gives result analysis of the above improvement processes. *Location Flagged* is number of inspected positions that TBL model flagged. *Accepted Judgement* is the number of TBL changed tags accepted where the current tag is not correct. *No-Change Required* is the rejected TBL judgement where current tag prevailed. *Manual Change*

| Name | Location Flagged | Accepted Judgement | No-Change Required | Manual Change | Effective Change | % Error Eliminated from the Corpus |
|---|---|---|---|---|---|---|
| Method1.1 | 25490 | 16612 | 5569 | 3309 | 19921 | 7.550 |
| Method1.2A | 26155 | 3605 | 20471 | 2079 | 5684 | 2.154 |
| Method1.2B | 631 | 33 | 555 | 43 | 76 | 0.029 |
| Method1.2C | 53 | 1 | 47 | 5 | 6 | 0.002 |
| Method1.3 | 4994 | 4994 | - | - | 4994 | 1.893 |
| Method2 | 11810 | 6549 | 4165 | 1096 | 7645 | 2.897 |
| Total | 69133 | 31794 | 30807 | 6532 | 38326 | 14.525 |

Table 7.2: Total statistics of outcomes of various data improvement methods

| Instance | TBL POS-tag | Accepted | Manual Change | Final POS-Tag | Meaning |
|---|---|---|---|---|---|
| ahụ/DEM | VPP | YES | | VPP | see |
| ahụ/DEM | VPP | NO | | DEM | that |
| n'/VAX | PREP | YES | | PREP | in/on/from |
| na/VAX | CJN | YES | | CJN | and |
| na-/NNC | VAX | YES | | YES | auxiliary verb |
| onye/NNM | NNC | YES | | YES | person |
| ndị/NNC | NNM | YES | | YES | people of |
| onwe/PRNREF | PRNEMP | YES | | PRNEMP | self |
| ya/PRN | PRNREF | NO | | PRN | her/him |
| unu/NNM | PRN | YES | | PRN | plural you |
| dịkwa/VCO | VSI_XS | YES | | VSI_XS | is also |
| kọrọ/VrV | VPP_XS | NO | | VrV | told |
| nyere/VCO | VSI_XS | NO | VrV | VrV | gave |
| ná/CJN | PREP | YES | | PREP | in/on/from |
| a/DEM | PRN | NO | | DEM | this |
| a/DEM | PRN | YES | | PRN | impersonal pronoun |
| ana/VPP | VAX_BPRN | YES | | VAX_BPRN | pronoun prefix "a" attached to "na-" |
| m/PRN | BPRN | YES | | BPRN | "I" bound to "a/e" pronoun |
| óké/NNC | NNH | YES | | NNH | boundary |
| nwere/VrV | VMOV | YES | | VMOV | [nwere ike] can |
| ike/NNC | VMOC | YES | | VMOC | [nwere ike] can |
| ekwesị/VPP_XS | VPP | NO | BCN | BCN | right/correct |
| ònye/WH | NNC | NO | | WH | who |
| ntachi/NNC | NNCV | YES | | NNCV | [ntachi obi] steadfastness |
| obi/NNC | NNCC | YES | | NNCC | [ntachi obi] steadfastness |
| esi/VPP | VSI_BPRN_XS | NO | VSI_BPRN | VSI_BPRN | simple verb "si" with pronoun prefix "e' |

Table 7.3: Some samples locations flagged by TBL inspected by human annotator expert

is where neither TBL proposed tag nor current tag was correct, the human annotator expert chose from the revised tagset. *Effective Change* is the number of effective change 'impact' each method made on the corpus, and *% Error Eliminated from the Corpus* is the rate of errors eliminated in each method. In method1.1 where we used TBL and silver standard materials, there are 25490 locations inspected on *IgbTNT1* with 19921 effective changes. That means this method flagged 25490 locations in *IgbTNT1* (≈10% of the entire corpus) for human annotator expert to inspect. This is easier and cheaper step compared to asking the human annotator expert to examine methodically the locations in the entire corpus where tagset revision changes are to be reflected. About 78% of these flagged locations where effectively changed, in effect, 7.550% errors were eliminated from *IgbTNT1* tagged corpus. Hence, this process is a very efficient way of bootstrapping POS tagging of a corpus or correcting errors in a tagged corpus. While in method1.2, 26839 word-tag pairs were inspected. There are three patterns of experiments in this method, and the following effective changes on $IgbTNT1^{I}$ to get an improved version of *IgbTNT2* are observe: 5,684 for Method1.2A (pattern 1), 76 for Method1.2B (pattern 2), and 6 for Method1.2C (pattern 3). Number of corrections for methods 1.2B and 1.2C patterns are quite low, and could arguably be dispensed with. For method1.2A pattern where one annotator's judgement disagreed with four other annotators' judgements, rate of corrections ( 14%) is substantially lower than method1.1, but number of corrections (2.154% errors were eliminated from the corpus) justifies this as an effective process.

Method2 row shows that 11810 locations are where three of taggers are disagreeing with the tags in the main corpus (*IgbTNT3*). After inspection, an effective change of 7645 was made to improve the corpus. We evaluated this and got an increased accuracy scores from 94.007% to 96.665%. See table 7.4 for accuracy scores on each improvement methods. The entire improvement process resulted in inspecting 26.20% of the main corpus with 14.525% effective change made.

A few samples of this experiment are displayed in table 7.3. The columns show the affected samples, TBL suggested tags, accepted (whether the TBL suggested tag was accepted by the human expert), manual correction (if TBL suggested tag and current *IgbTNT1* tag were wrong), and final state of tags. Interestingly, some tokens were correctly reclassified, even new tags introduced in the revised tagset as a result of the IAA decisions are correctly inserted into the main corpus. The Igbo corpus size of 263,854 tokens, which initially had 54 tags annotated according to the IgbTS0 tagset, now contains 63 tags, including all changes in the revised tagset.

## 7.3 Corpus Improvement and Tagging Accuracy

Semi-automatic methods have been applied to improve the IgbTNT corpus. We expected that each method would bring improvements in the corpus patterns consistency. In this section, we performed automatic tagging on all the outcomes of the above methods: *IgbTNT0*, *IgbTNT1*, *IgbTNT2*, *IgbTNT3* and *IgbTNT4* to show improvement rates. For the evaluation performance, we split the corpora into 10 folds. 10-fold subsets were created by slicing the the corpora into 822 sentences, each is 25,981 words on the average. Slicing on the sentences is making sure that each piece contained full sentences (rather than cutting off the text in the middle of a sentence). For 10-fold steps, we trained TBL

classifier on 9-fold and tested on the held-out. The experiment was performed on closed vocabulary assuming that there is no previously unseen words in the training session. The results are summarised in table 7.4.

| Fold | Accuracy | | | | |
|---|---|---|---|---|---|
| | *IgbTNT0* | *IgbTNT1* | *IgbTNT2* | *IgbTNT3* | *IgbTNT4* |
| 0 | 84.509 | 88.748 | 94.027 | 94.462 | 97.101 |
| 1 | 90.522 | 91.413 | 93.171 | 93.653 | 96.841 |
| 2 | 90.743 | 90.809 | 92.871 | 93.682 | 97.009 |
| 3 | 92.153 | 92.474 | 94.214 | 94.489 | 97.084 |
| 4 | 92.098 | 93.119 | 94.687 | 94.816 | 96.787 |
| 5 | 81.980 | 85.974 | 93.151 | 93.492 | 96.185 |
| 6 | 89.342 | 90.589 | 93.215 | 93.809 | 96.466 |
| 7 | 85.684 | 88.433 | 93.287 | 93.691 | 96.440 |
| 8 | 88.186 | 89.913 | 93.621 | 94.063 | 96.234 |
| 9 | 86.996 | 90.190 | 93.409 | 93.920 | 96.452 |
| **Average** | **88.221** | **90.166** | **93.565** | **94.007** | **96.665** |

Table 7.4: Simple accuracy on 10-fold evaluation for various outcomes of data improvement methods

From the table 7.4, we can deduce that there is constant improvement on the tagged corpus after each process. A total improvement score of 8.44% was achieved; cleaning up exercise *IgbTNT1* gave 1.95% improvement, TBL propagation gave an additional 3.40%, tag error check another 0.44%, and finally taggers committee approach gave extra 2.66%. The entire processes in table 7.2 flagged 69133 (26.20% of the corpus) word-tag pair positions which were inspected by a human annotator expert contributed 14.525% (by eliminating 38326 errors) improvement on the tagged Igbo corpus.

## 7.4 Re-usability

Igbo language has 30 dialects as a result of nasality and aspiration. Our tagset and corpus annotation is based on primarily standard Igbo, which omits the nasality and aspiration found in those dialects. The tagset (and associated guideline) are applicable to all 30 dialects, since these dialectal words play the same grammatical role as found in the standard Igbo texts, through which the tagset is developed. For example, the interrogative sentence *olee aha gị*? "what is your name?" in standard Igbo is said in different dialects as *ndee afụa gụ*?, *ndee awa ghụ*?, etc. "ndee" is equivalent to "olee" which makes the sentence interrogative, *afụa, ewa* is equal to "aha" and *gụ, ghụ* is equal to "gị". Therefore, if we create a dictionary of word-types from the Bible in all dialects, with standard Igbo as a reference point, the annotated Bible corpus in standard Igbo can be used to annotate other dialects with minimal errors.

## 7.5 Conclusion

We have discussed a methodology that helps to improve tagged corpus through exploiting by-products of Inter-Annotation Agreement (IAA) exercise due to tagset revisions. It is

a semi-automatic method that uses a machine learning (ML) algorithm where a human annotator expert is called severally within the loop to validate a particular judgement. This method inductively learn errors by comparing the truth with its corresponding erroneous state, and then apply the outcome on the entire erroneous state for improvement. The truth is a small subset of the 'erroneous' state that has been refined via IAA process. We observe that even the new tags introduced in the revised tagset due to IAA decisions were well propagated into the main corpus, and wrongly tagged tokens that were corrected in the IAA exercise were also identified and corrected in the main tagged corpus. Through this largely automated process, the quality of the original tagged corpus was improved. We also applied tagger combination method to suggest possible erroneous candidates in the tagged corpus for inspection by a human annotator expert.

The evaluation result shows that we achieved an improvement of 8.44% on automatic tagging accuracy over the entire process. The effort, time and money that would had been used to manually implement POS tagging of Igbo corpus were saved. In total, the entire processes gave 69133 (26.20% of the main corpus) positions inspected with 14.525% effective change made on the main corpus.

On re-usability, the TBL propagation method can be adopted to many annotation problems, especially low-resource languages with little linguistic materials. In Africa, of around 2000 languages in the continent, only a small number have featured in the NLP research field. Secondly, the text of this annotated corpus is in the standard Igbo. It is potentially re-usable on other dialects or genres aiming towards developing annotated corpora with correctable errors. There could be challenges such as the problem of unknown words when moving from religious genre to other genres or from standard dialect to other dialects.

It it important to note that we used the revised tagset and IgbTNT to annotate the novel mentioned in chapter 5. This annotation process was partly automated. First, we used a tagger trained on the IgbTNT on the novel text, and then hand-corrected errors found in the process. A total of 39960 tokens were tagged. This will enable us test how well generated taggers will perform across other genres.

For sake of clarity, we shall henceforth refer to this tagged corpora as IgbTNT for the religious texts represented by the Bible, IgbTMT for the modern texts represented by the novel, IgbC for the Igbo untagged corpus, IgbTC for the tagged Igbo corpus (both IgbTNT and IgbTMT), IgbNT for the Igbo untagged New Testament Bible texts, IgbMT for the Igbo untagged novel texts, and IgbTS for the revised tagset.

# Chapter 8

# Automatic Part of Speech Tagging

Part of speech (POS) tagging is a prerequisite step for many advanced Natural Language Processing (NLP) tasks such as information extraction, sentiment analysis, syntactic parsing, machine translation, etc. This chapter verifies whether our tagset and POS tagged corpus designed and developed for Igbo language in the previous chapters will deliver good tagging accuracy using existing POS taggers. We further check if anything can be learned about the language linguistic patterns that require POS taggers attention on accuracy.

In our verifications, we perform evaluative experiments featuring two different POS tagged genres and five existing POS taggers. The genres are the New Testament Bible corpus (IgbTNT) that represent tagged religious Igbo texts and the novel[1] (IgbTMT) that represent tagged modern Igbo texts. IgbTC stands for Igbo Tagged Corpus which is a combination of different Igbo text styles (e.g. IgbTNT and IgbTMT). Out of the five taggers used; three are statistical taggers, one is rule-based tagger, and the remainder is memory-based tagger.

To the best of my knowledge, research in Igbo Natural Language Processing (Igbo*NLP*) started with this work in 2013 when I began my PHD study. There was no literature found in Igbo NLP except linguistic literatures and some electronic texts of the language.

## 8.1 Tool Selection and Implementation

Igbo is a language in which a single stem can combine with affixes in in multiple different orders to produce many word forms. Hence, the Igbo tagset used in IgbTC was developed with the aim to capture all morphologically-inflected (morph-inflected) and non-inflected words in the language. There are feature markers attached to some of the tags used in IgbTC that can show the important characteristics of words in a sentence. Taking cognizance of these facts, we chose our tagging tools based on taggers that are commonly used, have done well on tagging generally, and have parameters for word feature extractions. Some exiting taggers use starting and ending $n$ length of letter sequences of each word as predictive features of unknown words (Brants, 2000b; Samuelsson, 1993). For example, $n = 4$ for *negotiable* will extract $-able$ which Brants' TnT tagger will use to predict that *negotiable* is likely to be adjective in English. Toutanova et al. (2003) uses *variable*

[1]Novel written in Igbo in 2013.

*length* suffixes up to a maximum length $n$ for extracting word features such that $n = 4$ for *negotiable* will generate [e,le,ble,able] feature list. These methods have worked well in languages like English and German whose derivational and inflectional affixes reveal much about the grammatical classes of words in question. However, it is uncertain how well they will perform in Igbo if not through testing them on the language corpora.

Table 8.1 shows five selected POS taggers and tools that implement them. They are all supervised taggers and represent five types of taggers discussed in section 4.3.2 of chapter 4. Since they are supervised taggers, a pre-tagged training corpus is reguired. This we achieved with corpora in table 8.6. Word feature extraction length for the taggers was set to $n$=5 because the longest suffixes in Igbo so far are 5 in length. Furthermore, the taggers achieve best performance at this length. For example, TnT and HunPOS accuracy scores at default length of 10 are 58.67% and 59.70% for unknown words in IgbTMT corpus, while at the length of 5 they scored 63.73% and 61.86% respectively.

| Tagger | Type | Tool |
|---|---|---|
| Baseline | Unigram | Self Coded in Python |
| SLLT[a] | Maximum Entropy | Stanford Tagger |
| TnT[b] | Hidden Markov Model | Brants Tagger |
| HunPOS[c] | Hidden Markov Model | Hungarian Tagger |
| FnTBL[d] | Transformation-Based Learning | FnTBL tagger |
| MBT[e] | Similarity-Based Reasoning | TiMBL[f] tagger |

[a]Stanford Log-linear Tagger by (Toutanova et al., 2003)

[b]Trigrams'n'Tags (Brants, 2000b)

[c]Hungarian Part-of-Speech Tagger is a reimplementation of TnT by (Halácsy et al., 2007)

[d]Transformation-based learning in the fast lane. Brill's TBL Brill (1995a) reimplemented by (Ngai and Florian, 2001)

[e]A memory-based part of speech tagger-generator by (Daelemans et al., 1996)

[f]Tilburg Memory-Based Learner

Table 8.1: POS taggers selected for experiments

SLLT is a Java implemented tagger. It is developed based on the ideas of preceding and following tag contexts through dependency network, broad use of lexicalization, effective use of priors, and fine-grained modelling of unknown words (Toutanova et al., 2003). SLLT accuracy scores are 97.24% and 89.04% for the overall and unknown words on the Penn Treebank English corpus.

TnT is Trigrams'n'Tagger implemented in C by Brants (2000b). It uses second order Hidden Markov Model (HMM). Model probabilities of TnT are extracted from the training corpus through maximum likelihood estimation. Its *-alength* word endings method use a suffix trie with maximum of suffix length *alength* to handle unknown words (Brants, 1999). TnT tagger accuracy scores on German NEGRA corpus are 96.7% for overall and 89.0% unknown words.

HunPOS (implemented in OCaml) is a reimplementation of Brants (2000b) TnT tagger. It is open source and free alternative to TnT (Halácsy et al., 2007). Its *-s n* parameter sets the *n*length of the longest suffix to be considered by the algorithm when it estimates an unseen word's tag distribution. HunPOS trigram tagger accuracy scores on Penn Treebank English corpus are 96.49% and 86.90% for the overall and unknown words, and

on the Hungarian corpus, it scored 98.24% and 95.96% for the overall and unknown words.

FnTBL is developed in C and Perl as fast reimplementation of Brill (1995a) Transformation-Based Learning (TBL). It starts with an initial state and requires a correctly tagged text, called *truth*, for training. The training process iteratively acquires an ordered list of rules that correct the errors found in the initial state until this initial state resembles the truth to some acceptable degree. FnTBL tagger overall performance score on the Penn Treebank WSJ is 96.76%. In Loftsson (2007), FnTBL accuracy scores on Icelandic corpus are 55.51% for the unknown words and 89.33% for the overall words.

MBT is Memory-Based Tagger implemented in C++. In the training phase, it gathers word $w$ and its context along with $w$'s correct tag in its memory as feature representative. New words $w_i$ are tagged in the testing phase by retrieving the tag of $w$ with most similar features from the memory, and assigning the tag to $w_i$. Feature patterns are defined to add extra information to the tagger concerning the contextual information and the form of the words to be annotated. This is done by the parameters *-p–* feature style for known words, and *-P–* feature style for unknown words. MBT results on WSJ corpus are 96.4% and 90.6% for the overall and unknown words.

## 8.2 Measures for Evaluation

The goal of evaluation in POS tagging is to understand how well a tagger performs on a specific language texts, either for comparison with other taggers or for understanding whether a new POS tagging system is needed for the language. The standard and generally used evaluation methods are hereby outlined:

For accuracy scores and error rates, we used

$$Accuracy = \frac{\text{number of correct tags produced by tagger}}{\text{total number of tokens/tags in the truth}} \tag{8.1}$$

$$error\ rate = 1 - Accuracy * 100 \tag{8.2}$$

Ambiguity rate is the average number of tags per word:

$$\frac{Total\ \#\ of\ unique\ tags\ per\ word\ types}{Total\ \#\ of\ word\ types} \tag{8.3}$$

Earlier, we calculated precision, recall and fmeasure for tag class $t$ using

$$precision_t = \frac{TP_t}{TP_t + FP_t} \tag{8.4}$$

$$recall_t = \frac{TP_t}{TP_t + FN_t} \tag{8.5}$$

Then we calculated their microaverages, macroaverages and fmeasures using

$$microaverage\ precision = \frac{\sum_{t \in T} TP_t}{\sum_{t \in T} TP_t + \sum_{t \in T} FP_t} \tag{8.6}$$

$$microaverage\ recall = \frac{\sum_{t \in T} TP_t}{\sum_{t \in T} TP_t + \sum_{t \in T} FN_t} \tag{8.7}$$

$$macroaverage\ precision = \frac{1}{|T|} \sum_{t \in T} precision_t \tag{8.8}$$

$$macroaverage\ recall = \frac{1}{|T|} \sum_{t \in T} recall_t \tag{8.9}$$

$$fmeasure = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{8.10}$$

T is the set of tags, t is a tag, TP is true positive, FP is false positive and FN is false negative. The $fmeasure$ can be interpreted as a weighted harmonic mean of the precision and recall.

In a classical POS tagging task, where each instance to be classified must receive only a single tag, accuracy is equal to both microaverage precision and microaverage recall. Illustrating this using table 8.2, lets assume that the total number (N) of tokens[2] to tag A is 7 and B is 5. Then TP (tokens correctly labelled A) + FP (tokens falsely labelled A) = total tokens labelled positively (3+5=8) and TP + FN (tokens falsely labelled **not** A) = total tokens that **should** have been labelled positive (3+4=7). If TP is divided by 8, you get precision (0.3750) answering the question "are all tokens labelled A correct?". If TP is divided by 7, you get recall (0.4286) answering the question "all tokens that suppose to get A, did they get it?". Since each token gets a tag (A or B) in POS tagging, total FP = total FN (5). Calculating microaverages and macroaverages:
Microaverage precision is $\frac{7}{7+5} = 0.5833$ (equation 8.6).
Microaverage recall is $\frac{7}{7+5} = 0.5833$ (equation 8.7).
fmeasure for Microaverage recall and precision is $\frac{2 \times (0.5833 \times 0.5833)}{0.5833 + 0.5833} = 0.5833$ (equation 8.10).

| | Tag | TP | FP | FN | N | Precision | Recall | Fmeasure |
|---|---|---|---|---|---|---|---|---|
| | A | 3 | 5 | 4 | 7 | 0.3750 | 0.4286 | 0.4000 |
| | B | 4 | 0 | 1 | 5 | 0.1000 | 0.8000 | 0.8889 |
| Total | 2 | 7 | 5 | 5 | 12 | | | |
| Macroaverages | | | | | | 0.6875 | 0.6143 | 0.6445 |
| Microaverages | | | | | | 0.5833 | 0.5833 | 0.5833 |

Table 8.2: Demonstrating precision, recall, fmeasure and their macro- and micro-averages calculations

Macroaverage precision is $\frac{37.50+100.00}{2} = 0.6875$, 2 is total number of tags (equation 8.8).
Macroaverage recall is $\frac{42.86+80.00}{2} = 0.6143$, 2 is total number of tags (equation 8.9).
fmeasure for Macroaverage recall and precision is $\frac{2 \times (0.6875 \times 0.6143)}{0.6875 + 0.6143} = 0.6445$ (equation 8.10).
See tables in appendix A.3, they contain tags precision, recall and their micro- and macro-averages and fmeasure for taggers on IgbTC. Compare these tables with IgbTC's overall accuracy scores in table 8.11.

[2]Or words.

## 8.3 Experimental Data Description

This section describes the properties (such as tag/word ambiguity) of the corpus data we used in the taggers' development experiments. Ambiguity reveals the proportion of tokens that are not ambiguous which the taggers will classify "for free" without struggle, and the proportion of tokens with more than one tag (ambiguous tokens) which the taggers have to struggle to classify. Table 8.3 shows the general properties of the Igbo corpora used, while table 8.4 and figure 8.1 show the most ambiguous words and the tag frequency distribution in IgbTC corpus. Table 8.5 displays the number tags used in each tagged corpus, and tag's frequency and probability distribution.

| Properties | IgbTNT[a] | IgbTMT[b] | IgbTC[c] |
|---|---|---|---|
| Token size | 263856 | 39960 | 303816 |
| Sentence size | 8219 | 2032 | 10253 |
| Type size | 6424 | 3122 | 8020 |
| tags Used | 63 | 61 | 66 |
| Ambiguity rate (amb. class) | 2.31 | 2.45 | 2.37 |
| Ambiguity rate (overall) | 1.11 | 1.09 | 1.13 |
| Ambiguous tokens | 29.73% | 34.88% | 36.65% |
| Ambiguous types | 8.50% | 6.44% | 9.35% |
| Inflected tokens | 11.89% | 14.07% | 12.18% |
| Non Inflected tokens | 88.11% | 85.92% | 87.82% |
| Inflected types | 65.63% | 57.68% | 65.26% |
| Non Inflected types | 34.36% | 42.32% | 34.74% |

[a]New Testament part of Igbo Tagged Corpus.
[b]Novel (Modern) texts part of Igbo Tagged Corpus.
[c]Igbo Tagged Corpus. Comprises IgbTNT and IgbTMT.

Table 8.3: The corpus data general statistics

The followings are observe from the figure and tables:

- Tags increase as corpus size increases (e.g. from table 8.3, 66 tags used in IgbTNMT $>$ 63 and 61 used in IgbTNT and IgbTMT).

- Word-type size increases as corpus size increases (e.g. from table 8.3, 8020 types used in IgbTNMT $>$ 6424 and 3122 used in IgbTNT and IgbTMT).

- *Ambiguity rate* from table 8.3 shows that tag/tokens ratio over ambiguous class is higher in IgbTMT with 2.45 (vs 2.31 in IgbTNT and 2.37 in IgbTNMT), and higher in IgbTNMT with 1.13 (vs 1.11 in IgbTNT and 1.09 in IgbTMT) over the overall class. Ambiguity rate is calculated with equation 8.3.

- The percentage *ambiguous tokens* from table 8.3 shows that taggers will disambiguate 29.71% tokens in IgbTNT, 34.87% tokens in IgbTMT and 36.65% tokens in IgbTNMT. This implies that taggers won't struggle to classify the remaining tokens (e.g. 70.29% tokens in IgbTNT) since they only get one tag.

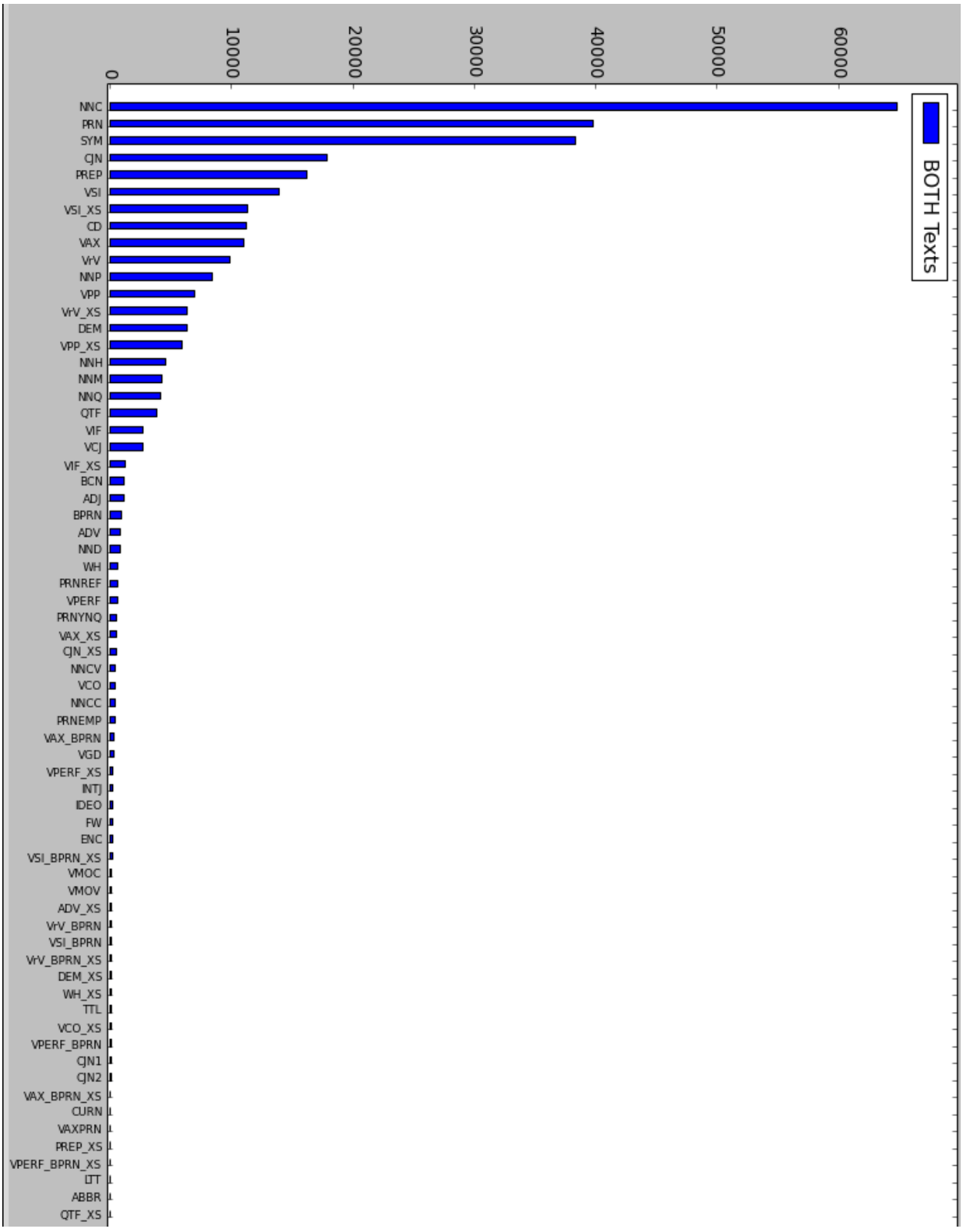



Figure 8.1: Tag frequency distribution of IgbTC corpora

| IgbTNT | | | |
|---|---|---|---|
| Token | tags | freq | tags and their frequency |
| aghara | 6 | 30 | NNCC=15 NNH=6 NNC=3 VrV=3 ADV=2 VPP_XS=1 |
| ike | 6 | 984 | NNC=443 NND=277 NNH=109 NNQ=83 VMOC=71 VIF=1 |
| aga | 5 | 152 | VPP=98 BCN=42 VAX_BPRN=7 NNQ=4 VSI=1 |
| agaghị | 5 | 146 | VAX_XS=85 VPP_XS=47 VSI_BPRN_XS=2 VSI_XS=7 VAX_BPRN_XS=5 |
| azụ | 5 | 94 | NNC=63 VPP=25 PREP=4 BCN=1 VPP_XS=1 |
| anọ | 5 | 160 | CD=99 VPP=46 VSI_BPRN_XS=10 VSI=4 BCN=1 |
| anya | 5 | 908 | NNH=589 NNC=270 VPP=28 NNCC=16 VSI=5 |
| arụ | 5 | 117 | VPP=79 BCN=27 NNC=7 VPP_XS=3 VSI=1 |
| adịghịkwa | 4 | 52 | VAX_XS=30 VPP_XS=14 VSI_XS=6 VAX_BPRN_XS=2 |
| otú | 4 | 791 | NNC=510 CJN=190 ADV=73 VAX=18 |
| IgbTMT | | | |
| Token | tags | freq | tags and their frequency |
| ahụ | 6 | 403 | DEM=299 NNC=67 VPP=17 NNH=11 NNCC=7 VSI=2 |
| aka | 5 | 139 | NNC=125 NNCC=6 NNH=5 VPP=2 BCN=1 |
| aga | 5 | 74 | VPP=37 BCN=19 VAX_BPRN=11 VSI=4 VAXPRN=3 |
| ike | 5 | 190 | NNC=108 VMOC=46 NND=19 NNH=14 NNQ=3 |
| ebe | 5 | 255 | NNC=232 CJN=14 VPP=7 BCN=1 VSI=1 |
| ma | 4 | 372 | CJN=326 CJN1=16 CJN2=16 VSI=14 |
| isi | 4 | 128 | NNC=120 VIF=4 NNH=3 NNCC=1 |
| onwe | 4 | 141 | PRNREF=111 NNCC=21 PRNEMP=8 NNC=1 |
| anọ | 4 | 56 | VPP=21 CD=31 VSI=3 VSI_BPRN=1 |
| abụ | 4 | 66 | NNCC=1 NNC=31 VPP=33 NNH=1 |
| IgbTC | | | |
| Token | tags | freq | tags and their frequency |
| ama | 7 | 142 | NNH=77 VPP=40 BCN=15 NNC=4 NNCC=3 VSI=2 VSI_BPRN=1 |
| ahụ | 6 | 4067 | DEM=3799 VPP=180 NNC=68 NNH=11 NNCC=7 VSI=2 |
| aga | 6 | 225 | VPP=135 BCN=61 VAX_BPRN=18 VSI=5 NNQ=4 VAXPRN=2 |
| anọ | 6 | 216 | CD=130 VPP=67 VSI_BPRN_XS=10 VSI=7 BCN=1 VSI_BPRN=1 |
| aghara | 6 | 33 | NNCC=15 NNH=6 NNC=6 VrV=3 ADV=2 VPP_XS=1 |
| arụ | 6 | 188 | VPP=140 BCN=27 NNC=15 VPP_XS=3 VSI=2 NNH=1 |
| ike | 6 | 1174 | NNC=551 NND=296 NNH=123 VMOC=117 NNQ=86 VIF=1 |
| ezi | 5 | 541 | NNQ=390 BCN=80 VPP=50 NNC=16 VSI=5 |
| asị | 5 | 413 | VPP=367 NNH=29 NNC=10 VSI=6 VSI_BPRN=1 |
| aka | 5 | 1025 | NNC=908 NNH=52 BCN=47 VPP=12 NNCC=6 |

Table 8.4: 10 most tag ambiguous words in Igbo corpora

- Table 8.4 shows that the frequent ambiguous tokens are mainly non-inflected or less inflected words. For example, *aghara* is the only inflected token in the table with a single suffix *-ra*. In Arabic, the highest rate of ambiguity appeared at the stem level, but decreases with inflection, and even decreases further when clitics are added (Attia, 2008). Also, Heid et al. (2006) reveal that the most frequent and ambiguous tokens in Northern Sotho are not morphologically-inflected. This indicates that the more ambiguous a word is, the more chances it has fewer or no suffixes, and the more frequent it is likely to be.

- The rate of *ambiguous types* in IgbTNMT is higher than in each of IgbTNT and IgbTMT. This is because a type which is unambiguous in one or both corpora may be ambiguous in the combined corpus (IgbTNMT). For example, a word type *ude* appeared only with **NNC** tag in IgbTMT meaning "pomade", and only with tag **NNH** complementing the verb *-sụ*[3] in IgbTNT, so that is unambiguous in both. In IgbTNMT, however, it would be classed as ambiguous, as would all of its occurrences, and hence also the higher rate of ambiguous tokens.

- The majority of most frequent tokens in Igbo are *not inflected* while the majority of most frequent types are *inflected*. Table 8.3 shows that *inflected tokens* in IgbTNT is 11.89%, and *inflected types* is 65.63%. But in *non-inflected*, IgbTNT contains 88.11% tokens, and 34.36% types. Figure 8.1 shows that tags with *XS*[4] extension are skewed on the right, and these are tags with low frequency for *inflected tokens*. These observations indicate that *inflected tokens* are one of the major constituents of rare word class in Igbo.

- Table 8.4 presents the frequency of $w/t_i$ where $w$ is an ambiguous word and $t_i$ represents different tags of $w$. This is to show how often an ambiguous word occur given a tag in its ambiguous set. For example, "ahụ" is 93% a demonstrative (DEM), 4.43% as participle (VPP) and only 0.05% a simple verb (VSI). This information is used in section 8.5.3 for automatic tagging results analysis on words having unique tags up to 5 and above.

| IgbTNT | | | IgbTMT | | | IgbTC | | |
|---|---|---|---|---|---|---|---|---|
| Tags | Freq | Prob | Tags | Freq | Prob | Tags | Freq | Prob |
| NNC | 55305 | 0.209604 | NNC | 9545 | 0.238864 | NNC | 64850 | 0.213452 |
| PRN | 34618 | 0.131201 | PRN | 5124 | 0.128228 | PRN | 39742 | 0.130810 |
| SYM | 33932 | 0.128601 | SYM | 4397 | 0.110035 | SYM | 38329 | 0.126159 |
| CJN | 16063 | 0.060878 | PREP | 2035 | 0.050926 | CJN | 17856 | 0.058773 |
| PREP | 14160 | 0.053666 | CJN | 1793 | 0.044870 | PREP | 16195 | 0.053305 |
| VSI | 12253 | 0.046438 | VSI | 1700 | 0.042543 | VSI | 13953 | 0.045926 |
| CD | 10657 | 0.040390 | VrV | 1553 | 0.038864 | VSI_$\alpha$ | 11288 | 0.037154 |
| VSI_$\alpha$ | 9809 | 0.037176 | VSI_$\alpha$ | 1479 | 0.037012 | CD | 11197 | 0.036855 |
| VAX | 9488 | 0.035959 | VAX | 1468 | 0.036737 | VAX | 10956 | 0.036061 |
| VrV | 8286 | 0.031404 | NNM | 1241 | 0.031056 | VrV | 9839 | 0.032385 |
| NNP | 7432 | 0.028167 | VPP | 1044 | 0.026126 | NNP | 8404 | 0.027662 |
| VPP | 5943 | 0.022524 | VrV_$\alpha$ | 984 | 0.024625 | VPP | 6987 | 0.022998 |
| DEM | 5418 | 0.020534 | NNP | 972 | 0.024324 | VrV_$\alpha$ | 6328 | 0.020828 |
| VrV_$\alpha$ | 5344 | 0.020254 | VPP_$\alpha$ | 903 | 0.022598 | DEM | 6300 | 0.020736 |
| VPP_$\alpha$ | 5027 | 0.019052 | DEM | 882 | 0.022072 | VPP_$\alpha$ | 5930 | 0.019518 |
| NNH | 4107 | 0.015565 | QTF | 680 | 0.017017 | NNH | 4528 | 0.014904 |
| NNQ | 3622 | 0.013727 | CD | 540 | 0.013514 | NNM | 4196 | 0.013811 |

[3] *ịsụ ude* "to breath heavily in pain"

[4] This is extensional suffix marker attached to a tag to indicate words that are morphologically-inflected.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| QTF | 3165 | 0.011995 | NNQ | 474 | 0.011862 | NNQ | 4096 | 0.013482 |
| NNM | 2955 | 0.011199 | NNH | 421 | 0.010536 | QTF | 3845 | 0.012656 |
| VCJ | 2638 | 0.009998 | ADV | 344 | 0.008609 | VIF | 2665 | 0.008772 |
| VIF | 2339 | 0.008865 | VIF | 326 | 0.008158 | VCJ | 2662 | 0.008762 |
| BCN | 1084 | 0.004108 | VPERF | 200 | 0.005005 | VIF_$\alpha$ | 1259 | 0.004144 |
| VIF_$\alpha$ | 1061 | 0.004021 | VIF_$\alpha$ | 198 | 0.004955 | BCN | 1141 | 0.003756 |
| ADJ | 934 | 0.003540 | CJN_$\alpha$ | 186 | 0.004655 | ADJ | 1090 | 0.003588 |
| BPRN | 885 | 0.003354 | ADJ | 156 | 0.003904 | BPRN | 925 | 0.003045 |
| NND | 671 | 0.002543 | NND | 147 | 0.003679 | ADV | 853 | 0.002808 |
| PRNYNQ | 535 | 0.002028 | PRNREF | 111 | 0.002778 | NND | 818 | 0.002692 |
| WH | 532 | 0.002016 | VAX_$\alpha$ | 98 | 0.002452 | WH | 616 | 0.002028 |
| ADV | 509 | 0.001929 | NNCV | 92 | 0.002302 | PRNREF | 586 | 0.001929 |
| PRNREF | 475 | 0.001800 | NNCC | 91 | 0.002277 | VPERF | 560 | 0.001843 |
| VAX_$\alpha$ | 431 | 0.001633 | WH | 84 | 0.002102 | PRNYNQ | 536 | 0.001764 |
| VPERF | 360 | 0.001365 | IDEO | 70 | 0.001752 | VAX_$\alpha$ | 529 | 0.001741 |
| VCO | 344 | 0.001304 | TTL | 66 | 0.001652 | CJN_$\alpha$ | 447 | 0.001471 |
| PRNEMP | 340 | 0.001289 | ENC | 64 | 0.001602 | NNCV | 394 | 0.001297 |
| NNCC | 302 | 0.001145 | BCN | 57 | 0.001426 | VCO | 394 | 0.001297 |
| NNCV | 302 | 0.001145 | VGD | 56 | 0.001401 | NNCC | 393 | 0.001294 |
| VAX_B | 297 | 0.001126 | VCO | 50 | 0.001251 | PRNEMP | 348 | 0.001145 |
| CJN_$\alpha$ | 261 | 0.000989 | VMOC | 46 | 0.001151 | VAX_B | 312 | 0.001027 |
| VPERF_$\alpha$ | 206 | 0.000781 | VMOV | 46 | 0.001151 | VGD | 261 | 0.000859 |
| VGD | 205 | 0.000777 | BPRN | 40 | 0.001001 | VPERF_$\alpha$ | 215 | 0.000708 |
| INTJ | 194 | 0.000735 | FW | 35 | 0.000876 | INTJ | 213 | 0.000701 |
| VSI_B$\alpha$ | 142 | 0.000538 | VCJ | 24 | 0.000601 | IDEO | 162 | 0.000533 |
| FW | 122 | 0.000462 | INTJ | 19 | 0.000475 | FW | 157 | 0.000517 |
| ADV_$\alpha$ | 100 | 0.000379 | CJN1 | 16 | 0.000400 | ENC | 154 | 0.000507 |
| IDEO | 92 | 0.000349 | CJN2 | 16 | 0.000400 | VSI_B$\alpha$ | 147 | 0.000484 |
| VrV_B | 91 | 0.000345 | LTT | 15 | 0.000375 | VMOC | 117 | 0.000385 |
| VrV_B$\alpha$ | 91 | 0.000345 | VAX_B | 15 | 0.000375 | VMOV | 117 | 0.000385 |
| ENC | 90 | 0.000341 | VPERF_$\alpha$ | 9 | 0.000225 | ADV_$\alpha$ | 100 | 0.000329 |
| VSI_B | 90 | 0.000341 | PRNEMP | 8 | 0.000200 | VrV_B | 97 | 0.000319 |
| VMOC | 71 | 0.000269 | VrV_B | 6 | 0.000150 | VSI_B | 96 | 0.000316 |
| VMOV | 71 | 0.000269 | VAXPRN | 6 | 0.000150 | VrV_B$\alpha$ | 94 | 0.000309 |
| DEM_$\alpha$ | 70 | 0.000265 | VSI_B | 6 | 0.000150 | DEM_$\alpha$ | 72 | 0.000237 |
| WH_$\alpha$ | 67 | 0.000254 | VSI_B$\alpha$ | 5 | 0.000125 | WH_$\alpha$ | 69 | 0.000227 |
| VCO_$\alpha$ | 63 | 0.000239 | VrV_B$\alpha$ | 3 | 0.000075 | TTL | 66 | 0.000217 |
| VPERF_B | 45 | 0.000171 | ABBR | 3 | 0.000075 | VCO_$\alpha$ | 63 | 0.000207 |
| CJN1 | 28 | 0.000106 | VPERF_B | 3 | 0.000075 | VPERF_B | 48 | 0.000158 |
| CJN2 | 28 | 0.000106 | WH_$\alpha$ | 2 | 0.000050 | CJN1 | 44 | 0.000145 |
| CURN | 24 | 0.000091 | VAX_B$\alpha$ | 2 | 0.000050 | CJN2 | 44 | 0.000145 |
| VAX_B$\alpha$ | 23 | 0.000087 | DEM_$\alpha$ | 2 | 0.000050 | VAX_B$\alpha$ | 25 | 0.000082 |
| PREP_XS | 22 | 0.000083 | CURN | 1 | 0.000025 | CURN | 25 | 0.000082 |
| VPERF_B$\alpha$ | 17 | 0.000064 | PRNYNQ | 1 | 0.000025 | VAXPRN | 22 | 0.000072 |
| VAXPRN | 16 | 0.000061 | | | | PREP_$\alpha$ | 22 | 0.000072 |
| QTF_$\alpha$ | 3 | 0.000011 | | | | VPERF_B$\alpha$ | 17 | 0.000056 |
| | | | | | | LTT | 15 | 0.000049 |
| | | | | | | ABBR | 3 | 0.000010 |
| | | | | | | QTF_$\alpha$ | 3 | 0.000010 |
| 63 | 263856 | 1.0 | 61 | 39960 | 1.0 | 66 | 303816 | 1.0 |

where $\alpha$ is XS– represents tokens with prefix/suffix, **B**$\alpha$ is BPRN_XS– represents tokens inflected by suffix and prefix. Latter is due to pronoun bound and, Refer to appendix A for further details on the tagset.

Table 8.5: Tags, frequency, and probability distribution table of all corpus data

## 8.4 Experimental Setup

To evaluate the ability of taggers, we used cross-validation to estimate how accurately they will perform in practice. Therefore in order to determine the accuracy scores, we performed 10-fold cross validation on each corpus and compute the average. We used nine of the ten folds (90%) as a known tagged text on which training was run, while the remaining 10% is an unknown but similar text against which trained taggers were tested for prediction. This is for the purpose of estimating the prediction power of the developed

taggers. Taggers trained and tested on similar texts will predict better than when tested on dissimilar texts.

Table 8.6 shows the average sizes of training and testing data. Corpora in table 8.7 are used to discuss the problem of tagging dissimilar texts in Igbo.

| Corpus | Sentence | Token |
|---|---|---|
| IgbTNT[a] | 8219 | 263856 |
| IgbTNT1[b] | 1220 | 39931 |
| IgbTMT[c] | 2032 | 39960 |
| IgbTC[d] | 10251 | 303816 |
| IgbTC1[e] | 3252 | 79892 |

[a]New Testament part of Igbo Tagged Corpus.
[b]A portion of New Testament Corpus comparable to IgbTMT size.
[c]Novel texts part of Igbo Tagged Corpus to represent modern Igbo texts.
[d]Igbo Tagged Corpus.
[e]Combination of IgbTNT1 and IgbTMT, which is a portion of Igbo Tagged Corpus.

Table 8.6: General statistics of the IgbTC corpora used in this experiments

| Corpus | Train | Test | # Test Sentence |
|---|---|---|---|
| IgbTNT | 237470 | 26386 | 821 |
| IgbTNT1[a] | 35938 | 3993 | 122 |
| IgbTMT | 35965 | 3996 | 203 |
| IgbTNT1→IgbTMT[b] | 35938 | 3996 | 203 |
| IgbTMT→IgbTNT1 | 35965 | 3993 | 122 |
| IgbTC1→IgbTMT | 71902 | 3996 | 203 |
| IgbTC1→IgbTNT1 | 71902 | 3993 | 122 |
| IgbTC1 | 71902 | 7989 | 325 |
| IgbTC | 273434 | 30382 | 1025 |

[a]List of Bible books were selected to form IgbTNT1 corpus, a comparable size of IgbTMT for fair comparison of both corpora
[b]Train tagger on IgbTNT1 and tested on IgbTMT. Use of different styles of texts.

Table 8.7: Average sizes of train, test, and sentence of Igbo corpus data used in this experiments

## 8.5 Experiment and Performance Evaluation

We evaluate the taggers' performance based on the following criteria: (1) the effectiveness of the tagging techniques in Igbo compared with the rich-resourced languages the taggers have been tested on, and (2) investigate the justification of having some of the tags in IgbTC marked as inflectional tags. There are two types of tags used in IgbTC: *normal tag* ($t$) that indicates words not morphologically-inflected (e.g. "bịa/VSI"), and *morphologically-inflected tag* ($t$_XS) are tags marked to indicate morphologically-inflected words (e.g. "bịakwaghị/VSI_XS").

### 8.5.1 Baseline Experiment

We started this evaluation experiments with the baseline tagging accuracy figures in tables 8.8 and 8.9. The essence of this tagging accuracy is to set the lower bound which the generated taggers have to achieve. The baseline tagger is based on the unigram tagging system where a token is classified by its most frequent tag in a training corpus. Tables 8.9 and 8.8 show the accuracy scores of training and testing on similar and dissimilar Igbo texts.

| Corpus | Train size | Test size | Unknown ratio | Unknown acc | known acc | Overall acc | Error rate |
|---|---|---|---|---|---|---|---|
| IgbTNT | 237470 | 26386 | 1.18% | 8.22% | 94.47% | 93.17% | 6.83% |
| IgbTMT | 35965 | 3996 | 4.90% | 18.66% | 93.25% | 89.17% | 10.83% |
| IgbTC | 273434 | 30382 | 1.39% | 9.20% | 93.94% | 92.75% | 7.25% |

Table 8.8: Average statistics and scores of the baseline tagging

| Corpus | Train size | Test size | Unknown ratio | Unknown acc | known acc | Overall acc | Error rate |
|---|---|---|---|---|---|---|---|
| IgbTNT1 | 35938 | 3993 | 3.18% | 12.63% | 94.67% | 91.73% | 8.27% |
| IgbTMT | 35965 | 3996 | 4.90% | 18.66% | 93.25% | 89.17% | 10.83% |
| IgbTC1 | 71902 | 7989 | 3.38% | 14.51% | 92.91% | 90.23% | 9.77% |
| IgbTNT1→IgbTMT | 35938 | 3996 | 16.44% | 28.84% | 87.94% | 78.25% | 21.75% |
| IgbTMT→IgbTNT1 | 35965 | 3993 | 14.70% | 20.80% | 89.65% | 79.51% | 20.49% |
| IgbTC1→IgbTMT | 71902 | 3996 | 4.25% | 17.09% | 91.80% | 88.27% | 11.73% |
| IgbTC1→IgbTNT1 | 71902 | 3993 | 2.48% | 9.65% | 93.90% | 91.55% | 8.45% |

Table 8.9: Average statistics and scores of the baseline tagging. This is for the purpose of comparison between the two genres in order to discuss the problem of training and tagging on dissimilar texts

### 8.5.2 Part of Speech (POS) Tagger Experiment

This experiment is based on the taggers default settings. Table 8.10 displays the accuracy scores of all taggers we used (see table 8.1). TnT and HunPOS make use of *word endings* length of 10 as default settings. SLLT generic features is instantiated by arbitrary contexts like extracting either the tag or the word from positions which is relative to the current words. For example, default extractor list in SLLT consists (-1,word), (0,word), (1,word), (-2,tag), (-1,tag), (w0,w-1), (w0,t-1).

Next, we consider methods to improve POS taggers performance by adding feature extraction techniques for processing rare/unknown words. We set length of extracting last/first letters of a word to $n = 5$ for last letters and $n = 1$ for first letter. The longest suffix found in Igbo is 5 character length, and prefix in Igbo is only a single letter long. To prevent clustering of unknown words in one fold, the sentences in each corpus were evenly distributed. Compare the size of unknown words in 8.11 and 8.10. Results in table 8.11 are calculated on the average.

| Corpus | Test size | unknown ratio | Taggers | Unknown Accuracy | known Accuracy | Overall Accuracy | Error Rate |
|---|---|---|---|---|---|---|---|
| IgbTNT | 26386 | 1.53% | Baseline | 8.22% | 94.47% | 93.17% | 6.83% |
| | | | MBT | 8.45% | 94.44% | 93.14% | 6.86% |
| | | | FnTBL | 8.99% | 97.90% | 96.56% | 3.44% |
| | | | HunPOS | 69.35% | 97.50% | 97.08% | 2.92% |
| | | | TnT | 68.12% | 97.27% | 96.83% | 3.17% |
| | | | SLLT | 51.83% | 98.08% | 97.37% | 2.63% |
| IgbTMT | 3996 | 5.45% | Baseline | 18.66% | 93.25% | 89.17% | 10.83% |
| | | | MBT | 18.62% | 93.55% | 89.45% | 10.55% |
| | | | FnTBL | 19.35% | 95.69% | 91.52% | 8.48% |
| | | | HunPOS | 59.70% | 95.74% | 93.74% | 6.26% |
| | | | TnT | 58.67% | 95.61% | 93.57% | 6.43% |
| | | | SLLT | 47.42% | 96.24% | 93.55% | 6.45% |
| IgbTC | 30382 | 1.72% | Baseline | 9.20% | 93.94% | 92.75% | 7.25% |
| | | | MBT | 10.02% | 93.83% | 92.40% | 7.60% |
| | | | FnTBL | 10.37% | 97.37% | 95.89% | 4.11% |
| | | | HunPOS | 70.50% | 96.86% | 96.42% | 3.58% |
| | | | TnT | 69.49% | 96.63% | 96.17% | 3.83% |
| | | | SLLT | 51.76% | 97.49% | 96.71% | 3.30% |

Table 8.10: Average statistics and scores based on taggers default setting

### 8.5.3 Discussions on Tagging Experiments

Comparing tables 8.10 and 8.11, unknown words accuracy scores show that word feature extraction length of 5 does better than length of 10 used in TnT and HunPOS default settings. The 5 character length is the longest length of affixes in Igbo. SLLT performance benefited from the use of *variable suffix lengths* starting from 1 up to the maximum length of 5 and prefix length of 1[5], while TnT and HunPOS used only a fixed length of 5. For example, words like *nwukwasịkwara*, taggers like TnT that uses *fixed length* for feature extraction will extract "wara". While SLLT that uses *variable suffix lengths* up to the maximum length $n$ will extract a list of strings ["a","ra","ara","wara"] for suffix, and ["n"] for prefix. SLLT in table 8.11 outperformed other taggers on unknown words accuracy with several points.

Results in table 8.10 show unknown, known and overall accuracy scores of all used taggers based on their default settings. HunPOS scored best on unknown words in all cases and overall best in IgbTMT, and SLLT on known words and overall best in IgbTNT and IgbTC. TnT and HunPOS are running neck-and-neck in unknown, known and overall words scores. Recall that HunPOS is a reimplementation of TnT which may be one of the reasons behind their close accuracy score ties. FnTBL does well only on the known and overall words, while MBT scored lowest. Tagging accuracy for FnTBL and MBT are relatively low compared to other taggers.

In the second experiment of table 8.11, we considered word feature extractors for processing unknown/rare words as a method to improve POS taggers performance. Accuracy scores reveal that all taggers performance on known words are commendable but not good enough on the unknown words. Generally, the overall scores are good despite the low

[5]We used length of 1 because prifix in Igbo uses vowel class of 8 elements, each element is one character long. See table 9.13.

| Corpus | Test size | unknown ratio | Taggers | Unknown Accuracy | known Accuracy | Overall Accuracy | Error Rate |
|---|---|---|---|---|---|---|---|
| IgbTNT | 26386 | 1.18% | Baseline | 8.22% | 94.47% | 93.17% | 6.83% |
| | | | MBT | 7.13% | 94.50% | 93.47% | 6.53% |
| | | | FnTBL | 7.69% | 98.03% | 96.97% | 3.03% |
| | | | HunPOS | 74.48% | 97.66% | 97.38% | 2.62% |
| | | | TnT | 75.47% | 97.35% | 97.10% | 2.90% |
| | | | $SLLT^{s}$ | 81.08% | 98.21% | 98.01% | 1.99% |
| | | | $SLLT^{SP}$ | 83.95% | 98.22% | 98.05% | 1.95% |
| | | | $SLLT^{*}$ | 83.43% | 98.28% | 98.11% | 1.89% |
| IgbTMT | 3996 | 4.90% | Baseline | 18.66% | 93.25% | 89.17% | 10.83% |
| | | | MBT | 17.13% | 93.40% | 89.66% | 10.34% |
| | | | FnTBL | 17.73% | 95.80% | 91.97% | 8.03% |
| | | | HunPOS | 61.86% | 95.86% | 94.18% | 5.82% |
| | | | TnT | 63.73% | 95.61% | 94.03% | 5.97% |
| | | | $SLLT^{s}$ | 70.76% | 96.24% | 94.97% | 5.03% |
| | | | $SLLT^{SP}$ | 78.48% | 96.21% | 95.33% | 4.67% |
| | | | $SLLT^{*}$ | 78.40% | 96.39% | 95.50% | 4.50% |
| IgbTC | 30382 | 1.39% | Baseline | 9.20% | 93.94% | 92.75% | 7.25% |
| | | | MBT | 9.54% | 93.91% | 92.72% | 7.28% |
| | | | FnTBL | 9.96% | 97.51% | 96.28% | 3.72% |
| | | | HunPOS | 71.65% | 97.03% | 96.67% | 3.33% |
| | | | TnT | 72.49% | 96.71% | 96.37% | 3.63% |
| | | | $SLLT^{s}$ | 77.27% | 97.64% | 97.36% | 2.64% |
| | | | $SLLT^{SP}$ | 81.27% | 97.64% | 97.41% | 2.59% |
| | | | $SLLT^{*}$ | 81.30% | 97.78% | 97.55% | 2.45% |

Table 8.11: Average statistics and accuracy scores based on additional settings provided in the taggers architecture (such as word endings, surrounding words, etc.). SLLT with *s* means generic setting with suffix for word feature extraction, *sp* is setting with suffix and prefix, and * means combination of both (*s* and *sp*) and other features extraction parameters

performance of the taggers on the unknown words, which can be credited to the small size of unknown words. Therefore, it's hard to generalize about taggers performance on the Igbo texts because of their overall accuracy scores. The tagging accuracy scores obtained by the best performing tagger *SLLT* on the overall words are comparable to 97.24% scored in English Penn TreeBank. But unknown words accuracy scores are considerably lower as compared to 89.04% scored in English Penn TreeBank. TnT scored better than HunPOS in unknown words, while HunPOS is better in known words with slight differences. The taggers performance scores are low in IgbTMT compared to IgbTNT, which could be caused by the IgbTMT unknown words size and text style. The IgbTMT unknown words size is greater than IgbTNT by 1.72% indicating taggers difficulty in tagging modern Igbo texts probably because of new words.

*Ambiguous tokens* in table 8.3 and the accuracy scores in table 8.11 reveal that out of 29.73% ambiguous tokens in IgbTNT, *MBT* tagger correctly classified 22.90%, which is added to the 70.27%[6] tokens with only one tag to make the overall accuracy score. Also compare other taggers performance on disambiguating the *ambiguous tokens* in table 8.3.

We split SLLT tagging into three variations using different word feature settings for

[6]All the taggers are suppose to get this score for free.

| Prefix | Meaning |
|---|---|
| a/e | indicates verb is participle if preceded by auxiliary |
| n/m | indicates noun or gerund formed through nominalization |
| ị/i | indicates infinitive verb |
| o/ọ | indicates noun or gerund formed through nominalization |

Table 8.12: Igbo prefixes and their meaning

| | | | | | Accuracy | | | |
|---|---|---|---|---|---|---|---|---|
| Corpus | Test size | Inflected Token Size | unknown ratio | Taggers | Unknown acc | known acc | Overall acc | Error Rate |
| IgbTNT | 26386 | 3138 | 7.67% | MBT | 00.00% | 95.34% | 88.02% | 11.98% |
| | | | | FnTBL | 00.00% | 98.11% | 90.59% | 9.41% |
| | | | | HunPOS | 78.22% | 98.13% | 96.61% | 3.39% |
| | | | | TnT | 80.30% | 97.87% | 96.54% | 3.46% |
| | | | | $SLLT^{s}$ | 84.79% | 98.31% | 97.28% | 2.72% |
| | | | | $SLLT^{SP}$ | 87.26% | 98.30% | 97.46% | 2.54% |
| | | | | $SLLT^{*}$ | 87.04% | 98.32% | 97.46% | 2.54% |
| IgbTNT1 | 3993 | 461 | 19.37% | MBT | 00.00% | 95.45% | 76.92% | 23.08% |
| | | | | FnTBL | 00.00% | 97.15% | 78.28% | 21.72% |
| | | | | HunPOS | 70.71% | 97.60% | 92.38% | 7.62% |
| | | | | TnT | 73.67% | 97.61% | 92.96% | 7.04% |
| | | | | $SLLT^{s}$ | 80.49% | 97.87% | 94.48% | 5.52% |
| | | | | $SLLT^{SP}$ | 85.37% | 97.90% | 95.42% | 4.58% |
| | | | | $SLLT^{*}$ | 83.86% | 97.92% | 95.14% | 4.86% |
| IgbTMT | 3996 | 562 | 24.06% | MBT | 00.00% | 97.74% | 74.19% | 25.81% |
| | | | | FnTBL | 00.00% | 98.68% | 74.90% | 25.10% |
| | | | | HunPOS | 66.88% | 98.61% | 90.93% | 9.07% |
| | | | | TnT | 70.12% | 98.62% | 91.73% | 8.27% |
| | | | | $SLLT^{s}$ | 77.21% | 98.56% | 93.40% | 6.60% |
| | | | | $SLLT^{SP}$ | 86.65% | 98.51% | 95.64% | 4.36% |
| | | | | $SLLT^{*}$ | 86.42% | 98.60% | 95.66% | 4.34% |

Table 8.13: Average statistics and accuracy scores based on tokens that are morph-inflected in the test data. SLLT with *s* means generic with suffix, *sp* is with suffix and prefix and * means combination of both (*s* and *sp*) and other features. The inflected token size is the number of words in test size that have tags with extensional suffix marker *XS* (compare with table 8.3)

suffix and prefix, viz; $SLLT^{s}$ means only suffix feature added, $SLLT^{sp}$ means suffix and prefix features added, and $SLLT^{*}$ means suffix, prefix and other features, such as word shapes[7] added. From table 8.11, performance scores reveal that SLLT performed best on the overall words when in $SLLT^{*}$ configuration, but performed best on the morph-inflected unknown words and unknown words when in $SLLT^{sp}$ configuration. $SLLT^{s}$ configuration negatively affect the general performance of the tagger. Toutanova et al. (2003) empirically observed that the prefix features for rare words were having a net negative effect on the accuracies, that the removal of it considerably increased the unknown and overall words accuracies in the Penn TreeBank English corpus. Conversely, SLT tagger's results using

[7]Features used to represent the abstract letter pattern of a word by mapping lower-case letters to 'x', upper-case to 'X', numbers to 'd', and retaining punctuation Jurafsky and Martin (2014).

$SLLT^{sp}$ configuration show that addition of prefix feature improved accuracy on the unknown words by 2.87%, 7.72% and 4.00%, which positively affects the overall accuracy in the Igbo corpora. This indicates that prefix[8] in Igbo is a good predictive element despite it is a single character long. Also see table 8.13 for more results on how addition of prefix improved the accuracy of morphologically-inflected (morph-inflected) words.

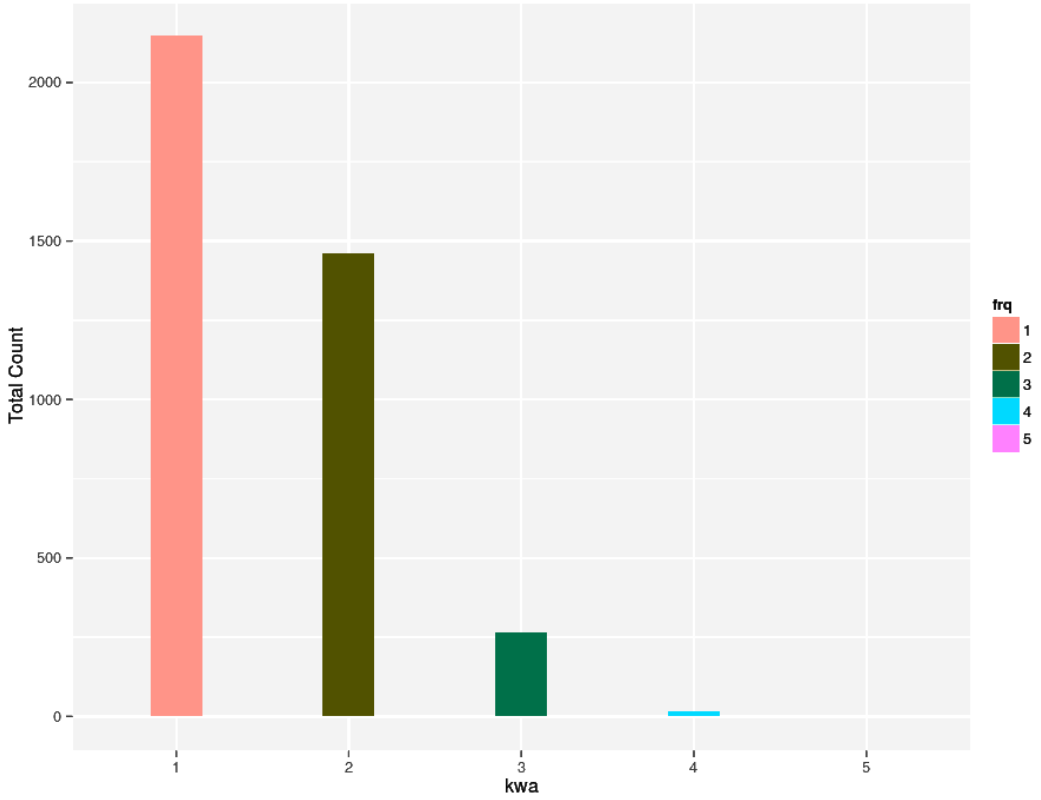



Figure 8.2: Frequency of an Igbo suffix *kwa* occurring at different positions immediately after the stem of an inflected word. 1 means a suffix position immediately after stem. If *kwa* is found at position 3 of an inflected word, it means there are other two suffixes found at positions 1 and 2. *kwa* is use in this figure as the last suffix of an inflected word. As the position of *kwa* moves to the right, the frequency of words with *kwa* and other suffixes before *kwa* decreases

Table 8.13 shows the performance scores of the taggers on the morph-inflected tokens for known, unknown and overall. The unknown inflected tokens are mainly morphologically-complex (morph-complex) ones that are rare. They are less frequent due to attachment of more number suffixes or use of rare suffixes. The frequent known morph-inflected tokens are the ones with a single or few suffixes. Figure 8.2 shows that a word becomes rare when it contains more number of suffixes. In Arabic, the highest rate of ambiguity appeared at the stem level, but decreases with inflection, and even decreases further when clitics are added (Attia, 2008). That is to say, the more ambiguous a word is, the more frequent it becomes. Also the less number of suffixes in a word, the more frequent that word will be

[8]We observe that morph-inflected words with a prefix constitute 4.60% of IgbTNMT corpus.

and vice versa. For example, the classes past tense (VrV) and perfect tense (VPERF) verbs undergo affixation through inflection[9] and they can be more complex with addition of extensional suffixes[10]. In both cases, they are morph-inflected words with the latter less frequent.

Figure 8.2 shows the frequency of Igbo suffix "kwa" attached at different positions of a word, counts start immediately after a stem. The frequency of words with "kwa" decreases as the position number increases. For example, in *bịakwa* "come also", the suffix "kwa" occurred at position 1; *bịagokwa*'s suffixes "go" and "kwa" occurred at positions 1 and 2. This indicates that a word becomes rare or unknown with more suffixes (like bars from 3 upwards in the figure). The accuracy scores of the taggers on the unknown words shows the complexity in tagging morph-complex tokens, such as tokens found in bars 4 and 5 of figure 8.2.

### Macro-Averages of Tags

In tagging system, taggers use histories in order to disambiguate focus words correctly. Recall in transition probability, Hidden Markov Model (HMM) based taggers use previous disambiguated tags to decide correct tag for the current ambiguous tags/words. This implies that if a relevant tag is missing or wrongly assigned at some point, a tagger may find it difficult disambiguating an ambiguous case correctly at that point, and this could degenerate tagger's performance over time.

Macro-averaging of tags calculates taggers performances on individual tag where every tag is treated with equal weight while micro-averaging of tags calculates taggers performances on individual tag where every tag is treated with different weight. The latter is equal to accuracy in POS tagging task as shown in section 8.2. Taggers' macro-averaging scores in table 8.14 is much less compared with table 8.11. A possible explanation of the poor macro-averaging scores is that taggers mostly identify correctly tags with high frequency of occurrence than the less frequent ones. Also, frequent tags in the *ambiguous tokens'* tag set are correctly classified more than the less frequent ones (see table 8.4 for tokens with more than 5 unique tags). Figure 8.3 is confusion matrices of tagging errors made by taggers on words having 6 or more unique tags.

### Word Level Accuracy Analysis

We look at performance of taggers based on tags assigned to tokens with high number of unique tags. We evaluated this using two most frequent words with high number of unique tags. From table 8.4, in IgbTC section, we selected "ahụ" and "ike" and calculated confusion matrices of how taggers classified them according to their unique tags. Figure 8.3 shows the resultant matrices, on top of each matrix are the truth tags while on

[9]This is inflectional class usually attached to a verb to express various forms of temporal relations of an event as either presently happening, already happened or still to happen. See inflectional class in the tagset table on appendix A for examples.

[10]We added XS to differentiate the former from the latter. Past tense in Igbo is marked with the addition of letter "r" and the harmonizing vowel "V" (-rV) while perfect tense is letter "g" and harmonizing vowel "V" (-gV). Illustrating this: *bịa*, *bịago*, *bịara*, *bịagoro*, *bịakwara*, *bịagokwara*, *bịachikwara*, *bịaghachikwara*, *bịaghachigoro*,*bịaghachigokwara* and so on. The first to five examples are frequently used while last five examples are less frequently used.

| | | Macroaverage | | |
|---|---|---|---|---|
| corpus | Tagger | MacroP | MacroR | $f$-sc |
| IgbTNT | MBT | 70.686 | 78.707 | 74.481 |
| | FnTBL | 84.259 | 89.940 | 87.007 |
| | HunPOS | 87.374 | 89.163 | 88.259 |
| | TnT | 88.065 | 87.934 | 88.000 |
| | SLLT | 86.350 | 89.986 | 88.131 |
| IgbTNT1 | MBT | 59.703 | 65.216 | 62.338 |
| | FnTBL | 69.114 | 74.711 | 71.804 |
| | HunPOS | 72.978 | 75.036 | 73.993 |
| | TnT | 75.010 | 75.771 | 75.389 |
| | SLLT | 72.329 | 76.310 | 74.266 |
| IgbTMT | MBT | 50.935 | 58.665 | 54.528 |
| | FnTBL | 59.209 | 67.221 | 62.961 |
| | HunPOS | 65.665 | 69.386 | 67.474 |
| | TnT | 66.224 | 68.870 | 67.521 |
| | SLLT | 64.239 | 68.715 | 66.402 |
| IgbTC | MBT | 67.085 | 75.508 | 71.048 |
| | FnTBL | 80.512 | 86.937 | 83.601 |
| | HunPOS | 83.452 | 85.559 | 84.492 |
| | TnT | 84.527 | 84.780 | 84.653 |
| | SLLT | 82.854 | 87.149 | 84.947 |

Table 8.14: Overall average of macro-averaging of all the tags

the left side are the tags assigned by taggers. From table 8.4, "ahụ" is 93.41% DEM and remaining 6.59% is distributed over other tags (NNC, NNCC, NNH, VPP, VSI). MBT tagger classified all "ahụ" as DEM since the frequency of "ahụ" functioning as DEM is very high. Thus, for MBT classifying "ahụ" as DEM, recall (R) is 100% and precision (P) lower at 93.41%[11], while in other labels, R and P are 0. Across all labels, R=P=A(accuracy)=93.41%. For SLLT classifying "ahụ" as DEM, R is 99.71%[12] and P is 98.67%[13] respectively, and across all labels P=R=A=98.24%[14]. None of the taggers got "ahụ" as VSI, and FnTBL tagger introduced two tags "BCN" and "PRN" which none of other taggers used and are not among the truth tags. Both cases are likely triggered by use of contexts. Compare this figure with figure 8.4, for example, observe the high rate of confusion that exist in tagging between "DEM and NNC" and "DEM and VPP".

[11] $\frac{93.41}{93.41+6.59}$, where 93.41 is true positive (TP) and 6.59 is false positive (FP).
[12] $\frac{93.14}{93.14+0.27}$, where 93.14 is TP and 0.27 is false negative (FN).
[13] $\frac{93.14}{93.14+1.25}$, where 93.14 is TP and 1.25 is FP.
[14] Sum of diagonals in SLL matrix: 93.14+1.08+0.07+0.02+3.93+0.

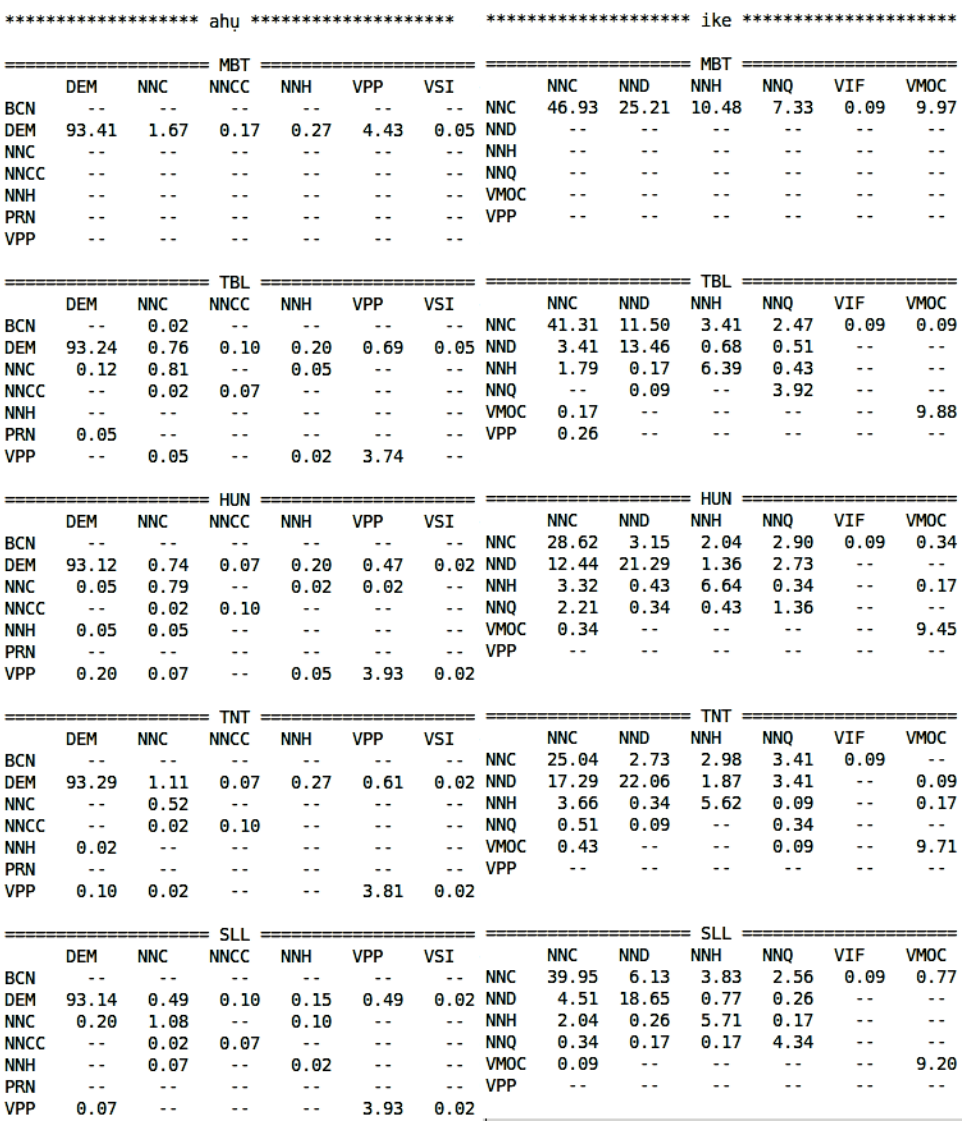



Figure 8.3: Confusion matrices of tagging errors made by taggers on some Igbo words with high number of unique tags. SLL in this figure is $SLLT^*$

## 8.6 Comparison Between Different Genres

This section discusses the problem of tagging dissimilar texts in Igbo using the corpora in table 8.7. The results are summarized in table 8.15, and from the scores we observe

| | | | | Accuracy | | | |
|---|---|---|---|---|---|---|---|
| corpus | Test size | unknown ratio | Taggers | Unknown Acc Score | known Acc Score | Overall Acc Score | Error rate |
| IgbTNT1 | 3993 | 3.18% | Baseline | 12.63% | 94.67% | 91.73% | 8.27% |
| | | | MBT | 12.71% | 94.64% | 92.03% | 7.97% |
| | | | FnTBL | 13.91% | 97.29% | 94.63% | 5.37% |
| | | | HunPOS | 64.87% | 97.33% | 96.31% | 3.69% |
| | | | TnT | 66.26% | 97.27% | 96.29% | 3.71% |
| | | | SLLT | 76.08% | 97.66% | 96.97% | 3.03% |
| IgbTMT | 3996 | 4.90% | Baseline | 18.66% | 93.25% | 89.17% | 10.83% |
| | | | MBT | 17.13% | 93.40% | 89.66% | 10.34% |
| | | | FnTBL | 17.73% | 95.80% | 91.97% | 8.03% |
| | | | HunPOS | 61.86% | 95.86% | 94.18% | 5.82% |
| | | | TnT | 63.73% | 95.61% | 94.03% | 5.97% |
| | | | SLLT | 78.40% | 96.39% | 95.50% | 4.50% |
| IgbTNT1→IgbTMT | 3996 | 16.44% | Baseline | 28.84% | 87.94% | 78.25% | 21.75% |
| | | | MBT | 29.48% | 88.00% | 78.38% | 21.62% |
| | | | FnTBL | 30.39% | 90.04% | 80.23% | 19.77% |
| | | | HunPOS | 49.99% | 89.31% | 82.84% | 17.16% |
| | | | TnT | 48.33% | 89.40% | 82.65% | 17.35% |
| | | | SLLT | 53.61% | 90.09% | 84.09% | 15.91% |
| IgbTMT→IgbTNT1 | 3993 | 14.70% | Baseline | 20.80% | 89.65% | 79.51% | 20.49% |
| | | | MBT | 20.84% | 89.84% | 79.69% | 20.31% |
| | | | FnTBL | 21.62% | 90.49% | 80.37% | 19.63% |
| | | | HunPOS | 47.26% | 90.82% | 84.41% | 15.59% |
| | | | TnT | 51.05% | 90.24% | 84.48% | 15.52% |
| | | | SLLT | 44.11% | 90.70% | 83.85% | 16.15% |
| IgbTC1→IgbTMT | 3996 | 4.26% | Baseline | 17.09% | 91.80% | 88.27% | 11.73% |
| | | | MBT | 16.12% | 91.89% | 88.67% | 11.33% |
| | | | FnTBL | 16.58% | 95.12% | 91.78% | 8.22% |
| | | | HunPOS | 62.97% | 95.03% | 93.65% | 6.35% |
| | | | TnT | 65.25% | 94.60% | 93.34% | 6.66% |
| | | | SLLT | 77.35% | 95.73% | 94.94% | 5.06% |
| IgbTC1→IgbTNT1 | 3993 | 2.48% | Baseline | 9.65% | 93.90% | 91.55% | 8.45% |
| | | | MBT | 11.03% | 93.91% | 91.85% | 8.15% |
| | | | FnTBL | 11.49% | 96.57% | 94.46% | 5.54% |
| | | | HunPOS | 69.16% | 96.58% | 95.90% | 4.10% |
| | | | TnT | 69.60% | 96.38% | 95.72% | 4.28% |
| | | | SLLT | 77.98% | 97.08% | 96.61% | 3.39% |
| IgbTC1 | 7989 | 3.39% | Baseline | 14.51% | 92.91% | 90.23% | 9.77% |
| | | | MBT | 14.20% | 92.90% | 90.23% | 9.77% |
| | | | FnTBL | 14.67% | 95.89% | 93.11% | 6.89% |
| | | | HunPOS | 65.05% | 95.79% | 94.75% | 5.25% |
| | | | TnT | 66.42% | 95.48% | 94.49% | 5.51% |
| | | | SLLT | 77.52% | 96.41% | 95.77% | 4.23% |

Table 8.15: Average statistics and accuracy scores of tagging dissimilar Igbo texts.

1. that the overall score for IgbTMT is always lower compared to IgbTNT and IgbTC.

Accuracy scores of taggers on the IgbTNT/IgbTNT1 drop whenever it is combined with IgbTMT (IgbTC and IgbTC1) (see table 8.15). This is an indication of the difference in the text textures between the two genres, which could be that *IgbTMT* text style is more difficult for taggers than *IgbTNT*.

2. that when taggers are trained and tested on dissimilar texts (e.g. IgbTNT1 → IgbTMT), the tagging accuracy scores for taggers dropped by several points, especially on the unknown words, compared to when trained and tested with similar texts. It is even worse when trained on IgbTMT corpus and tested on IgbTNT1 corpus. The performance of taggers is affected by the large ratio of new patterns and tokens encountered in the test data due to different text styles.

3. that the tagging accuracy scores on the unknown words for MBT and FnTBL increase. This is because of increase in the size of nouns within the unknown words for dissimilar texts.

4. that when IgbTNT1 and IgbTMT are combined (IgbTC1), there is a substantial drop in the unknown words ratio, and the accuracy scores increase. The tagging accuracy decreases as ratio of unknown words increases.

5. Unknown words in English have a high proportion of proper nouns. By contrast, the major causes of increase in unknown words for Igbo is because of nouns (names of things) and morphologically-inflected words. Morphologically-inflected words are found in nouns, verbs, and other classes with majority in verbs. Tseng et al. (2005) examines the problem of POS tagging of different varieties of Mandarin Chinese and found out that major cause of unknown words when POS tagging across different genres in Mandarin Chinese is not proper nouns but mainly morphological inflections of words.

6. Also, see observation in section 9.4.5 of chapter 9.

## 8.7 Comparative Analysis with Other Languages

This section handles one of the goals of evaluating taggers performance highlighted in sections 8.2 and 8.5. It is of good interest to know how automatic part-of-speech (POS) taggers perform on a specific language since there are different POS tagging techniques they use (e.g. decision tree, transformation, HMM based techniques), and which kind of taggers performs best may depend on a language corpus. This comparison will help to justify the best choice of tagger, the good and bad aspect of them considering Igbo language. We compare how the taggers' performance on Igbo language different from other languages they have been tested on. This comparison will be based on the tagging results in tables 8.11 and the taggers' accuracy scores achieved in other languages they have been tested on. We summarize our observations as follows:

- SLLT scored 97.24% on the overall tokens, and considerably high score of 89.04% on the unknown words for English Penn treebank corpus (Toutanova et al., 2003). Compared to the unknown words score in Igbo corpora (see table 8.11), SLLT accuracy score in English is better by several points.

- TnT tagger (Brants, 2000b) results on German NEGRA corpus are 96.7% and 89.0% for the overall and unknown words. TnT unknown words score in German is better than all its unknown words accuracy scores in Igbo. TnT unknown words accuracy score in the NEGRA corpus is by 13.53% better than its best performing score of 75.47% in IgbTNT. The overall accuracy scores in IgbTNT and IgbTC corpora are commendable. These accuracies are remarkably good compared to accuracy scores of 71.68% (unknown words) and 90.44% (overall) in Icelandic texts[15] (Loftsson, 2007). In Hungarian texts, TnT scored 97.42% on overall accuracy (Halácsy et al., 2006), which is close to 97.10% overall accuracy score in IgbTNT. Also, TnT recorded overall accuracy score of 96.46% on Penn TreeBank English corpus comparable to 96.37% accuracy score on IgbTC corpus.

- According to Halácsy et al. (2007), HunPOS Trigram Tagger results on Penn Treebank English corpus are 96.49% and 86.90% for overall and unknown words. On hungarian texts, it achieved accuracy scores of 98.24% and 95.96% for overall and unknown words. HunPOS unknown words accuracy scores in English and Hungarian texts are better than its best tagging accuracy scores in Igbo corpora by several points.

- FnTBL tagger performance score on Penn Treebank WSJ is 96.76% for overall words (Ngai and Florian, 2001), and accuracy scores of 55.51% and 89.33% for unknown and overall words scores on the Icelandic texts (Loftsson, 2007). It performs poorly in unknown words for Igbo compared to Icelandic unknown words accuracy score, but performance scores on the overall and known words are considerably high.

- Memory-based tagger results on WSJ corpus are 96.4% and 90.6% for overall and unknown words (Daelemans et al., 1996). MBT achieves lowest accuracy scores in all experiments compared to other taggers (except baseline). Accuracy scores of

[15]Icelandic used 639 tags in Icelandic frequency dictionary corpus.

MBT and baseline are almost the same in most cases. MBT achieved a remarkable score on the unknown words in English texts than in Igbo. Its performance in Icelandic texts is almost same with FnTBL in overall score but better than FnTBL in unknown words with 59.40% accuracy score (Loftsson, 2007). Conversely, in Igbo corpus FnTBL performs better than MBT in all experiments.

- Generally, the overall words accuracy scores are good despite the low performance of the taggers on the unknown words, which can be credited to the small size of unknown words. Therefore, it's hard to generalize about taggers performance on the Igbo texts because of their overall accuracy scores. Best performing tagger among all of them is SLLT, especially in handling unknown words.

- Training size, number of tags, and rate of ambiguity for tags and tokens can affect the tagger's performance. The size of the Igbo corpora we used might contribute to the poor scores of these taggers on the unknown Igbo words. For instance, the training size of IgbTC corpus (see tables 8.3 and 8.7) is about 30% of the size of Penn Treebank and Hungarian corpora used in the design and development of the taggers used in this experiment. However, it is about 86% corpus size of NEGRA used by Brants in the design and development of TnT. The Penn Treebank, an English corpus, consists of approximately 1.2 million tokens, 50000 sentences, and 57962 word types (containing 13% ambiguous types and 76% ambiguous tokens) (Brants, 2000b; Daelemans et al., 1996; Toutanova et al., 2003; Halácsy et al., 2007), while Hungarian corpus contains 1161015 tokens and 70083 sentences (Halácsy et al., 2007). NEGRA is a German corpus that consists of 355000 tokens and 20000 sentences (Brants, 2000b).

## 8.8 Most Frequent Tagging Errors

There are two different ways of grouping errors made by taggers for purpose of discussion: tag type error occurs when tag $t_1$ is proposed by tagger but $t_0$ is the correct tag while word error occurs when a word $w$ is wrongly assigned a tag $t$ by the tagger. Illustrating this with IgbTC, the total number of tags where SLLT tagger proposed $t_1$ instead of $t_0$ is 7458, which if divided by 303816 (total number of tokens in IgbTC) is 2.45% (equivalent to SLLT error rate in table 8.11). In this 7458, the percentage error contributed by SLLT in proposing another tag $t_1$ to be NNC ($t_0$) is 0.484%. The constituents of this 0.484% error are the percentage errors made by SLLT in proposing NNH to be NNC (NNC>NNH) is 0.219%, NNC>NNM is 0.088%, etc. Summary of all the common errors made by the taggers (except MBT) are shown in table 8.16. Compare this table with figure 8.4 that represents the confusion matrix[16] of the most frequent tagging errors made by the taggers. This figure shows that rate of tagging errors that exits between NNC and other tags (especially the noun family) is high. Also, observe in figure 8.4 how tagging errors are clustered in the region of **XS** tags (tags with affixes), this region is mostly morph-inflected words.

[16]We combined the outputs of all taggers, found the most frequent tagging errors and use the statistics to plot the confusion matrix.

$w$ in the word error are most frequent words corresponding to the most frequent tags. Compare word error in table 8.16 with figures 8.4 and 8.3[17] and tables in appendix A.3[18]. The information in this table is valuable for developing more robust tagging system and to certain extent, it requires world or semantic knowledge from human experts to correct these errors properly.

| Tagging Error | | | | | | | |
|---|---|---|---|---|---|---|---|
| SLLT | | TnT | | HunPOS | | FnTBL | |
| $t_0 > t_1$ (7458) | Error | $t_0 > t_1$ (11028) | Error | $t_0 > t_1$ (10119) | Error | $t_0 > t_1$ (11298) | Error |
| NNC>NNH | 0.219% | NNC>NNH | 0.498% | NNC>NNH | 0.479% | NNC>NNH | 0.221% |
| NNC>NNM | 0.088% | NNC>NNM | 0.176% | NNC>NNM | 0.151% | NNC>NNM | 0.090% |
| NNC>CJN | 0.037% | NNC>NND | 0.076% | NNC>CJN | 0.064% | NNC>NNQ | 0.047% |
| Total error | 0.484% | Total error | 1.075% | Total error | 1.019% | Total error | 0.538% |
| NNH>NNC | 0.313% | PRN>DEM | 0.276% | NNH>NNC | 0.232% | VSI_XS>NNC | 0.334% |
| NNH>NNQ | 0.015% | PRN>BPRN | 0.026% | NNH>NNQ | 0.014% | VSI_XS>VPP_XS | 0.056% |
| NNH>NND | 0.008% | PRN>PRNYNQ | 0.009% | NNH>NND | 0.012% | VSI_XS>VrV_XS | 0.019% |
| Total error | 0.343% | Total error | 0.320% | Total error | 0.276% | Total error | 0.459% |
| PREP>CJN | 0.127% | NNH>NNC | 0.261% | PREP>CJN | 0.183% | NNH>NNC | 0.370% |
| PREP>VSI_XS | 0.013% | NNH>NND | 0.016% | PREP>VSI_XS | 0.031% | NNH>NNQ | 0.019% |
| PREP>VrV_XS | 0.003% | NNH>NNQ | 0.015% | PREP>VrV_XS | 0.008% | NNH>NND | 0.010% |
| Total error | 0.146 % | Total error | 0.315% | Total error | 0.224% | Total error | 0.428% |
| VSI_XS>VPP_XS | 0.049% | PREP>CJN | 0.253% | PRN>DEM | 0.176% | VPP_XS>VPP | 0.215% |
| VSI_XS>VrV_XS | 0.027% | PREP>VSI_XS | 0.017% | PRN>BPRN | 0.025% | VPP_XS>NNC | 0.050% |
| VSI_XS>NNC | 0.014% | PREP>VrV_XS | 0.008% | PRN>PRNYNQ | 0.008% | VPP_XS>VSI_XS | 0.042% |
| Total error | 0.139% | Total error | 0.281% | Total error | 0.215% | Total error | 0.333% |
| VPP_XS > VSI_XS | 0.077% | VSI_XS>VPP_XS | 0.069% | VSI_XS>VPP_XS | 0.066% | VrV_XS>NNC | 0.259% |
| VPP_XS > BCN | 0.012% | VSI_XS>PREP | 0.037% | VSI_XS>VrV_XS | 0.031% | VrV_XS>VSI_XS | 0.008% |
| VPP_XS > VPP | 0.009% | VSI_XS>VrV_XS | 0.032% | VSI_XS>PREP | 0.028% | VrV_XS>PREP | 0.004% |
| Total error | 0.128% | Total error | 0.224% | Total error | 0.191% | Total error | 0.276% |
| ADV>CJN | 0.073% | VPP_XS>VSI_XS | 0.077% | NNM>NNC | 0.123% | PREP>CJN | 0.133% |
| ADV>NNC | 0.026% | VPP_XS>VPP | 0.013% | VPP_XS > VSI_XS | 0.066% | PREP>VSI_XS | 0.012% |
| ADV>CD | 0.013% | VPP_XS>VAX_XS | 0.012% | VPP_XS > VPP | 0.012% | PREP>VrV_XS | 0.002% |
| Total error | 0.120% | Total error | 0.143% | Total error | 0.122% | Total error | 0.149% |
| Overall total | 2.45% | | 3.63% | | 3.33% | | 3.72% |
| Word Error | | | | | | | |
| $w > t_1$ (7458) | Error | $w > t_1$ (11028) | Error | $w > t_1$ (10119) | Error | $w > t_1$ (11298) | Error |
| na>CJN | 0.127% | a>DEM | 0.276% | a>DEM | 0.176% | na>CJN | 0.133 |
| na>PREP | 0.042% | a>PRN | 0.020% | a>PRN | 0.064% | na>PREP | 0.041 |
| Total error | 0.169% | Total error | 0.296% | Total error | 0.240% | Total error | 0.174% |
| ndi>NNC | 0.091% | na>CJN | 0.252% | na>CJN | 0.182% | a>PRN | 0.067 |
| ndi>NNM | 0.043% | na>PREP | 0.020% | na>PREP | 0.051% | a>DEM | 0.050 |
| Total error | 0.135% | Total error | 0.272% | Total error | 0.233% | Total error | 0.117% |
| ka>CJN | 0.092% | ndi>NNM | 0.092% | ndi>NNC | 0.104% | ndi>NNC | 0.072 |
| ka>ADV | 0.008% | ndi>NNC | 0.085% | ndi>NNM | 0.093% | ndi>NNM | 0.044 |
| Total error | 0.104% | Total error | 0.178% | Total error | 0.197% | Total error | 0.117% |
| Overall total | 2.45% | | 3.63% | | 3.33% | | 3.72% |

Table 8.16: Top most frequent tagging and word errors made by taggers (except MBT). SLLT in this table is $SLLT^*$

[17]Confusion matrix for high frequent words with high number of unique tags.

[18]Contains precisions and recalls for individual tags.

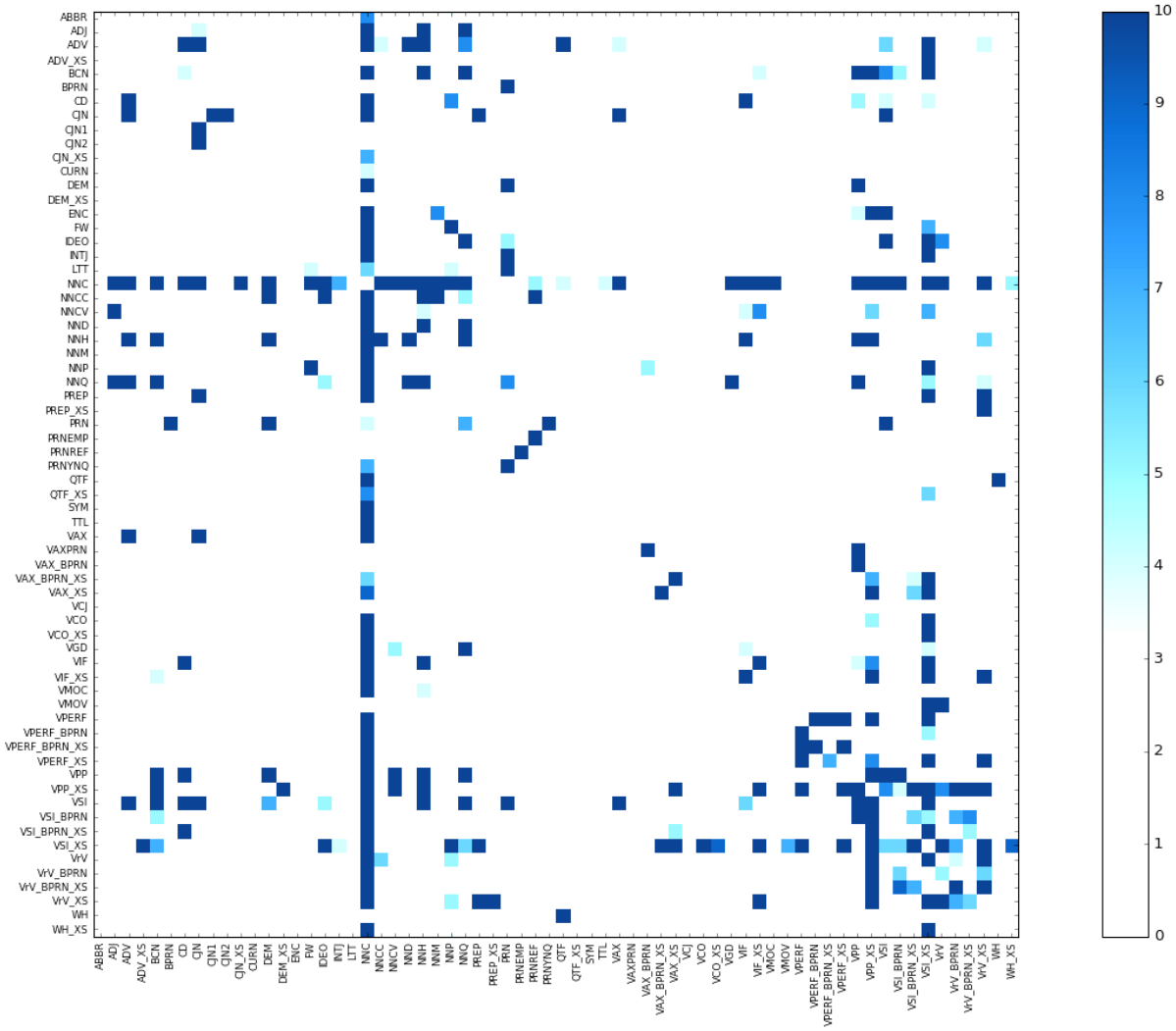



Figure 8.4: Confusion matrix of most frequent tagging errors made by taggers

## 8.9 Tagging on Different Igbo Tagset Granularities

This section justifies one of the criteria highlighted in section 8.5 about tagging on different tag forms. IgbTNT and IgbTMT corpora comprise of different tag forms designed according to the tagset developed in chapter 6. Tags are designed into two parts: *Normal tags* ($t$) for words that are not morphologically-inflected (morph-inflected) and *morph-inflected tags* ($t$_XS) for words that are morph-inflected having one or more affixes. There are 63 and 61 tags used on IgbTNT and IgbTMT. To justify the use of *XS* marker, we evaluate the effect of removing it from the tags. We use IgbTNT and IgbTMT for this experiment. IgbTNT contains 63 tags with 21 of them are marked *XS* ( are in the form $t$_XS), and IgbTMT contains 61 tags with 21 of them are marked *XS*. Figure 8.5 shows the outcomes of four variations of experiments conducted on the IgbTNT and IgbTMT using SLT tagger.
**MG** in the figures (8.5 and 8.6) means that morph-inflected tokens in IgbTNT and IgbTMT are tagged with $t$_XS tag form. Taggers were trained, tested, and evaluated on the corpus without removing *XS*. See results in table 8.11.

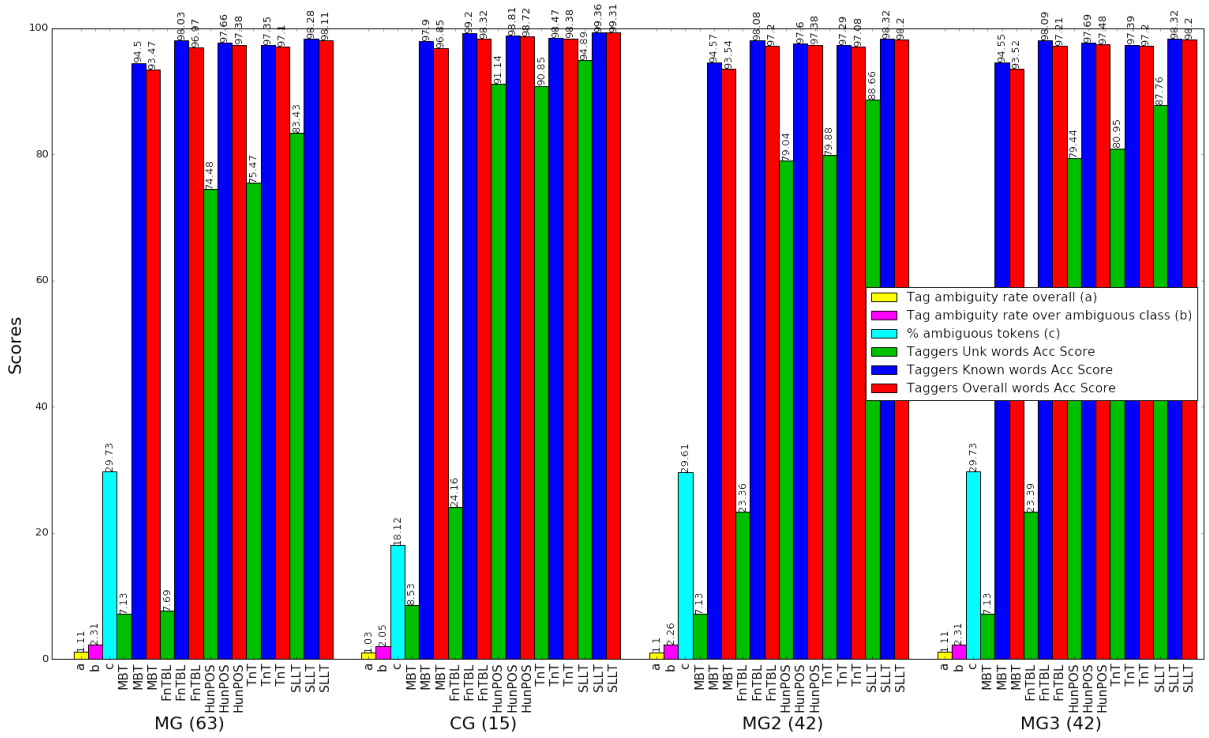



Figure 8.5: Different number of tags found in IgbTNT (263856) and the effects on taggers performance. SLLT in this figure is $SLLT^*$

**MG2** is where we removed _XS marker from the tags in IgbTNT and IgbTMT, the 63 tags of IgbTNT was reduced to 42 tags and 61 tags of IgbTMT to 45[19]. Then, taggers were trained and tested on IgbTNT and IgbTMT based on 90%:10% cross validation.

[19]Tags LTT (for lists using alphabets) and ABBR (for abbreviations like UN) are used in IgbTMT only and VPERF_BPRN_XS (for morph-inflected perfect tense and pronoun bound verbs) tag used only in IgbTNT.

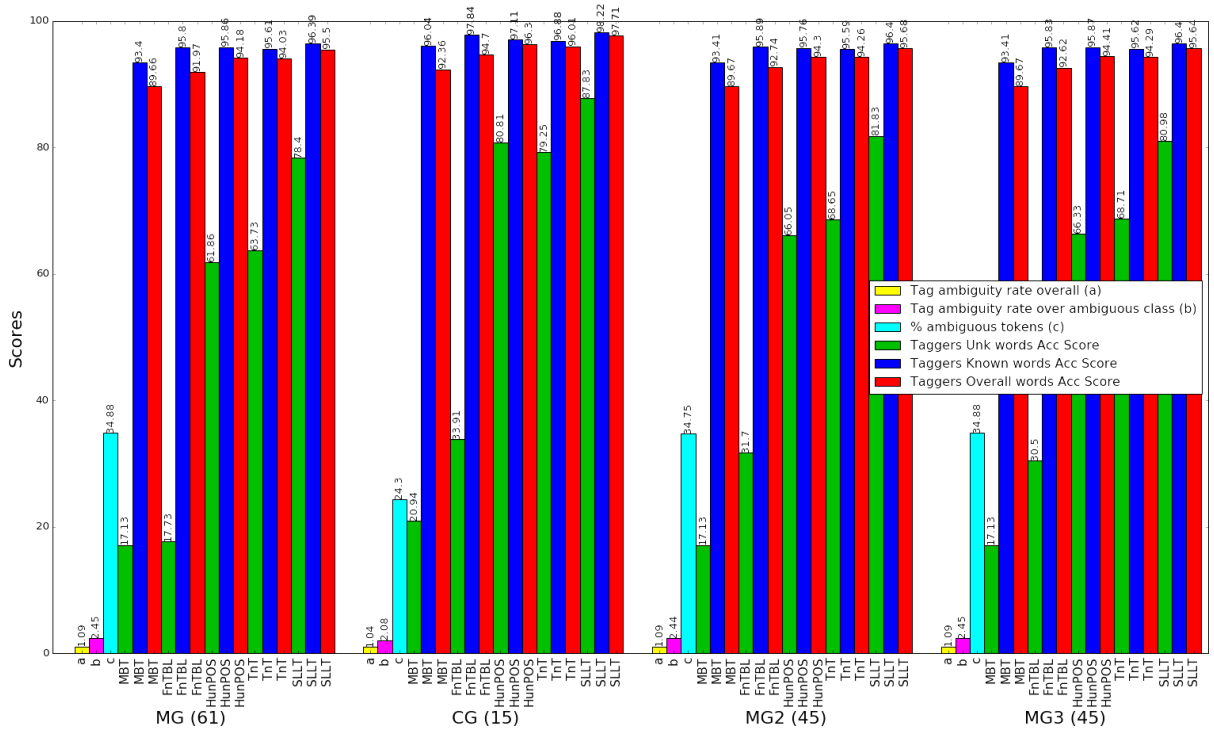



Figure 8.6: Different number of tags found in IgbTMT (39960) and the effects on taggers performance. SLLT in this figure is $SLLT^*$

Evaluation was carried out on the SLLT's test results, which contains only $t$ tags. From the figures (8.5 and 8.6), we observe the followings:

- for IgbTNT, the ratio of tags per word reduced by 0.05 over ambiguous class and 0.02 on the overall, word ambiguity percentage ratio reduced by 0.12%. The accuracy scores on the unknown words, known known and overall words generally increased. For example, the SLLT's accuracy scores increased by 5.23% (unknown words), 0.04% (known words) and 0.09% (overall) respectively.

- for IgbTMT, the ratio of tags per word reduced by 0.02 over ambiguous class and 0.004 on the overall, word ambiguity percentage ratio reduced by 0.13%. The accuracy scores on the unknown words, known words and overall words generally increased. For example, the SLLT's accuracy scores increased by 3.43% (unknown words), 0.01% (known words) and 0.18% (overall) respectively.

**MG3** means we trained and tested the taggers on the IgbTNT and IgbTMT, and then strip $XS$ from the $t$_XS tags in the tagger's output. This collapsed 63 tags in the taggers' output to 42 tags on which we carried out evaluation. Compare accuracy scores of taggers on MG3 and MG in both figures (8.5 and 8.6), you will notice slight differences with MG3 taking the lead in most cases. Although the general accuracy scores drop in MG compare to MG2 and MG3, but the accuracy scores increase when $XS$ markers are removed from MG tags (MG3), which is better than or equal to MG2. Therefore, it depends on the user's interest whether to use the $XS$ marker or not, since the both approaches (MG and

MG3) deliver good accuracy scores. MG gives information about the morphological parts of the language making the language corpus to be more informative.
**CG** (coarse-grained) in the figures is where 63 and 61 tags of IgbTNT and IgbTMT were mapped to 15-tag CG of Igbo tagset[20]. We observe the followings after training and testing taggers on IgbTNT and IgbTMT:

- for IgbTNT, the ratio of tags per word reduced by 0.25 over ambiguous class and 0.08 on the overall. Word ambiguity percentage ratio reduced by 11.61%, the accuracy scores on the unknown words, known words and overall words generally increased. For example, the SLLT's accuracy scores increased by 11.46% (unknown words), 1.08% (known words) and 1.20% (overall) respectively.

- for IgbTMT, the ratio of tags per word reduced by 0.37 over ambiguous class and 0.05 on the overall, word ambiguity percentage ratio reduced by 10.58%, the accuracy scores on the unknown words, known words and overall generally increased. For example, the SLLT's accuracy scores increased by 9.43% (unknown words), 1.83% (known words) and 2.21% (overall words) respectively.

The increase in the accuracy scores as a result of different sizes of tagset used in this experiment is not surprising. It has been discussed in the literature that the less the tagset size the more accurate is the tagging performance of the taggers (Atwell, 2008; De Pauwy et al., 2012). That means that there are few cases of ambiguous words, which implies that the percentage of unambiguous words will increase. However, if we are to trade-off developing more informative tagged corpus to accuracy, the trivial and uninformative tagged corpus containing a tag 'WORD' for identifying *whether a token is a word or not* would be optimal. It is important that most (if not all) the grammatical key player-words are assigned tags based on the grammatical role they play on a sentence. For example, if we decide to do away with the *XS*, the goal of capturing the morph-inflected tokens, which also are part of the Igbo grammar will be defeated. Capturing morph-inflected tokens is one of the key points towards performing full-scale computational morphology in Igbo.

## 8.10 Determinants of Tagging Accuracy

How hard is the tagging problem? In this section, we discuss the answers to this question by looking at the following factors that influence tagging accuracy. The following characteristic aspects of languages could affect the POS tagging system.

### 8.10.1 Text texture

There are different writing styles that affect the use of words within a sentence in texts. Style is normally used to address a specific context, purpose, or audience[21]. Hence, a writer would like to choose words and structures sentences to style up a text for his/her audience. That is, the coherence of texts is dependent on the use of words and sentences, and this is what can be regarded as text texture. Texture is the basis for unity and

[20] See section 6.1.1 of chapter 6.
[21] `http://www.learnnc.org/lp/editions/few/684` [accessed: 15-04-2016]

semantic interdependence within a text. Texts without texture are simply bunches of isolated sentences without relationship to each other (Crane, 2006). For example, given to five translators is an English text to be translated into Igbo in isolation, this will output five different texts determined by the translators' writing styles (*say* wordings). The Igbo Bible texts we used as one of the corpora is translated from English, we observe that there are two variations in the use of perfect tense suffix "-la" and "-wo". While "-wo" in the corpus unambiguously signify perfect tense "-la" could be signifying perfect tense and negation. This could be avoided by using the primary standard orthographic letters "-go". This particular instance affected taggers performance on the inflected verbs that are simple, participle, and perfect tense (See table 8.16 and figure 8.4).

Taggers' performance is affected by a language text textures, and the relationships that exist between words. This can be better understood by comparing how easy it is to read and comprehend different sentences in fictions and political speech texts. Giesbrecht and Evert (2009) identify in their work "easy" genres collectively classified to newspaper text where taggers performed better than problematic genres where all taggers performance falls below 94%. Looking at inner text texture through the verbal complex structure in Igbo, it comprises two parts represented by *verbs* and *inherent complements*, where *verbs* could be any verbal class (*VSI, VPP, VrV, etc*), and *inherent complements* are nouns represented by the tag *NNH*. Their occurrence positions are limited within a sentence, but can be immediate after each or *n* words apart. *Inherent complements* can occur on either side of the verbs they are complementing, but mainly *inherent complements* are preceded by their *verbs*. For example, observe the positions of *verbs* and their *complements* in the followings: *ọsọ/NNH a m gbara/VrV mịtara ezi mkpụrụ* "This run I ran bore good fruit", *Ọ na- agba/VPP ọsọ/NNH* "S/he is running", *Onye ọka/NNCV ikpe/NNCC ahụ kpere/VrV mmadụ niile mere ihe ọjọọ ikpe/NNH* "That Chief justice judged everybody that did bad thing" and *Ọ gbara/VrV nwunye ya aldugwaghịm/NNH*. These words can be ordinary common nouns "NNC" if they are standing alone in a sentence (like *ọsọ/NNC a bụ maka ndị na-enweghị ike* "This run is for people that don't have strength"). In "ọka/NNCV ikpe/NNCC", NNCV is for nominalization[22], and NNCC is the noun *inherent complement*.

To evaluate how easy it is for taggers to classify the words of a sentence, we scored tagging accuracy of words in IgbTC1 corpus on sentential levels. The best tagger (SLLT) scored 42.81% on sentence level accuracy over 3252 sentences. On tag level, only *NNH* out of 66 tags contributed 14.07% of the total error, and it is the topmost tag taggers mistaken to be NNC in table 8.16. While on genres, the SLLT tagger made total errors of 51.33% and 56.15% on IgbTMT and IgbTNT1 sentences. This indicates that Igbo Modern texts (IgbTMT) represented using a recently written novel is not easy genre compared to the religious texts (IgbTNT) represented using Bible[23] (IgbTMT sentence size is about 60% of IgbTNT1). SLLT got sentence accuracy of 56.34% in English Penn treebank III corpus (Toutanova et al., 2003), which is 13.53% higher than 42.81% scored in Igbo.

[22]Verbs that changed to nouns.
[23]Obtained from`jw.org`

### 8.10.2 Tagset Granularity

This is the number of tags in the tagging scheme. There are 70 tags in the Igbo language tagset, and only 66[24], 63 and 61 of them were used in the IgbTC, IgbTNT and IgbTMT corpora. This means that taggers will have more tags in tag sequence to choose from for each word in Igbo compared to 45-tag Penn TreeBank tagset. But there will be less tags to disambiguate in Igbo compared to 139 tags of Swedish tagset, 341 tags of Dutch tagset, 660 tags of Icelandic tagset and 1171 tags of Czech tagset. Morph-inflected languages tends to have more tags because of the grammatical functions these morphemes performs in the language. At the design stage of Igbo tagset (see chapter 6), we identified over 30 morphological classes which we collapsed to *XS* to denote any word with any kind of affixes. This reduced our tagset to a medium grained size of 70 tags. Comparing tagger performance on the languages discuss in the immediate above section, taggers POS tagging on a very large tagset data perform poorly, as in Czech language. TnT and FnTBL performs better in Igbo language than Icelandic, Dutch and Swedish texts.

### 8.10.3 Lexical Ambiguity

Lexical ambiguity reveals how many words a tagger will get right without disambiguation. It is influenced by the size of the tagset, that is, the greater the size of the tagset the high the ambiguity rate. High ambiguity rate results that taggers will struggle to disambiguate a considerable size of tokens. A corpus with less percentage of lexical ambiguity can be referred to as straightforward POS tagging task, which is a regular case for languages that are both conjunctively[25] and morphologically written like Zulu language (De Pauwy et al., 2012). The ambiguity ratios for word types and tokens are summarized for Igbo tagged corpus (IgbTC) in table 8.3. Following this, the percentage of number of word types in ambiguous class are 8.50%, 6.44% and 9.35% for IgbTNT, IgbTMT and IgbTC corpora. These percentages, although small amount of the vocabulary, account for the most frequent words in Igbo hence resulting to 29.71%, 34.87% and 36.65% ambiguous tokens for IgbTNT, IgbTMT and IgbTC corpora. Analysing the performances of taggers based on this, the best tagger, SLLT performed full disambiguation of these words ambiguity rates and correctly disambiguated 27.82% of 29.71%, 30.37% of 34.87% and 34.20% of 36.65%, and incorrectly disambiguated 1.89% of 29.71%, 4.50% of 34.87% and 2.45% of 36.65%. These later figures are the error rates in tables 8.11. For comparison with other taggers and languages see table 8.17. This table was calculated by comparing tables 8.3 and 8.11.

The very high (59.66%) ambiguous words ratio of Icelandic corpus (IFD) shows that most of IFD words (mainly frequent) have more than one meaning. This is a difficult tagging task compared to IgbTC. From table 8.17, MBT, FnTBL and TnT scored 89.28%, 89.33% and 90.44% respectively in IFD (Loftsson, 2007). This means that MBT, FnTBL and TnT disambiguated 48.69%, 48.99% and 50.10% of 59.66% with remainder of $< 11$ for all of them.

[24]All non inflected tags are used. The unused tags are ones with extensional suffix marker (XS) that happened to occur without suffixes.

[25]The practical orthography rendering of "I will work for them" in Zulu is written as one word, namely *ngizobasebenzela* instead of *ngi-zo-ba-sebenzela* in Nothern-Sotho (Louwrens and Poulos, 2006).

| Corpus | Ambiguous word ratio | Taggers | Correctly Disambiguated | Incorrectly Disambiguated |
|---|---|---|---|---|
| IgbTNT | 29.71% | MBT | 23.18% | 6.53% |
| | | FnTBL | 26.68% | 3.03% |
| | | HunPOS | 27.09% | 2.62% |
| | | TnT | 26.81% | 2.90% |
| | | SLLT | 27.82% | 1.89% |
| IgbTMT | 34.87% | MBT | 24.53% | 10.34% |
| | | FnTBL | 26.84% | 8.03% |
| | | HunPOS | 29.05% | 5.82% |
| | | TnT | 28.90% | 5.97% |
| | | SLLT | 30.37% | 4.50% |
| IgbTC | 36.65% | MBT | 29.37% | 7.28% |
| | | FnTBL | 32.93% | 3.72% |
| | | HunPOS | 33.32% | 3.33% |
| | | TnT | 33.02% | 3.63% |
| | | SLLT | 34.20% | 2.45% |
| Icelandic | 59.66% | MBT | 48.69% | 10.72% |
| | | FnTBL | 48.99% | 10.67% |
| | | TnT | 50.10% | 9.56% |
| English WSJ | 55.00% | MBT | 51.40% | 3.60% |
| | | FnTBL | 51.70% | 3.30% |
| | | HunPOS | 51.58% | 3.48% |
| | | TnT | 51.46% | 3.54% |
| | | SLLT | 52.24% | 2.76% |

Table 8.17: Taggers performance scores on disambiguating ambiguous words. SLLT in this table for IgbTNT, IgbTMT and IgbTC is $SLLT^*$

## 8.11 Conclusion

We have discussed in this chapter the POS tagging evaluative experiments for the IgbTC developed in this research. Instead of re-inventing the wheel, we evaluated the existing tagging techniques on IgbTC by conducting several experiments. We empirically observed that our aim to develop an automatic POS tagging system from IgbTC produced using the 70-tag Igbo tagset is indeed a successful effort. The 70-tag tagset of Igbo developed in this research captured the key linguistics features of IgbTC on sentential level. It is surprising that these independent taggers developed and tested mostly on European languages did well in Igbo considering the morphological nature of the language. The efforts made by individual taggers on the known words apart from tokens they freely tagged without disambiguation are considerably good. But their unknown words accuracy scores are by several points low compared to other languages they have been tested on. It is interesting to note how the parameters for handling unknown words for HunPOS, TnT and SLLT worked well for some of the Igbo words that led to their fair accuracy scores on the unknown words.

We started this experiment with whether we can develop a POS tagging system suitable for Igbo considering the fact that it is a new language in NLP. Since non of

the existing taggers' techniques have been tested on the language, we used the tagged corpora developed in this research for Igbo to serve as a test-bed for these taggers. From literature, Stanford Maximum Entropy log-linear tagger (SLLT), Trigram'n'tagger (TnT), Hungarian Part-of-speech tagger (HunPOS), Transformation-Based Leaning on the Fast Lane (FnTBL), and Memory-Based tagger (MBT) are statistical, memory, and rule based taggers that have performed well in POS tagging problems. We chose these taggers considering the strong feats they have already achieved in other languages in handling unknown words. Their unknown words accuracy scores recoded in Igbo are not encouraging. This is possibly because their techniques are based in one particular order (extracting last $n$ letters of a word), they were unable to capture all the cases of morphological characters that are to serve as important cues for handling unknown Igbo words. This led to their low accuracy scores compared to other languages. Though Stanford Log-Linear Tagger (SLLT) benefited from using variables up to the value of $n$ and extracting letters of a word from both beginning and ending parts. SLLT is the best tagger among all the taggers used with best accuracy scores of 83.95% and 98.11% for unknown and overall tokens. The overall POS tagging accuracy scores of some taggers are approximately close to human ceiling, from figure 6.3 in chapter 6, best pair human annotators' agreement score is 98.71% (Cohn's Kappa) or 98.83% (raw agreement).

We also performed tagging across genres to evaluate the difficulty of moving from one genre to another. Our results revealed that major problem associated with this NLP task is the unknown words. Unknown words ratio increased when we trained and tested on dissimilar texts than when we used similar texts. But when these dissimilar texts are combined into one the unknown words ratio decreased, and the overall accuracy increased. The elements of unknown words in Igbo are not mainly nouns as the case may be in English, but relatively there are sizeable number of morph-inflected words. Words are morphologically built in tune with the story line or writing styles in a text.

Our evaluation reveals that the linguistic patterns of IgbTC is highly consistent. Study has shown that one of the major draw-back for POS tagging classifiers' performance is noise in the data. Noise creates inconsistency in pattern detection thereby generating low facts for disambiguating ambiguous instances. The high accuracy scores of taggers used show that IgbTC contains less errors which indicate high level of consistency in IgbTC. These errors are the points that contributed taggers failures, and precisely where there is need to insert human judgements, which could be making corrections, adding clues or labelling of more data to improve the tagger's performance. The "not good enough" performances of taggers in previously unseen words in the training data led to the investigation into what could increase accuracy on the unknown words in the next chapter.

# Chapter 9

# Morphological Features for Prediction in Igbo

This chapter discusses how to process Igbo words by using the morphological characteristics of the language. There are three major sections of this chapter: first is using morphological reconstruction method to develop morphological segmentation module that will find the actual affixes in Igbo given any morphologically-inflected (morph-inflected) word. Next is the development of an automatic error correction method that will improve the correctness of tags assigned to words that are morphologically-inflected in Igbo tagged corpus (IgbTC). Final section is the improvement of tagging accuracy on the morph-inflected words that are new or previously unseen in the taggers training. This is a linguistically-motivated approach that will enable taggers to make use of the language morphological information in order to improve their performance. We develop a tagger based on this approach, and compare performance to other taggers on the morpho-inflected words that are not seen in the taagers lexicon.

Morphological reconstruction is a linguistically-informed segmentation into root and affixes. Knowledge of the root and the associated affixes are used to process unknown words. Thede and Harper (1997) investigates whether a parser can parse unknown words using morphology and syntactic parsing rules. They use morphological recognition that uses knowledge about affixes to predict the possible parts-of-speech (POS) of words in the TIMIT corpus without using any direct knowledge concerning the word's stem, which greatly improved their parser. Milne (1986) uses morphological reconstruction to resolve ambiguity while parsing, and Light (1996) exploits morphological cues that find meaning of words by using various information sources. Sawalha and Atwell (2009) develop a morphological analyzer that uses linguistic knowledge of Arabic language as well as corpora to verify the linguistic information.

The idea is to show the uses of morphology for analysis of words in order to improve taggers' performance. This is important especially when there is a limited corpus, and it is expected that taggers cope with the language new words that are not in the lexicon. By using knowledge of root and associated affixes, appropriate tags for morph-inflected words that are complex and unseen in the training data can be predicted. Machine learning tools used have been discussed extensively in the previous chapters. Transformation-based learning on the fast lane (FnTBL) (Ngai and Florian, 2001) is a reimplementation of Brill (1995a)'s TBL. It is a machine learning algorithm that starts with an initial state and

correctly tagged text (*truth*). The training process iteratively acquires an ordered list of rules that correct errors found in the initial state, until this resembles the truth to an acceptable degree. The output of morphological segmentation will benefit FnTBL's linguistic pattern detection.

## 9.1 Morphological Parser

This section discuses the used morphological reconstruction method to present words in morph-inflected class in morphological learnable patterns (root and associated affixes). We design a module for segmenting morphemes and stems of morph-inflected words found in IgbTC such that their stems and affixes are classified as stem (ROOT), prefix (PRE)[1] and suffix (SUF) tags irrespective of their grammatical functions. This will generate a tag set of {PRE ROOT $SUF_{i...n}$} for any given morph-inflected word. For example, this word *enwechaghị* tagged "VPP_XS" in the IgbTC will have the form "e/PRE nwe/ROOT cha/SUF ghị/SUF" after morphological reconstruction. The plan here is to use these morphological clues to predict the correct tags for the morph-inflected words.

The approach is, for any given word $w$, the stem $cv$ is extracted and all $n$ possible morphological parts attached to $cv$ are generated. Stem in Igbo is a formation of $cv$ that starts with a consonant $c$ and ends with a vowel $v$ (Emenanjo, 1978), where $c$ could be a single letter or double in the case of digraph. Digraphs are two character strings pronounced as one sound, and are non split (examples "gh", "ch", "kw", "gb", "gw", "nw", "ny", "sh", "kp"). We used a list of suffixes from (Emenanjo, 1978) as a dictionary to search for valid morphological forms. To test how robust this system is, we avoided using any tag information from IgbTC for tracking of morph-inflected words. Therefore, for any given word, if there is $n$ valid morphological part(s) attached to its $cv$, then the word will be detected and reconstructed (e.g. enwechaghị: "e/PRE nwe/ROOT cha/SUF ghị/SUF"). Otherwise, that word is not morph-inflected.

This is not a full scale computational morphology in Igbo, we only focused on morph-inflected words that are verbs since they constitute the majority of words in the morph-inflected class. We avoided full scale morphological analysis at this stage because of time constraints. The system uses a dictionary of Igbo suffixes in its module to perform morphological parsing process on words that are morphologically-inflected. In the case of verbs' nominalization to nouns, we used nominalizing prefixes (n,m,o,u,ọ,ụ) to track these instances, and avoid entering them for reconstruction. Another important clue is the use of word-shape, verb shapes normally starts with *VCV, CV, CVV, VCVCV, CVCVCV* ("C" is consonant and "V" is vowel), etc., but cannot end with a *C*. For example, verbs "atụkwasịrị" and "banyekwa" have common word-shapes of "VCVCVCVCV" and "CVCVCV" for verbs but words "mpịakọta", "Kapanịọm" and "mgbaasị" have word-shapes "CCVVCVCV", "CCVVCV" and "CVCVCVVC" different from the verbs.

How accurate is this system in tracking morph-inflected words that are verbs? Igbo tagset is designed to have special tags given to morph-inflected words. We used this information to build lexicon of all morph-inflected words that are verbs, and compared it with the output of the morphological parser. For example, there are 31,383 morph-inflected verbs in the IgbTC, the parser extracted 35208 words from IgbTC corpus, and out of this

[1]Prefix in Igbo is only a single character long.

number, 29817 (95.01%) are morph-inflected verbs and 5391 are not. The remaining 4.99% of morph-inflected verbs are mostly where there is a single character inflection called open vowel suffix. That means, they are inflection caused by vowels, examples are *lee le+e, rụọ rụ+ọ, mịa mị+a, etc.* These word-forms require a more robust computational morphology to segment properly. For example, there are non morph-inflected verbs that have ***X**ịa* form related to *mịa* (example is *bịa*). The 5,391 words have the same word-shape with verbs, example is "ọchịchịrị" which is a noun with the same shape "VCVCVCV" as verb "ekwusakwa" (VCVCVCV). Furthermore, most of the words we found in this 5,391 words are mainly common nouns, therefore we used list of noun class constructed from the corpus to eliminate them.

## 9.2 Current State of Igbo Tagged Corpus

The entire improvement processes reported in chapter 7 resulted in inspecting 26.20% of IgbTC with 14.601% effective change made and accuracy increased from ≈88% (initial state of IgbTC) to ≈96% (current state of IgbTC) obtained by training and testing FnTBL tagger on IgbTC sets on 10-fold cross validation over the corpus size.

## 9.3 Improving the Correctness of Morph-Inflected Words Tags

The quality of part-of-speech (POS) annotated corpus is crucial in the development of automatic POS taggers. In POS tagging system, taggers use context in order to disambiguate focus words correctly. For example, in transition probability, Hidden Markov Model (HMM) based taggers use previous tags to decide correct tag for the current ambiguous words, that is, words with more than one tags (Jurafsky and Martin, 2014). This implies that if an irrelevant tag is wrongly assigned at some point in the corpus, a tagger will learn the wrong morphosyntactic information and use it in disambiguating ambiguous cases wrongly at similar points, and this will degenerate the tagger's performance over time.

The major source of errors in a tagged corpus is the way in which they were developed. There are different options to consider towards developing POS annotated corpora. It could be manually annotate the entire or a significant amount of the corpus, mixed method- to manually annotate a part, and the remainder semi-automatically, or opt for purely automatic method. Automatic annotation is less error-free but can produce many more POS tagged corpora than humans can reasonably achieve. Manual is more error-free, but very labour-intensive and costly. In semi-automatic annotation, manual steps can come in several stages of the overall process and the output is hand checked. The outcome of this process is often used to train taggers to perform automatic POS tagging and to test their performance. Therefore, any deviation from the regularities which the taggers are expected to learn as a result of errors in the assignment of tags in a corpus means the taggers' possibility to get confused about probability distribution of tags assigned in the corpus (Pavel and Karel, 2002). Despite careful human efforts in pre- and post-editing phases of POS annotation, tagged corpora still contains errors certainly caused by human

mistakes. It is therefore necessary to develop efficient methods that will automatically detect errors in a tagged corpora, and possibly suggest plausible tags for correction which can be investigated by humans. This will greatly reduce the extensive labour of a human annotator expert going through the entire tagged texts methodically to find and correct errors.

| Taggers | Overall Scores | Unknown Scores | Inflected Unknown Words Scores |
|---|---|---|---|
| SLLT | 98.05% | 77.77% | 58.01% |
| HunPOS | 97.33% | 65.84% | 48.68% |

Table 9.1: Results using SLLT and HunPOS on current state IgbTNT

After first round POS tagging experiment using the current state of IgbTC, we were concerned about tagger's performance on the new or previously unseen words (unknown words). The quality level of tags assigned to the morph-inflected words in the corpus may be one the causes of the poor accuracy scores of taggers since the majority of unknown words are morph-inflected (see results in table 9.1). Therefore, we developed an automatic method that find errors where the assignment of tags violates the status of words that are morph-inflected in IgbTC. Igbo is morphologically-rich language in which new words are coined into the language vocabulary stream mainly through the use of morphology. A single stem in Igbo can produce as many possible word-forms using affixes of varying lengths from 1 to 5, which only extends the original meaning of the words (see table 9.2). We used this automatic process to exploit morphological information in Igbo as a means to correct those words that are morph-inflected which are incorrectly tagged in the IgbTC.

| Word-form | Stem and Affixes | Meaning |
|---|---|---|
| ri | ri | eat |
| iri | i+ri | to eat |
| ga-eri | ga-+e+ri | will eat (auxiliary verb hyphnated to participle) |
| ga-ericha | ga-+e+ri+cha | will eat completely |
| ga-erichaịrị | ga-+e+ri+kwa | will eat also |
| richarịrị | ri+cha+rịrị | must eat completely |
| richakwa | ri+cha+kwa | eat completely also |
| richara | ri+cha+ra | ate completely |
| richakwara | ri+cha+kwa+ra | ate completely also |

Table 9.2: Illustrating word formation in Igbo using morphology

### 9.3.1 Related Work

There have been works done in correcting errors found automatically in a tagged corpus. Instead of going through tagged corpora[2] word by word or sentence by sentence by human annotator expert to find and correct errors, an efficient means can be developed that

[2]Perhaps tagged in a fashion to avoid extensive manual tagging all through or because there is a wish to improve existing tagged corpus.

uses the human expert in its process loop to correct errors found or make suggestions, to improve method's efficiency. Brill and Marcus (1992) use a semi-automatic way for tagging an unfamiliar text and then applied learned rules to both correct errors and find where contextual information can repair tagging mistakes with little help from a native speaker. Taljard et al. (2008) and Heid et al. (2006) use lexicon that contains 7000 known words and their annotations, a noun and verb guesser to pre-tag 40000 tokens of Northern Sotho's texts. The output was reviewed manually and correct guesses are added to the lexicon. Thus the size of the lexicon grows continuously. Finding and correcting errors to make more accurate annotated data as experimented in Loftsson (2009) and Helgadóttir et al. (2012) is method of correcting errors found automatically in a tagged corpus. Loftsson (2009) and Helgadóttir et al. (2012) apply trained POS taggers singly and combined, respectively, then the outputs were compared with the gold standard and differences found were marked as error candidates for verification. In this experiment, we apply an automatic method that learns rules from the morphologically reconstructed words in Igbo tagged corpus (IgbTC) and then apply these rules to find and propose tags for all morph-inflected words not tagged properly. All positions where these changes occurred are inspected and corrected by human annotator expert for quality assurance.

### 9.3.2 The Experiment

Igbo tagset is defined in two parts: $\alpha$ and $\alpha\_XS$, where $\alpha$ represents any non morph-inflected tag and $XS$ is to indicate presence of any affix in a word that is morph-inflected (See chapter 6 and appendix A for tagset design and development). This experiment automatically find and correct those morph-inflected words that suppose to be tagged as $\alpha\_XS$ but are not in IgbTC. For the automatic error correction method experiment, we used the following tools: IgbTNT, morphological segmentation discussed above and FnTBL. In order to test the impact of this error correction method on the corpus, we evaluate accuracy on the words that are morph-inflected using the following taggers: Stanford Log-linear Tagger (SLLT) (Toutanova et al., 2003) and Hungarian part-of-speech (HunPOS) tagger (Halácsy et al., 2007) (a reimplementation of Brants (2000b)'s TnT). The output of morphological parser will benefit the FnTBL's linguistic pattern detection. SLLT and HunPOS have robust word features extraction techniques for prediction. For example, SLLT uses variables up to $n$ in extracting first/last letters of a word such that $n = 4$ for *negotiable* will generate the extraction list [e,le,ble,able] to serve as proxy for linguistic affixes.

FnTBL was trained and tested on the outputs of the morphological parser (see outputs in table 9.3). FnTBL's lexical lookup module uses unigram tagging to generate its initial state, and then a rule application module proceeds iteratively to correct some of the initial tags on the basis of the truth state. We used morphological parser method to override the FnTBL's module for generating initial state. We did this by assigning "ROOT" to all the verb stems while the associated affixes are given SUF (suffixes) and PRE (prefixes), and then ROOT will be replaced with the verb's tag from IgbTC in the FnTBL's truth state. For example, in table 9.3, the verb *nwukwasị* tagged "VSI_XS" in the IgbTNT will have the forms "nwu/ROOT kwasị/SUF" and "nwu/VSI_XS kwasị/SUF" for FnTBL's initial and truth states respectively.

Table 9.4 shows the inflected words and tags from tagged Igbo corpora (IgbTMT

| | Morphologically Reconstructed | |
|---|---|---|
| Word form | FnTBL Initial State | FnTBL Truth State |
| nwukwasị | nwu/ROOT kwasị/SUF | nwu/VSI_XS kwasị/SUF |
| nwukwara | nwu/ROOT kwa/SUF ra/SUF | nwu/VrV_XS kwa/SUF ra/SUF |
| nwukwasịrị | nwu/ROOT kwasị/SUF rị/SUF | nwu/VrV_XS kwasị/SUF rị/SUF |
| ịnọdonwu | ị/PRE nọ/ROOT do/SUF nwu/SUF | ị/PRE nọ/VIF_XS do/SUF nwu/SUF |
| abịakwara | a/PRE bịa/ROOT kwa/SUF ra/SUF | a/PRE bịa/VPP_XS kwa/SUF ra/SUF |
| ịzụpụtara | ị/PRE zụ/ROOT<br>pụ/SUF ta/SUF ra/SUF | ị/PRE zụ/VIF_XS<br>pụ/SUF ta/SUF ra/SUF |
| hapụrụ | ha/ROOT pụ/SUF rụ/SUF | ha/VrV_XS pụ/SUF rụ/SUF |

Table 9.3: Some samples of morphological-complex words morphologically reconstructed into stems and affixes to serve as FnTBL states. FnTBL will be trained on these states

| Inflected Word | Tag | TBL Test Data | TBL Transformation Rule | TBL Transformed Tag |
|---|---|---|---|---|
| pụọrọ<br>(from IgbTMT) | VrV | pụ ROOT<br>ọ SUF<br>rọ SUF | $r0_t$: ROOT => VrV<br>$r2_t$: VrV => VrV_XS | VrV_XS |
| ịhapụrụ<br>(from IgbTNT) | VrV_XS | ị PRE<br>ha ROOT<br>pụ SUF<br>rụ SUF | $r0_z$: ROOT => VSI_XS<br>$r1_z$: VSI_XS => VPP_XS<br>$r3_z$: VPP_XS => VIF_XS | VrV_XS |

Table 9.4: Some output examples of FnTBL's predicted tags using morphological information

and IgbTNT), TBL test data that contains reconstructed morph-inflected words, TBL transformation rule which is an ordered rule list FnTBL generated during training session using data in table 9.3, and TBL transformed tag which is the final tag FnTBL predicted.

The "transformed tag" column in table 9.4 is the FnTBL's predicted tags using its transformational rules ("TBL transformation rule" column) generated from table 9.3 data. For example, the inflected word "pụọrọ" was tagged "VrV" (Past tense verb[3]) which indicates only inflectional part (rV[4]) of the words. But FnTBL rules transformed this tag "VrV" to VrV_XS indicating the presence of suffix (XS). The transformational rules are contextual driven and here are the meanings:

- Rules $r0_t$ and $r0_z$ used the same context but gives different tags (VrV and VSI_XS). The context is "pos_0=ROOT word:[-2,-1]=ZZZ", which implies, if tag is ROOT and there is a boundary marker (ZZZ)[5] found within the previous two positions, change ROOT to VrV or VSI_XS. This is prefix[6] optional, which implies if the immediate previous position is prefix, then the next previous position is a boundary marker,

[3]A verb becomes past tense through inflection (addition of rV) and it could become more complex by addition of suffixes.

[4]rV means letter "r" and any vowel (a,e,i,ị,o,ọ,u,ụ) which is a past tense marker in Igbo (Ikegwuonu, 2011).

[5]This is at the end of every data instance (usually a sentence, but in this case, a word segmented into its morphemes), so that transformation-based rules can refer to this as a context element, so as to "anchor" their use to either beginning or end of the sequence.

[6]There is only a single length prefix found in some words in Igbo.

otherwise the immediate previous line is a boundary marker. The reason for $r0_t$ and $r0_z$ using the same context but gave different tags is dependent on the two corpora used. IgbTMT contains more words having only past tense inflectional (INFL[7]) part (rV), that is words of the form *root-rV*, in the training data than IgbTNT. There are *root-rV, pre-root-rV, pre-root-suf(s)-rV, root-suf(s)-rV* forms, and if we exclude *root-rV*, the output tag of FnTBL using the same context changes to VSI_XS for IgbTMT, that is, "pos_0=ROOT word:[-2,-1]=ZZZ => VSI_XS", which is the same as IgbTNT.

- $r2_t$ context is "pos_0=VrV pos:[1,3]=SUF => pos=VrV_XS", that is, change VrV to VrV_XS if tag is VrV and there is a SUF (suffix) tag found within the range of 1 to 3 after stem.

- $r1_z$ context is "pos_0=VSI_XS pos_-1=PRE pos_1=SUF => pos=VPP_XS", that is, change VSI_XS to VPP_XS if tag is VSI_XS and previous tag is PRE and following tag is SUF.

- $r3_z$ context is "pos_0=VPP_XS word:[-2,-1]=ị => pos=VIF_XS", that is, change VPP_XS to VIF_XS if POS tag is VPP_XS and there is a prefix "ị" within the two previous positions.

| IgbTC Before Error Correction | IgbTC After Error Correction |
|---|---|
| nwukwasịkwara/VrV | nwukwasịkwara/VrV_XS |
| pụkwaghị/VrV_XS | pụkwaghị/VSI_XS |
| bụrụkwa/VrV_XS | bụrụkwa/VSI_XS |
| laara/VrV | laara/VrV_XS |
| waara/VrV | waara/VrV_XS |
| zooro/VrV | zooro/VrV_XS |
| zukwaara/VrV | zukwaara/VrV_XS |
| kwughachikwa/VCO | kwughachikwa/VSI_XS |
| kwuluwo/VSI_XS | kwuluwo/VPERF |
| ịhapụrụ/VrV_XS | ịhapụrụ/VIF_XS |
| kwọọ/VSI | kwọọ/VSI_XS |
| gbawasịa/VrV | gbawasịa/VSI_XS |
| tọgbọbu/VrV | tọgbọbu/VSI_XS |
| funahụ/NNC | funahụ/VSI_XS |
| tachie/NNCV | tachie/VSI_XS |
| pụọrọ/ VrV | pụọrọ/VrV_XS |

Table 9.5: Sample of morph-inflected words corrected

Any location where FnTBL suggested a tag different from what is in the corpus was flagged as candidates for inspection and correction. Firstly, we automatically verified if FnTBL predicted tag and tag in the corpus have the same base tag[8], and if there is any instance in the corpus where the tag of this instance and FnTBL's predicted tag have the

[7]Inflection in Igbo comprises two parts: past tense (rV) and perfect tense (PERF). See tagset in appendix for description.

[8]The following tags *VrV, VrV_XS, VrV_BPRN, VrV_BPRN_XS* have a common base tag of VrV.

same base tag and stem, we chose FnTBL's tag. Further explanations using examples in tables 9.4 and 9.5, *pụọrọ* was tagged "VrV" in IgbTMT corpora but FnTBL suggested "VrV_XS". In this case, VrV is the base tag in both FnTBL's predicted tag and the corpus tag, and there is existence of a suffix (SUF). Also if there exists in the corpus an inflected word tagged VrV_XS, which has the same stem "pụ" with *pụọrọ* and base tag "VrV" is the same with FnTBL's predicted tag, "VrV_XS" tag will be chosen. Another interesting example is "ịhapụrụ" where FnTBL suggested the right tag "VIF_XS" (morph-inflected infinitive verb) using the prefix "ị" information even though the last two letters usually indicates VrV_XS or VrV tag. Every other remaining cases (like kwughachikwa/VCO and kwughachikwa/VSI_XS, bụrụkwa/VrV_XS and bụrụkwa/VSI_XS in table 9.5 where there are different $\alpha$) were manually corrected. With this data improvement method, we corrected a total of 380 samples (all morph-inflected) in IgbTNT. For quality assurance, all these positions were inspected by a human annotator expert.

For training and testing SLLT and HunPOS on IgbTNT, IgbTNT was divided into train and test data on a 10-fold cross validation over the corpus size. The unknown word ratio is the percentage of words previously unseen in the train data.

Table 9.6 shows the results when we applied SLLT and HunPOS on the IgbTNT. After the application of this error correction process, SLLT and HunPOS accuracy scores on IgbTNT generally increased. The effect is very prominent in the accuracy of the unknown words (especially the inflected words). Compare tables 9.6 and 9.1.

| Taggers | Overall Scores | Unknown Scores | Inflected Unknown Words Scores |
|---|---|---|---|
| SLLT | 98.11% | 83.43% | 86.81% |
| HunPOS | 97.38% | 74.48% | 78.16% |

Table 9.6: Results using SLLT and HunPOS after this error correction method on IgbTNT

From table 9.6, the accuracy scores, after this error correction method, show that SLLT gained extra 0.06% for overall, 5.66% for unknown words and 28.8% for morph-inflected words that are unknown. The impact of this experiment on the morph-inflected words that are unknown shows that the majority of the corrected tags belong to the unknown words class which are mostly morph-inflected words that are less frequent. Notice from figure 8.2 of chapter 8 that the more addition of suffixes to a word, the less frequent and then rare/unknown it tends to become (also see table 9.9[9]). The accuracy scores are not about experiment in handling unknown words, rather we are showing the level of effects of this error correction technique on the sides of unknown words (both those that morph-inflected) and overall words.

In this experiment, we have shown how we used stems and associated affixes to transform morph-inflected words that were tagged wrongly to their correct tags in IgbTNT. Through morphological reconstruction, an actual linguistically-informed segmentation into roots and affixes, morph-inflected words in IgbTC are represented in machine learnable pattern that FnTBL exploited to identify and suggest plausible tags for those tags assigned to the morph-inflected words that violated their true status. Human annotator expert inspected all the affected positions on IgbTNT for quality assurance. This experiment

[9]For % proportion of morph-inflected words in unknown words ratios of table 9.8.

improved the quality of morph-inflected class of IgbTC (both IgbTNT and IgbTMT) to give what is now the current version of IgbTC and that is what we used in the next section experiments.

## 9.4 Morphologically-Complex Unknown Words

The effective handling of previously unseen words (unknown words) during the training session of part-of-speech (POS) taggers is an important factor in its performance. Some trainable POS taggers use suffix (and sometimes prefix) strings as a cue in handling unknown words (in effect serving as a proxy for actual linguistic affixes). In the context of creating a tagger for Igbo, we compare the performance of some existing taggers, implementing such an approach, to a novel method for handling morphologically-complex[10] (morph-complex) unknown words, based on morphological reconstruction, an actual linguistically-informed segmentation into root and associated affixes. Handling unknown words is an important task in NLP because unknown words class will continue to grow as new words are coined, and words associated with ethnic groups leak into the main-stream vocabulary. Unknown words in agglutinative language like Igbo is majorly caused by inflection with morphemes, for example, "nwukwasị" is known word in the training data that becomes unknown in the testing data due to "kwara" in "nwukwasịkwara".

Also in this section, we looked at the prospect of incorporating some meaningful morphology tags into the morph-inflected words, and perform tagging on morpho-tags. For example, morpho-tag in English can be illustrated as follows: (1) An airplane flies high and (2) The airplanes fly high, with morphology enriched tagset one can tag those as: (1) an/Det airplane/N-3sg fly/V-3sg high/Adv ./P (2) the/Det airplane/N-3pl fly/V-3pl high/Adv ./P. (Aibek et al., 2014; Elworthy, 1995). This is particularly important in agglutinative languages. Furthermore, we investigate if prefix is an important prediction cue in Igbo considering the fact it is only a single character length.

### 9.4.1 Related Literature

There have been works already done on POS tagging of unknown words and a number of features proposed for handling unknown words are based on $n$ neighbours of words/tags (where $n$ could be 1,2 or 3), prefixes, suffixes/word-endings and spelling cues like capitalization (Ratnaparkhi et al., 1996; Toutanova et al., 2003; Brants, 2000b; Halácsy et al., 2007). Brill (1995a)'s transformation-based error-driven learning (TBL) uses morphology to handle unknown words during POS tagging. It begins first by tagging unknown words as proper nouns if capitalized or common nouns otherwise. Then it learns various transformational rules from the corpus during training and applies these transformations to re-tag unknown words. Kupiec (1992)'s hidden Markov model assigns probabilities and state transformations to a set of suffixes of unknown words. Samuelsson (1993) uses starting and ending $n$ length of letter sequences of each word as predictive features of unknown words, and Brant shows that word endings like $-able$ is likely to be adjective in

[10]A word is morphologically-complex when it contains 3 or more affixes and becomes less frequent. For example, "bụ" occurred 3794 times as a root and 2579 times as a word in IgbTNT (New Testament Bible corpus). Some variations of "bụ" and their frequencies as a result of affixation are: bụrụ-1008, bụkwa-124, bụrụkwa-108, abụkwa-27, abụrụkwa-2, bụrụkwanụ-2, etc. Also see figure 8.2 in chapter 8.

English. Toutanova et al. (2003) uses variables up to the length of $n$ for extracting word features such that $n = 4$ for *negotiable* will generate [e,le,ble,able] feature list. These methods have worked well in languages like English and German whose derivational and inflectional affixes reveal much about the grammatical classes of words in question.

### 9.4.2 Problem Description

Igbo has many frequent suffixes and prefixes (Emenanjo, 1978). A single stem in Igbo can produce many word-forms and each suffix extends the original meaning of the former and can be interlocked with verb stem in variable order like the followings *abịakwa "a-bịa-kwa", bịakwaghị "bịa-kwa-ghị", bịaghịkwa "bịa-ghị-kwa", bịaghachiri "bịa-gha-chi-ri", bịachighara "bịa-chi-gha-ra", bịaghachirịrị "bịa-gha-chi-rịrị"*, etc. These suffixes have different grammatical classes and they contribute to the meaning of any word they are attached to (Emenanjo, 1978), which is extended to the sentence as a whole.

Suffix extraction method that uses a fixed or variable order but with upper limit to extract letters of a word may have problems to pick up on all morphological cues for predicting of unknown morphologically-complex (morph-complex) words in Igbo. Most existing taggers' word feature extraction methods are based on extracting the last $n$ letters of a word such that $n$ =4 for word "negotiable" will take *-able* for some taggers or [e,le,ble,able] for taggers using variables from 1 up to $n$ letters. For example, an Igbo word *bịaghachirịrị* "must come back" has three suffixes of lengths 3, 4 up to length of 10, use of the existing methods on this morph-complex word will miss the chances of extracting more linguistically-informed cues for prediction in Igbo.

Table 9.7 illustrates sample contents of Stanford log-linear tagger (SLLT) feature extraction list, where "-a" and "-ra" are only linguistically-informed morphological cues extracted.

| Extracted | Meaning | Example |
|---|---|---|
| ExtractorWordPref(len1,w0) | first letter of focus word | n_wukwasịkwa |
| ExtractorWordSuff(len1,w0) | last letter of focus word | nwukwasịkwar_a |
| ExtractorWordSuff(len2,w0) | last 2 letters of focus word | nwukwasịkwa_ra |
| ExtractorWordSuff(len3,w0) | last 3 letters of focus word | nwukwasịkw_ara |
| ExtractorWordSuff(len4,w0) | last 4 letters of focus word | nwukwasịk_wara |
| ExtractorWordSuff(len5,w0) | last 5 letters of focus word | nwukwasị_kwara |

Table 9.7: Rare/unknown words extractor lists of SLLT

It is a non trivial task to tokenize and manually tag enough training data that will account for all possible morphs in morph-inflected words of the language considering time factor. This led us to introduce *extensional suffix* tag *XS* in the tagset (see chapter 6) to mark all words that are morph-inflected, which is aimed towards full-scale computational morphology in Igbo.

### 9.4.3 Previous Tagging

Why is unknown words accuracy scores of taggers low in Igbo considering their performance in other languages. We investigated this and come up with the followings:

- The unknown words class in Igbo is mostly the case of nouns and morph-complex words, with latter dominantly high unlike in English where proper nouns form majority of unknown words. In table 9.9, it is shown that the average number of morph-complex words forms largely part of the unknown words in IgbTC (compare tables 9.8 and 9.9). From section 9.4.2, we have explained why the existing word features extraction method lacks the capacity to handle morphological-complex words[11] in Igbo. Taggers require more linguistically-informed affixes as word features to handle them properly.

- The noun class in Igbo tagset, designed in previous chapters, contains 8 different nouns. Therefore, FnTBL and other taggers that use capitalization for guessing most likely tag for unknown noun words, it is highly probable that most at times the chosen tag will not be the right tag for the given words (this can be seen in most frequent errors made by taggers in chapter 8). This is possible in English where there is only singular and plural nouns.

Since FnTBL that is rule-based tagger achieved a relatively high accuracy on known words and it has a powerful inductive method that learns the patterns of a language like grammar rules without human help. Presumably training it on a carefully constructed morphological characteristics of morph-inflected words should improve the performance of tagging process on the morph-complex word that are unknown.

We only handled the first case of the two above cases found in the previous tagging because the morphological segmentation module only handles verbal morphology which constitute large part of morphological inflection and unknown words classes in Igbo.

### 9.4.4 Experiment

The experimental aim is to find taggers performance on the new or previously unseen morph-complex words (unknown words). Unknown words arise from the previously unseen words in the training data constructed using 10-fold cross validation over the corpus size. There are two phases in this experiment: one used original forms of morph-inflected words and the other used morphologically reconstructed forms of morph-inflected words into roots and affixes (the actual linguistically-informed prefixes and suffixes). The latter experiment has four variations of patterns in data presentation.

#### Experimental Data

The corpus data used in this experiment are IgbTNT1[12] that represents religious genre and IgbTMT[13] for modern Igbo texts genre (IgbTMT). IgbTNT1 is about 15% of IgbTNT comparable to the size of IgbTMT.

[11]Some interesting illustration of morph-complex words formation from a single verb stem: *bịa*, *bịago*, *bịara*, *bịagoro*, *bịakwara*, *bịagokwara*, *bịachikwara*, *bịaghachikwara*, *bịaghachigoro*,*bịaghachigokwara* and so on. The first to three examples are more frequently used, next two are frequently used while remainders are less frequently used.

[12]Obtained from `jw.org`.

[13]Obtained from the author and written in 2013.

### Experimental Tools: POS Taggers and Classifiers

We chose tagging tools that generally did well on POS tagging and with parameters for word feature extractions for handling unknown words. We chose the following taggers: SLLT, TnT, HunPOS, and FnTBL. See section 8.1 of chapter 8 for taggers description. We also use Naive Bayes classifier (NBC) (Murphy, 2006) and Linear Support Vector Machine (LSVM) (Andrew, 2000) for choosing best tag between two different tags predicted by taggers.

### Experimental Setup

IgbTNT1 and IgbTMT were set into train and test data on a 10-fold cross validation over their sizes. Table 9.8 shows the average statistics of words in IgbTNT1 and IgbTMT used in experiment 1, and table 9.9 shows the average statistics of morph-inflected words in IgbTNT1 and IgbTMT used in experiment 2. The test column in table 9.9 is the average number of morph-complex unknown words in IgbTNT1 and IgbTMT. IgbTC1 is a combination of IgbTNT1 and IgbTMT in a stratified method.

The unknown word ratio is the percentage of words previously unseen in the train data. Comparing tables 9.8 and 9.9, if 3.18%, 4.90% and 3.39% are unknown word ratios in IgbTNT1, IgbTMT and IgbTC1 corpora, that means there are 69.29%, 68.37% and 71.22% of unknown words that are morph-complex in IgbTNT1, IgbTMT and IgbTC1.

| Corpus | Train | Test | Unknown Ratio |
|---|---|---|---|
| IgbTNT1 | 35938 | 3993 | 3.18% |
| IgbTMT | 35965 | 3996 | 4.90% |
| IgbTC1 | 71902 | 7989 | 3.39% |

Table 9.8: Average sizes of train, test, and unknown words ratio for the first experiment

| Corpus | Train | Test | % proportion of unknown words that are Morph-Complex |
|---|---|---|---|
| IgbTNT1 | 4120 | 088 | 69.29% |
| IgbTMT | 4855 | 134 | 68.37% |
| IgbTC1 | 8975 | 193 | 71.22% |

Table 9.9: Average sizes of train, test, and percentage morph-complex words occupied in unknown words. Train data contains morph-inflected words and test data contains morph-complex words that are unknown words

### Experiment 1: Using Original Word-Forms

HunPOS, TnT and SLLT taggers were applied on the data described in table 9.8. Word feature extraction length was set to $n$=5 because the longest suffixes in Igbo so far are 5 in length, and these taggers had performed well at this length[14] (see tables 8.10 and 8.11 in chapter 8). Tagging was done on the entire tokens, which allowed them to use

[14]Default settings of TnT and HunPOS tagger uses $n$ length of 10.

$n$ neighbouring information in disambiguation. Taggers performance were measured by comparing morph-complex unknown words they correctly tagged against the total number of morph-complex unknown words in the truth data. Results are shown in table 9.10. FnTBL only got scores in non morph-inflected words that are unknown.

| Corpus | HunPOS | TnT | SLLT |
|---|---|---|---|
| IgbTNT1 | 70.73% | 73.94% | 83.77% |
| IgbTMT | 67.17% | 70.37% | 86.48% |
| IgbTC1 | 70.28% | 73.16% | 84.67% |

Table 9.10: Accuracy scores on morph-complex unknown words

**Experiment 2: Using Morphologically Reconstructed Word-Forms**

We refer to Stanford Log-linear POS Tagger as 'SLLT2' and rule-based tagger "FnTBL2" to differentiate them from 'SLLT' and "FnTBL" used in the experiment 1 and chapter 8 tagging experiment.

Using morphological parser developed, morph-inflected words in table 9.9 were morphologically reconstructed into actual linguistically-informed stems and affixes to form a new training and testing data for SLLT2 and FnTBL2. For example, table 9.11 shows the two states of FnTBL2 for training and testing morph-inflected words in IgbTNT1 reconstructed into the actual linguistically-informed prefixes (PRE), stems (ROOT) and suffixes (SUF). FnTBL2's lexical lookup module that generates its initial state was overridden by morphological parser outputs. From table 9.11, the word "nwukwasị" tagged "VSI_XS" in IgbTNT1 after morphological parsing will be patterned "nwu/ROOT kwasị/SUF" for FnTBL2's initial state and "ROOT" will be changed to "VSI_XS" in FnTBL's truth state. In contrast of FnTBL2 setup, SLLT2 uses the FnTBL2 truth state as its training data, that is, using train data of the form "nwu/VSI_XS kwasị/SUF" (there is no use of ROOT) in its training session.

There are four variations of patterns in data presentation. In each of the following variations, FnTBL2's initial state train data uses ROOT tag for stems while both FnTBL2's truth state and SLLT2 train data use the morph-inflected words true tags as truth for stems. The patterns are:

- Classify stems and associated affixes as ROOT, PRE (prefix) and SUF (suffix) irrespective of their grammatical functions. *Pattern1* of table 9.11.

- Introduce past tense marker (rV) tag for any SUF that is rV. A test on the inflectional class. *Pattern2* of table 9.11.

- Introduce few morph-tags (see table 9.12). This tests the prospect of morph-tags in the Igbo computational morphology. *Pattern3* of table 9.11.

- Collapsed prefix and root together. This tests the strength of prefix as a predictive element considering it is only a single character. In English, addition of prefix as feature caused negative effect on the accuracy of unknown words (Toutanova et al., 2003). *Pattern4* of table 9.11.

| Word form | FnTBL2 Initial State | FnTBL2 Truth State |
|---|---|---|
| | Pattern1 PRE+SUF | |
| nwukwasị | nwu/ROOT kwasị/SUF | nwu/VSI_XS kwasị/SUF |
| nwukwara | nwu/ROOT kwa/SUF ra/SUF | nwu/VrV_XS kwa/SUF ra/SUF |
| nwukwasịrị | nwu/ROOT kwasị/SUF rị/SUF | nwu/VrV_XS kwasị/SUF rị/SUF |
| ịnọdonwu | ị/PRE nọ/ROOT do/SUF nwu/SUF | ị/PRE nọ/VIF_XS do/SUF nwu/SUF |
| abịakwara | a/PRE bịa/ROOT kwa/SUF ra/SUF | a/PRE bịa/VPP_XS kwa/SUF ra/SUF |
| enwechaghị | e/PRE nwe/ROOT cha/SUF ghị/SUF | e/PRE nwe/VSI_XS cha/SUF ghị/SUF |
| | Pattern2 PRE+SUF+rV | |
| nwukwasị | nwu/ROOT kwasị/SUF | nwu/VSI_XS kwasị/SUF |
| nwukwara | nwu/ROOT kwa/SUF ra/rV | nwu/VrV_XS kwa/SUF ra/rV |
| nwukwasịrị | nwu/ROOT kwasị/SUF rị/rV | nwu/VrV_XS kwasị/SUF rị/rV |
| ịnọdonwu | ị/PRE nọ/ROOT do/SUF nwu/SUF | ị/PRE nọ/VIF_XS do/SUF nwu/SUF |
| abịakwara | a/PRE bịa/ROOT kwa/SUF ra/SUF | a/PRE bịa/VPP_XS kwa/SUF ra/SUF |
| enwechaghị | e/PRE nwe/ROOT cha/SUF ghị/SUF | e/PRE nwe/VSI_XS cha/SUF ghị/SUF |
| | Pattern3 Includes All Morpho-tags | |
| nwukwasị | nwu/ROOT kwasị/LSUF | nwu/VSI_XS kwasị/LSUF |
| nwukwara | nwu/ROOT kwa/rSUF ra/rV | nwu/VrV_XS kwa/rSUF ra/rV |
| nwukwasịrị | nwu/ROOT kwasị/rSUF rị/rV | nwu/VrV_XS kwasị/rSUF rị/rV |
| ịnọdonwu | ị/PRE nọ/ROOT do/iSUF nwu/iSUF | ị/PRE nọ/VIF_XS do/iSUF nwu/iSUF |
| abịakwara | a/PRE bịa/ROOT kwa/eSUF ra/APP | a/PRE bịa/VPP_XS kwa/eSUF ra/APP |
| enwechaghị | e/PRE nwe/ROOT cha/xSUF ghị/NEG | e/PRE nwe/VSI_XS cha/xSUF ghị/NEG |
| | Pattern4 Collapsed PRE and ROOT together. All(-PRE) | |
| nwukwasị | nwu/ROOT kwasị/LSUF | nwu/VSI_XS kwasị/LSUF |
| nwukwara | nwu/ROOT kwa/rSUF ra/rV | nwu/VrV_XS kwa/rSUF ra/rV |
| nwukwasịrị | nwu/ROOT kwasị/rSUF rị/rV | nwu/VrV_XS kwasị/rSUF rị/rV |
| ịnọdonwu | ịnọ/ROOT do/iSUF nwu/iSUF | ịnọ/VIF_XS do/iSUF nwu/iSUF |
| abịakwara | abịa/ROOT kwa/eSUF ra/APP | abịa/VPP_XS kwa/eSUF ra/APP |
| enwechaghị | enwe/ROOT cha/xSUF ghị/NEG | enwe/VSI_XS cha/xSUF ghị/NEG |

Table 9.11: Some various patterns of morph-inflected words from 9.9 morphologically reconstructed into stems and affixes to serve as FnTBL2's train data states and SLLT train data (FnTBL2 Truth State)

| Tag/Marker | Meaning |
|---|---|
| APP | Applicative |
| NEG | Negative |
| INFL | Inflection for perfect tense |
| rV | Inflection for past tense |
| LSUF | Last suffix marker for morph-inflected simple verb |
| xSUF | suffix within morph-inflected simple verb |
| eSUF | Suffixes within morph-inflected participle |
| iSUF | Suffixes within morph-inflected infinitive |
| rSUF | Suffixes within morph-inflected past tense verb |

Table 9.12: Morph-tags and meanings

Our plan is for FnTBL2 to generate transformational rules that will only transform ROOT tags to final tags making use of morphology (affixes), these final tags are the tags for morph-complex words that are unknown. The FnTBL2 transformational rules are morphologically context-dependent. That is, the use of prefix, stem, suffix and their positions within morph-inflected words to predict the appropriate tags of these words. For example, in figure 9.1, "kwa" occur more frequently in VSI_XS class than other classes. It occur mostly in positions 1 and 2 and less frequently in positions 3 and 4. Compare this figure with table 9.11 by observing the positions of "kwa" in each word in the table and their classes.

In relation to the morphological contexts illustrated in figure 9.1, the idea is to use those morphological contexts in the following way:

*Change current TAG to new TAG if current TAG and one or more of* {***SPACE, PREFIX, STEM, SUFFIX(s) and/or their POSITIONS*** } *happen.*

These are samples of FnTBL2 transformational rules generated after training using data in table 9.11

1. *pos_0=VPERF word:[1,3]=kwa => pos=VPERF_XS*, this rule means if current tag is VPERF (perfect tense) and there is "kwa" in either position 1 or 3 after stem, change VPERF to VPERF_XS. Observe from figure 9.1 that "kwa" occurs at positions 1 to 4 (frequently at positions 2 and 3) for VPERF_XS class.

2. *pos_0=VrV_XS word_2=kwa => pos=VSI_XS*, this means if the current tag is VrV_XS (past tense morph-inflected verb) and suffix at position 2 after stem is "kwa", change current tag to VSI_XS. This rule is that the probability of a morph-inflected word's class when a suffix "kwa" is observed generally at position 2 after stem is VSI_XS. This is because from the statistics in figure 9.1 show that "kwa" is more frequent at position 2 of a morph-inflected word in VSI_XS class than other classes. This rule is general to certain degree, for example, there are 437 places this rule fired in the test data and about 47% of it got positive impact while remainder are places where resultant tags should have been other tags like VrV_XS, VPP_XS or VPERF_XS. Next item is one of the ways of dealing with this rule.

3. *pos_0=VSI_XS word_0=pụ word_1=ta word_2=kwa => pos=VrV_XS*, if the current tag is VSI_XS and stem is "pụ" and next two suffixes are "ta" and "kwa", change

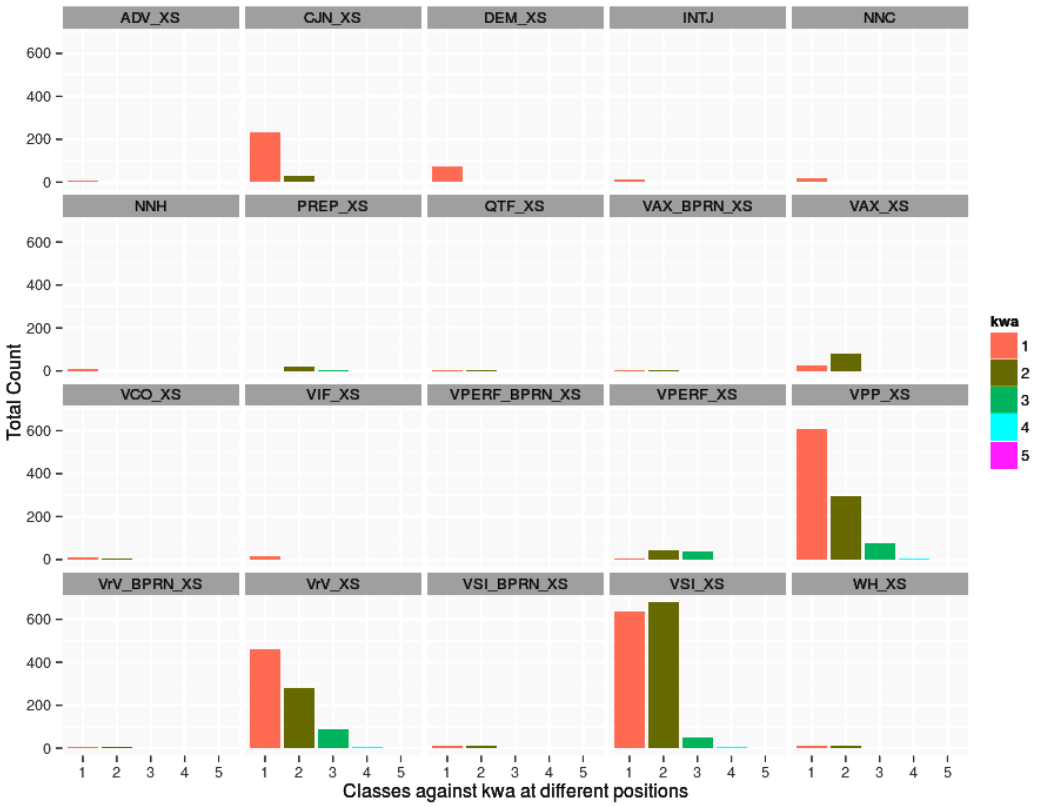



Figure 9.1: Thorough analysis of the suffix "kwa". Also observe that classes are skewed to verbs as positions shift away right from the stem. 1 is immediately after the stem

current tag to VrV_XS. That is, the probability of a morph-inflected word's class when a suffix "kwa" is observed at position 2 along with contexts *stem ="pụ" and immediate suffix after stem = "ta"* within the same vicinity is VrV_XS. This rule scored 100% positive impact on the test data. It fired at six places and made zero negative impact. That means when suffix "kwa" at position 2 occurred along with stem "pụ" and another suffix "ta" at position 1, it is more VrV_XS than VSI_XS.

4. *pos_0=VPP_XS word_0=nwe word_1=kwa word_2=ra => pos=VSI_BPRN_XS*, this implies that if current tag is VPP_XS and stem is "nwe" and suffixes at positions 1 and 2 after stem are "kwa" (see figure 9.1) and "ra", change VPP_XS to VSI_BPRN_XS (VSI_XS that is pronoun bound).

| Prefix | Meaning |
|---|---|
| a/e | indicates verb is participle if preceded by auxiliary |
| n/m | indicates noun or gerund formed through nominalization |
| ị/i | indicates infinitive verb |
| o/ọ | indicates noun or gerund formed through nominalization |
| u/ụ | indicates noun or gerund formed through nominalization |

Table 9.13: Prefixes and their meaning

Justification to override TBL's lexical module for generating initial state is to trick FnTBL into generating only rules that change "ROOT" to its appropriate tag using prefixes and suffixes clues. This is to avoid generating some "stupid" rules that will learn to change prefixes/suffixes tags which might cause "false" context to stand in place of "true" context that would have been helpful in TBL's decision making, thereby hindering good performance. For example, in IgbTMT corpus, overriding TBL's lexical module performed better by scoring 91.99% accuracy with **only** 55 rules compared to when used TBL's lexical module that scored 88.95% accuracy with 125 rules. The rules and accuracy scores are average computations on 10-fold cross validation over each corpus size.

We performed tagging with FnTBL2 and SLLT2 on all the patterns mentioned above. FnTBL2 was used because it works well on patterns using its inductive means and SLLT because it outperformed others in experiment 1. Results are shown in table 9.14.

| Corpus | Taggers | Pattern1 | Pattern2 | Pattern3 | Pattern4 |
|---|---|---|---|---|---|
| IgbTNT1 | FnTBL2 | 78.03% | 82.81% | 90.44% | 82.78% |
| | SLLT2 | 66.31% | 67.11% | 66.53% | 70.87% |
| IgbTMT | FnTBL2 | 78.96% | 86.03% | 91.99% | 85.95% |
| | SLLT2 | 74.45% | 75.27% | 76.01% | 77.15% |
| IgbTC1 | FnTBL2 | 83.75% | 86.23% | 88.46% | 83.27% |
| | SLLT2 | 76.41% | 77.62% | 76.09% | 76.54% |

Table 9.14: Accuracy scores on the morph-complex words based on different approaches

### 9.4.5 Discussions

Table 9.15 show how the root and the associated affixes served as important cues for predicting tags of the morph-complex unknown words.

| Inflected Word | TBL Test Data | TBL Transformation Rule | TBL Transformed Tag |
|---|---|---|---|
| | | Pattern1 | |
| begorochaa | be/ROOT go/SUF ro/SUF chaa/SUF | $r0$: ROOT => VrV<br>$r1$: VrV => VrV_XS<br>$r32$: VrV_XS => VSI_XS | VSI_XS |
| kwụpụta | kwụ/ROOT pụ/SUF ta/SUF | $r0$: ROOT => VrV<br>$r1$: VrV => VrV_XS | VrV_XS |
| wukwasịkwara | wu/ROOT kwasị/SUF kwa/SUF ra/SUF | $r0$: ROOT => VSI_XS<br>$r4$: VSI_XS => VrV_XS<br>$r6$: VrV_XS => VSI_XS | VSI_XS |
| ghọgburu | ghọ/ROOT gbu/SUF ru/SUF | $r0$: ROOT => VSI_XS<br>$r4$: VSI_XS => VrV_XS | VrV_XS |
| | | Pattern2 | |
| wukwasịkwara | wu/ROOT kwasị/SUF kwa/SUF ra/rV | $r0$: ROOT => VSI_XS<br>$r5$: VrV => VrV_XS | VrV_XS |
| ghọgburu | ghọ/ROOT gbu/SUF ru/rV | $r0$: ROOT => VSI_XS<br>$r1$: VSI_XS => VrV<br>$r3$: VrV => VrV_XS | VrV_XS |
| | | Pattern3 | |
| begorochaa | be/ROOT go/xSUF ro/APP chaa/LSUF | $r0$: ROOT => VrV<br>$r1$: VrV => VSI_XS | VSI_XS |
| kwụpụta | kwụ/ROOT pụ/xSUF ta/LSUF | $r0$: ROOT => VrV<br>$r1$: VrV => VSI_XS | VSI_XS |
| wukwasịkwara | wu/ROOT kwasị/rSUF kwa/rSUF ra/rV | $r0$: ROOT => VSI_XS<br>$r5$: VrV => VrV_XS | VrV_XS |

Table 9.15: Examples of some transformational rules generated by FnTBL2 that fired and their transformational trails and final predicted tags

Table 9.15 has the following elements: inflected word from IgbTNT and IgbTMT, *TBL test data* contains reconstructed morph-inflected words, *TBL transformation rule* is an ordered rule list FnTBL generated during training session using data in table 9.11, and *TBL transformed tag* is the final tag FnTBL predicted, which is returned as appropriate tag for the morph-complex unknown word. This table contains summaries of some rules that fired resulting to the predicted tags in patterns 1 and 3. The average number of rules generated in all patterns are: there are 119, 53 and 55 rules generated in patterns 1, 2 and 3 from IgbTMT corpus, and there are 87, 60 and 60 rules generated in patterns 1, 2 and 3 from IgbTNT. See table 9.14 for tagging accuracy.

The followings are the explanations of the rules that transformed initial tags "ROOT" to final tags:

1. The inflected word "begorochaa" (from IgbTMT corpus) is a morph-inflected simple verb (VSI_XS) with "be" as the stem and three suffixes. Comparing the transformational rules in the patterns:

   - There are three rules that fired in pattern1 to transform the initial tag "ROOT" to the final tag "VSI_XS", while only two fired in pattern3 to transform the initial tag "ROOT" to the final tag "VSI_XS".
   - Contexts used by pattern1 rules that transformed "ROOT" to "VSI_XS":
     - $r0$: *pos_0=ROOT word:[-2,-1]=ZZZ => pos=VrV*, this means if current tag is ROOT and there is a boundary marker (ZZZ)[15] found within the

[15]This is at the end of every data instance (usually a sentence, but in this case, a word segmented

previous two positions, change ROOT to VrV or VSI_XS. This is prefix[16] found within the previous two positions that changed ROOT to VrV.

- $r1$: *pos_0=VrV pos_1=SUF pos_2=SUF => pos=VrV_XS*, this means if current POS tag is VrV and next two tags are SUFs (tags in positions 1 and 2 after stem) change VrV to VrV_XS. This rule signifies that there are more inflected past tense verbs (*pre-root-suf(s)-rV and root-suf(s)-rV*) in the corpus.
- $r32$: *pos_0=VrV_XS word:[1,3]=chaa => pos=VSI_XS*, this means if current tag is VrV_XS, and the suffix "chaa" is found in any position within the range of 1 to 3 after stem, change VrV_XS to VSI_XS.

- Contexts used by pattern3 rules that transformed "ROOT" to "VSI_XS":
  - $r0$: Same as pattern1.
  - $r1$: *pos_0=VrV pos:[1,3]=LSUF => pos=VSI_XS*, that is, if current tag is VrV and there is tag "LSUF" found within the next three positions (1 to 3), change VrV to VSI_XS. From table 9.12, LSUF is a morph-tag to mark the last suffix of all inflected simple verb (VSI_XS).

2. The inflected word "kwụpụta" (from IgbTMT corpus) is a morph-inflected simple verb (VSI_XS) with "kwụ" as the stem and two suffixes. Comparing the transformational rules in the patterns:

   - Contexts used by pattern1 rules that transformed "ROOT" to "VrV_XS":
     - $r0$: Same as $r0$ in "begorochaa".
     - $r1$: Same as $r1$ in "begorochaa". This rule wrongly transformed ROOT to VrV_XS because the context is too general. This is handle in pattern3.
   - Contexts used by pattern3 rules that transformed "ROOT" to "VSI_XS":
     - See pattern3 of "begorochaa" in the above item.

3. The inflected word "wukwasịkwara" (from IgbTNT corpus) is a morph-inflected past tense verb (VrV_XS) with "wu" as the stem and three suffixes. Comparing the transformational rules in the patterns:

   - There are three rules that fired in pattern1 transforming ROOT to VSI_XS instead of VrV_XS. In patterns 2 and 3, there are two rules that fired, each transforming ROOT to VrV_XS.
   - Contexts used by pattern1 rules that transformed "ROOT" to "VSI_XS":
     - $r0$: Same as $r0$ in "begorochaa" except that the output tag is VSI_XS.
     - $r4$: *pos_0=VSI_XS pos_1=SUF pos_2=SUF => pos=VrV_XS*, this rule is the same with $r1$ of "begorochaa" except that the current tag is VSI_XS.

into its morphemes), so that transformation-based rules can refer to this as a context element, so as to "anchor" their use to either beginning or end of the sequence.

[16]There is only a single length prefix found in some words in Igbo.

- $r6$: *pos_0=VrV_XS word_2=kwa => pos=VSI_XS*, this implies that if current tag is VrV_XS and the next two suffix is "kwa", change VrV_XS to VSI_XS. This rule changed what would have been the correct tag back to tag $r0$ changed. Notice in figure 9.1 that "kwa" occurs more in VSI_XS class than other classes. This is corrected in the following patterns 2 and 3.

- Contexts used by pattern2 rules that transformed "ROOT" to "VrV_XS":
  - $r0$: Same as $r0$ in pattern1 of this item.
  - $r5$: *pos_0=VSI_XS pos:[1,3]=rV => pos=VrV_XS*, this implies that if current tag is VSI_XS and there exist a tag "rV" in any position within the range of 1 to 3 after stem, change VSI_XS to VrV_XS.

- Contexts used by pattern3 rules that transformed "ROOT" to "VrV_XS":
  - $r0$: Same as $r0$ in pattern1 of this item.
  - $r5$: *pos_0=VSI_XS pos_2=rSUF => pos=VrV_XS*, this implies that if current tag is VSI_XS and there exist a tag "rSUF" in next two position, change VSI_XS to VrV_XS. From table 9.12, rSUF is a suffix marker to identify suffixes found within inflected past tense verbs (VrV_XS).

4. The inflected word "ghụgburu" (from IgbTNT corpus) is a morph-inflected past tense verb (VrV_XS) with "ghụ" as the stem and two suffixes. Comparing the transformational rules in the patterns:

   - Contexts used by pattern1 rules that transformed "ROOT" to "VrV_XS":
     - $r0$: Same as $r0$ in pattern1 of "wukwasịkwara".
     - $r4$: Same as $r4$ in pattern1 of "wukwasịkwara". This rule is too general, though it favours inflected past tense verbs (VrV_XS) but disfavours inflected simple verbs (VSI_XS). See item above where inflected word "kwụpụta" is discussed.

   - Contexts used by pattern2 rules that transformed "ROOT" to "VrV_XS":
     - $r0$: Same as $r0$ in pattern1 of "wukwasịkwara".
     - $r1$: *pos_0=VSI_XS pos:[1,2]=rV => pos=VrV*, this implies that if current tag is VSI_XS and there exist "rV" tag within next positions (1 to 2), change VSI_XS to VrV.
     - $r3$: *pos_0=VrV pos_2=rV => pos=VrV_XS*, that means if current tag is VrV and tag "rV" is found in next two position after stem, change VrV to VrV_XS. pos_2=rV means there is a SUF in pos_1.

We observe that rules $r0_t$ and $r0_z$ used the same context but gave different tags (VrV and VSI_XS). The context is "pos_0=ROOT word:[-2,-1]=ZZZ", which implies, if tag is ROOT and there is a boundary marker (ZZZ) found within the previous two positions, change ROOT to VrV or VSI_XS. This is prefix[17] optional, meaning if the immediate previous position is prefix, then the next previous position is a boundary marker, otherwise

[17]There is only a single length prefix found in some words in Igbo.

the immediate previous line is boundary marker. The reason for $r0_t$ and $r0_z$ using the same context, but gave different tags is dependent on the two corpora used. IgbTMT contains more words having only past tense inflectional (INFL[18]) part (rV), that is words of the form *root-rV*, in the training data than IgbTNT. There are *root-rV, pre-root-rV, pre-root-suf(s)-rV, root-suf(s)-rV* forms, and if we exclude *root-rV*, the output tag of FnTBL using the same context changes to VSI_XS for IgbTMT, that is, "pos_0=ROOT word:[-2,-1]=ZZZ => VSI_XS" is the same as IgbTNT. Obviously, this is due to different styles of writing in both texts.

The accuracy scores of both experiments are shown in tables 9.10 and 9.14. The accuracy scores are only for morph-complex unknown words class in IgbTNT1, IgbTMT and IgbTC1. There are four variations in the second experiment which is based on the way morph-inflected words are patterned (see table 9.11). Comparing the accuracy scores of both experiments:

- *pattern1*: "PRE+SUF" of tables 9.14 and 9.11 is where we performed tagging that only recognizes morphological elements before and after a stem as "PRE" (prefix) and "SUF" (suffix) respectively. The results shows that FnTBL did better than SLLT2 in all cases. Comparing with table 9.10 of first experiment, FnTBL2 performed better than other taggers except SLLT.

- *pattern2*: "PRE+SUF+rV" in tables 9.14 and 9.11 shows where we added a tag "rV" to indicate the past tense presence in a morph-inflected words. This generally improves the accuracy of the results for FnTBL2 and SLLT2, and FnTBL2 scored better than majority of taggers in first experiment (see table 9.10).

- *pattern3*: "All" in tables 9.14 and 9.11 is where we introduced more tags, tags to indicate grammatical functions and suffixes found within a morph-inflected words. For example, suffixes having "rV" form can indicate past tense or applicative, therefore we introduced "APP" for applicative while "rV" is for past tense, "ghị" suffix unambiguously indicate negation (NEG), "eSUF" to mark other suffixes if it is not "APP, NEG, INFL and rV", etc.. Table 9.12 shows examples of all the morph-tags and explanations. This gave best scores of 90.44%, 91.99% and 88.46% for FnTBL2. These scores are several points better than scores achieved by taggers used in the first experiment (see table 9.10).

- *pattern4*: Prefixes in *pattern3* were collapsed to their stems "All(-PRE)" in tables 9.14 and 9.11. This is to verify if prefix is a good predictive feature in Igbo considering it is only one character length. Comparing columns "All(-PRE)" and "All" in table 9.14 shows that there are lot of figures lost in the accuracy scores of column "All" for FnTBL2 (e.g. about 7.66% in IgbTNT1). This is contrary to English where addition of prefix as feature caused negative effect on the accuracy of unknown words (Toutanova et al., 2003). Surprisingly, SLLT2 increased in its accuracy against decrease in FnTBL2 scores. But an experiment using SLLT tagger's technique for

[18]Inflection in Igbo comprises two parts: past tense (rV) and perfect tense (PERF). "VPERF" and "VrV" are inflectional tags for perfect and past tenses of verbal words that have no suffixes. The words are inflected by "-rV" for past tense and "-gV, -lV, -wV" for perfect tense, where "r,g,l,w" are letters and "V" is any vowel letter (a,ọ,ụ,ị,e,o,u,i). See tagset in appendix for description.

handling unknown words ($SLLT^s$ and $SLLT^{sp}$) in table 8.11 of chapter 8 shows that using only suffix features ($SLLT^s$) on IgbTMT gave accuracy of 70.76% for general unknown words and 77.26% for morph-complex words that are unknown, and addition of prefix features $SLLT^{sp}$ improved the accuracy on the morph-complex unknown words by 5.44% and general unknown words by 9.46%. The reason for SLLT2's accuracy increment can be explained in regard with "PRE" ambiguity and the small size of the training data (see table 9.9). "PRE" tag is used to indicate prefix whether it is "ị/i" for infinitive or "a/e" for participle and simple verbs, therefore, collapsing it with the stem removes this ambiguity.

## 9.5 Taggers Accuracy on Morph-Complex Words

The rare integrants of Igbo words constitute mainly words that are morphologically complex. This has been shown on table 9.9. Table 9.16 shows the accuracy scores of various implementations of handling unknown words by combining the method illustrated above and taggers produced in chapter 8. We used the FnTBL2 results in the experiments performed in section 9.4.4 to improve accuracy of the general unknown words, therefore TBL in table 9.16 is referring to FnTBL2. Explaining items in table 9.16, *corpus column* is IgbTC1 representing the combination of IgbTNT1 and IgbTMT in a stratified method. Next column is the *size* of words that are unseen during training session (unknown words), followed by the *sizes of morph-complex unknown words* in the unknown words. *Taggers column* contains various combinations of taggers and classifier. For example, FnTBL is where a single tagger was used only, FnTBL+TBL means choosing the tags FnTBL2 predicted for morph-complex unknown words over the tags predicted by the single tagger, FnTBL+TBL$^{nb}$ means using Naive Bayes classifier (NBC) to choose the best predicted tags between FnTBL2 and a single other tagger for morph-complex unknown words, and FnTBL+TBL$^G$ means choosing correct tags for morph-complex unknown words from FnTBL2 and single tagger predicted tags by comparing their tags with the gold standard tags. NBC selects tag1 for tagger1 or tag2 for tagger2 (the best choice) given those features used by both taggers in making their predictions. For example, SLLT+TBL$^{nb}$ is 0.71% better than SLLT+TBL, and 3.42% better than SLLT single tagger, that means, 3.42% is the percentage of correct tags NBC selected from both FnTBL2 and SLLT. If we decide to do away with tags predicted for morph-complex unknown words by SLLT and make use of FnTBL2 predicted tags, we end up having 80.23% and loosing 0.71% that might come from choosing those tags SLLT predicted right for morph-complex unknown words. SLLT+TBL$^G$ reveals that there are still 2.88% percentage of correct tags NBC did not get.

| | Sizes | | | Taggers | Accuracy Scores | | |
|---|---|---|---|---|---|---|---|
| Corpus | # of Unk | # of Infltok in Unk | # of Non-Infl in Unk | | Unk acc | Infltok acc | Non/Infl acc |
| IgbTC1 | 271 | 193 | 78 | FnTBL | 14.20% | 00.00% | 50.60% |
| | | | | FnTBL+TBL | 77.65% | 88.46% | |
| | | | | FnTBL+TBL$^{nb}$ | 77.65% | 88.46% | |
| | | | | FnTBL+TBL$^{G}$ | 77.65% | 88.46% | |
| IgbTC1 | 271 | 193 | 78 | MBT | 14.67% | 00.00% | 48.97% |
| | | | | MBT+TBL | 77.18% | 88.46% | |
| | | | | MBT+TBL$^{nb}$ | 77.18% | 88.46% | |
| | | | | MBT+TBL$^{G}$ | 77.18% | 88.46% | |
| IgbTC1 | 271 | 193 | 78 | HunPOS | 65.05% | 70.28% | 52.02% |
| | | | | HunPOS+TBL | 78.00% | 88.46% | |
| | | | | HunPOS+TBL$^{nb}$ | 77.80% | 88.19% | |
| | | | | HunPOS+TBL$^{G}$ | 81.28% | 93.08% | |
| IgbTC1 | 271 | 193 | 78 | TnT | 66.42% | 73.16% | 49.86% |
| | | | | TnT+TBL | 77.32% | 88.46% | |
| | | | | TnT+TBL$^{nb}$ | 77.81% | 89.19% | |
| | | | | TnT+TBL$^{G}$ | 80.94% | 93.56% | |
| IgbTC1 | 271 | 193 | 78 | SLLT | 77.52% | 84.67% | 59.78% |
| | | | | SLLT+TBL | 80.23% | 88.46% | |
| | | | | SLLT+TBL$^{nb}$ | 80.94% | 89.50% | |
| | | | | SLLT+TBL$^{G}$ | 83.82% | 93.52% | |

Table 9.16: Average accuracy scores on the overall unknown words (Unk acc), morph-complex unknown words that are verbs (Infltok acc) and remainders (Non/Infl acc: mostly non inflected unknown words and few other classes that are inflected)

| Test data | Size | Unknown word[a] | MI in UWR[b] |
|---|---|---|---|
| ESSAY | 2921 | 177 | 93 |
| NEWS | 407 | 80 | 16 |
| POEM | 584 | 83 | 30 |
| STORY | 248 | 11 | 11 |

Table 9.17: Statistics of dissimilar texts used

[a]Words in test data not seen in the train data. Train data is IgbTC. See table 8.8 in chapter 8 for IgbTC description.

[b]Proportion of unknown words that are morph-inflected.

We compare the performances of taggers generated from IgbTC on dissimilar Igbo texts. For this experiment, we collected the following text styles from the web: news, essay, poem, and story texts. Table 9.17 shows the sizes of tokens and morphologically-complex unknown words in different Igbo texts we collected compared with IgbTC[19]. For each text collected, we used Igbo morphological parser (discuss in section 9.1) to detect, reconstruct and classify words that are unknown and morphologically complex. For example, ESSAY is 0.96% of IgbTC, and there are 93 morphologically-complex words detected in ESSAY not found in IgbTC. We judge taggers performance based on these detected words that are labelled unknown and morphologically complex. Table 9.18 shows the performance scores of all taggers used. SLT, HunPOS and TnT are taggers trained on IgbTC, while TBL is FnTBL tagger trained only on morphologically-inflected words found in IgbTC. It uses morphological clues in handling morphologically-complex words in Igbo, while

[19]See table 8.8 in chapter 8 for description.

| | Inflsc[a] | | | | TBL Alongside Other Taggers | | |
|---|---|---|---|---|---|---|---|
| Test data | Hun | TnT | SLT | TBL[b] | Unksc[c] | Overall | Tagger |
| ESSAY | 67.74 | 67.74 | 89.25 | 91.40 | 79.10 | 90.14 | slt |
| | | | | | 80.23 | 90.21 | slt+tbl[d] |
| | | | | | 50.28 | 87.98 | tnt |
| | | | | | 62.71 | 88.74 | tnt+tbl[d] |
| | | | | | 51.98 | 88.05 | hun |
| | | | | | 64.41 | 88.81 | hun+tbl[d] |
| NEWS | 56.25 | 56.25 | 68.75 | 81.25 | 41.25 | 86.24 | slt |
| | | | | | 43.75 | 86.73 | slt+tbl |
| | | | | | 56.25 | 87.47 | tnt |
| | | | | | 61.25 | 88.45 | tnt+tbl |
| | | | | | 75.00 | 93.37 | hun |
| | | | | | 80.00 | 94.35 | hun+tbl |
| POEM | 36.6 | 33.33 | 70.00 | 86.67 | 39.76 | 78.60 | slt |
| | | | | | 45.78 | 79.45 | slt+tbl |
| | | | | | 20.48 | 75.17 | tnt |
| | | | | | 39.76 | 77.91 | tnt+tbl |
| | | | | | 25.30 | 77.74 | hun |
| | | | | | 43.37 | 80.31 | hun+tbl |
| STORY | 63.64 | 90.91 | 72.73 | 100.00 | 65.62 | 91.13 | slt |
| | | | | | 75.00 | 92.34 | slt+tbl |
| | | | | | 59.38 | 89.92 | tnt |
| | | | | | 62.50 | 90.32 | tnt+tbl |
| | | | | | 40.62 | 88.31 | hun |
| | | | | | 53.12 | 89.92 | hun+tbl |

Table 9.18: Percentage performances of taggers developed from IgbTC on different styles of texts in Igbo

[a]Performance scores of taggers on morphologically-complex unknown words.

[b]Tagger produced on morphologically-inflected words. Morphologically-inflected words are generated using morphological parser develop in section 9.1.

[c]Performance scores of taggers on unknown words.

[d]Both taggers are run alongside each other. TBL predicted tags on morphologically-complex unknown words replace tags predicted by other taggers.

| Word | Tag | TBL | SLT | TnT | HunPOS |
|---|---|---|---|---|---|
| lechasịrị | VrV_XS | VrV_XS | VrV_XS | VrV_XS | NNC |
| jụa | VSI_XS | VSI_XS | NNC | VSI_XS | VIF_XS |
| atọkarịcha | VSI_XS | VSI_XS | NNC | VSI_XS | VSI_XS |
| ịbọkasị | VIF_XS | VIF_XS | VIF_XS | NNCV | NNCV |
| pụa | VSI_XS | VSI_XS | NNC | VSI_XS | NNC |

Table 9.19: Some samples of taggers output

*SLT+TBL, TnT+TBL and HunPOS+TBL* are when both taggers are run alongside each other. TBL predicted tags on morphologically-complex unknown words replace the tags predicted by other taggers. The performance scores reveal that TBL outperformed other

taggers with several points notwithstanding that it was trained only on the morphological elements of words in IgbTC that are morphologically-inflected, and training was done without use of any sentence clues. Table 9.19 shows samples of morphologically-inflected words in STORY texts not found in IgbTC that taggers wrongly classified except TBL.

## 9.6 Conclusion

We have shown that stems and associated affixes are good for predicting appropriate tags for morph-complex unknown words and morph-inflected words improperly tagged in Igbo tagged corpus (IgbTC). Through morphological reconstruction, a linguistically-informed segmentation of stems and affixes, morph-inflected words are represented in machine learnable pattern. Taggers exploit these morphological characteristics during tagging process for predicting appropriate tags for wrongly tagged morph-inflected words and handling unknown words that are morph-complex. Increase in the accuracy of taggers on IgbTC after the application of an automatic error correction method developed using this morphological reconstruction method reveals that the error correction method is effective.

Also, the performance of FnTBL2 that inductively learns linguistic patterns reveals that using actual linguistically-informed affixes as word features for morph-complex words in Igbo is better than the existing word feature extraction methods. Most existing taggers only extract strings of characters from last (sometimes first) letters of a word up to length of $n$. This will be unable to capture various forms of morphemes associated with the morph-complex words that form the majority of unknown words in IgbTC. In Igbo language, a single root can produce as many possible word-forms as possible, which is possible through using affixes of varying lengths from 1 to 5, which only extends the original meaning rather changing it.

Naive Bayes classifier (NBC) was used to improve on the accuracy of the general unknown words. Different taggers with different implementation techniques for handling unknown words were applied on the morph-complex unknown words in IgbTC. Among these taggers, TBL (FnTBL2) rule-based tagger outperformed them but there are places in the used corpus where TBL failed that these taggers passed. The NBC classifier was used to choose arbitrarily those appropriate tags for the morph-complex unknown words between TBL and one other taggers (tagger2). NBC selecting tag1 for TBL or tag2 for tagger2 is dependent on those features they used in making their predictions. The accuracy scores of the general unknown words increased.

A comparative analysis that involves the use of taggers trained on IgbTNT and IgbTMT, parts of the main corpora IgbTC, to tag dissimilar Igbo texts indicates that the use of linguistically-motivated approach as an extra knowledge-source to the taggers is suitable for processing words that are morphologically-inflected in Igbo. This method achieves impressive accuracy scores of the range 82%-100% while other taggers achieves the range of 33%-90% accuracy scores on the four dissimilar texts of the language. When we run other taggers that did not use this extra knowledge-source alongside the TBL that used it which replaces the tags predicated by the former with the tags predicted by the latter on the morphologically-complex unknown words, the former accuracy scores increases on the unknown words.

# Chapter 10

# Summary and Future Work

## 10.1 Summary

This thesis set out to build Natural Language Processing (NLP) resources for Igbo language which has not featured in NLP research mainstream, and to use NLP and machine learning existing computations to make the process more efficient. In this last chapter, we will review the research contributions of this thesis, as well as discuss directions for future research.

### 10.1.1 Contributions

Before April 2013, there was no single literature or tool available for NLP in Igbo language. Although there are Igbo electronic texts and linguistic literatures. Today, an African language Igbo can boast of some published NLP papers presented at good ranked conferences, and some contents of Basic Language Resources Kits (BLARK). Igbo people are set to enjoy the benefits of NLP technology for computer use and information access, which will contribute to their communication within the global information society. This research has spurred two (more to join) other Igbo native speakers into taking part in Igbo*NLP* research projects. Also, some NLP researchers we met at different conferences are already asking for the developed resources for Igbo. These are as a result of great efforts made in this research. Developing NLP tools for the first time for any new language is a non trivial task, but we succeeded in developing some NLP resources for Igbo. The followings enumerate and succinctly explain main research contributions of this thesis.

#### Framework

This (chapter one) provides a well organized platforms in terms of motivations and objectives that are springboard of the various NLP developments in this research.

#### Backgrounds

This contains a well researched and detailed backgrounds focusing on Igbo linguistic literatures (chapter two), tagset and corpora that covered African and non African languages (chapter three), NLP and machine learning algorithms required for this research (chapter four).

### The Corpus

This contains the developmental stages ranging from texts collection to data preparations (like tokenization and normalization methods), and corpus analysis (chapter five). So far, a corpus size of about 1 million has been developed for Igbo. It comprises six genres, viz; religious, novel, poem, story, easy, and news texts. This has been presented in "Tagset and Corpus Development for Igbo, an African Language" LAW III paper (one of COLING workshops in 2014).

### Linguistic Annotation and Tagged Corpus Improvement

About 300k of the I million Igbo corpus has been part-of-speech (POS) tagged using annotation scheme (tagset) developed in this research (chapter six). The methods used for transferring the linguistic materials from the tagset to the selected 300k corpus are manual and automatic (chapters six and seven). Methods are novel and parts of them are in the papers presented in "Tagset and Corpus Development for Igbo, an African Language" LAW III paper (one of COLING workshops in 2014); "Use of Transformation-Based Learning in Annotation Pipeline of Igbo, an African Language" at Joint Workshop on Language Technology in RANLP conference 2015; and "Improving accuracy of Igbo Corpus Annotation Using Morphological Reconstruction and Transformation-Based Learning" at TALAf workshop in JEP-TALN-RECITAL 2016.

### Part-of-Speech Tagging

This contains POS taggers developed for Igbo and exploratory experimental results on POS tagging the 300k Igbo tagged corpus (IgbTC) (chapter eight). POS tagging was done on similar and dissimilar texts which discussed the issues of moving from one genre to another. Interesting features about the language compared to other languages were identified.

### Igbo Morphological Features

Interesting morphological features are found, and discussed (chapter nine). We developed an automatic method that uses morphological reconstruction (a linguistically segmentation into roots and affixes) to find appropriate tags for all morphologically-inflected words that were not tagged properly in the corpus. Also we developed an approach that exploited the morphological features of Igbo to solve poor handling of unknown words (previously unseen words during training session of taggers) by existing taggers. We found out that unknown words in Igbo are mostly the cause of morphologically-complex words unlike in English where it is mainly nouns. Morphologically-complex words are due to addition of more affixes to Igbo words. Our analysis shows that Igbo words become less frequent and then complex when suffixes in an inflected word are from 3 upwards[1] (chapters eight and nine). Part of this research has been presented in "Predicting Morphologically-Complex Unknown Words in Igbo", 19th International Conference on Text, Speech and Dialogue (TSD2016), Brno, Czech Republic. To appear in Volume 9924 of the Lecture Notes in Computer Science series published by Springer.

[1]Highest number of suffixes seen so far in an inflected word is 6.

## 10.2 Future Research

We are going to discuss here some direction for future work.

### 10.2.1 The BLARK

A substantial effort is needed to further research towards expanding the Basic Language Resources Kits (BLARK) for Igbo language. For example, an European-based language processing pipeline follows the sequential processing steps of tokenizing, tagging, morphological analysis, syntactic parsing, etc. We have successfully achieved the first two steps and partial morphological analysis.

### 10.2.2 More Improvement on the Unknown Words

Apart from method described already in this thesis, improving performance on unknown words is mostly not a matter of being able to better distinguish between nouns and verbs but could be helpful. For example, one could opt for backing off to a regular expression tagger, for which regular expressions to detect verbal morphology would need to be developed. If no verbal morphology is detected, the word is probably a noun. Alternatively one may want to experiment with training a character-level ngram model on distinguishing verbs from nouns, or some other kind of classifier that is sensitive to morphology.

There are things to do in order to increase the current performance on the unknown words. Considering *consonant* and *vowel* word shapes of Igbo words, there are possible clues this will be revealing: if a word is not found in this shape will treated as foreign words, can help differentiate verbs and nouns and possibly other classes. Also, a positional marker can be used to trace how often a suffix/enclitics occur at a particular position in an inflected word form.

### 10.2.3 A Single POS Tagger for Realtime Operation

Majority of the unknown words in Igbo is caused by morphological inflection. We used morphological reconstruction to perform low scale morphological analysis of these words which helped to improve accuracy on the unknown words (discussed in the immediate previous chapter). This shows that morphological characteristics are very important cues for predicting unknown words (especially morphologically-complex words) tags in Igbo. Igbo language needs a POS tagger that will extract actual linguistically-informed affixes based on morphological order of occurrence for handling morphologically-complex words on full-scale basis that the trained tagger has not previously seen. A feature extraction for Igbo tagger should be able to extract stems, prefixes and suffixes, such as *a_PREFIX bịa_STEM gha_SUFFIX chi_SUFFIX rịrị_SUFFIX*, given any morphologically-inflected word. This is a step towards effective full scale computational morphology in Igbo. This may involve an integration of a morphological parser component in any existing tagger that has done well on the language in order to build a single tagger for Igbo. This integration system is different from (Loftsson, 2007) definition where an exiting component of a tagger can be integrated into another tagger. A single realtime POS tagger will enable

use of contextual information on sentential level alongside with morphological cues for prediction.

### 10.2.4 Towards Developing Large Corpus Size for Igbo

Igbo language has 30 different writing conventions. Homogeneous collection of electronic texts in this research usually ends up in heterogeneous electronic text form. This issue prevented us from collecting large sized texts from different genre to avoid inducing errors like wrong word type size in the corpus. For example, a corpus can contain *nine, nile, niile,* – they mean 'all', which suppose to be one token, instead they will be counted as unique tokens, which is creating wrong number of word types. A letter-based normalisation or transformation system that will recognize words of other dialects and normalize their strings to the standard dialect's strings is valuable.

### 10.2.5 Morphological Computation

It is important to perform full morphological analysis on Igbo. Experiment in chapter 9 excludes some morph-inflected classes (like nouns) as it will lead to full morphological analysis which is beyond the research scope. Also, morphological analysis on the compound verbs and exploiting $n$ neighbouring words contexts are ignored. These lapses will hide some important information required for NLP task. Of course, this is pointing towards building a large-scale computational morphologies for Igbo. Dealing with noun multiword expressions is also essential.

# Bibliography


Acharya, J. (1991). *A descriptive grammar of Nepali*. Washington, D.C.: Georgetown University Press.

Adedjouma, S. A., John, O. R. A., and Mamoud, I. A. (2013). Part-of-speech tagging of yoruba standard, language of niger-congo family. *Research Journal of Computer and Information Technology Sciences*, 1:2–5.

Agbo, M. S. (2013). Orthography theories and the standard igbo orthography. *Language in India*, 13(4).

Aibek, M., Zhandos, Y., Islam, S., and Anuar, S. (2014). On certain aspects of kazakh part-of-speech tagging. In *Application of Information and Communication Technologies (AICT), 2014 IEEE 8th International Conference*, pages 1–4. IEEE.

Al-Sulaiti, L. and Atwell, E. S. (2006). The design of a corpus of contemporary arabic. *International Journal of Corpus Linguistics. John Benjamins Publishing Company*, 11(2):135–171.

Alejandro, G. and Beatriz, A. (2013). Languages of africa. http://www.languagesgulper.com/eng/Africa.html. Accessed: 2016-01-10.

Allwood, J., Grönqvist, L., and Hendrikse, A. P. (2003). Developing a tagset and tagger for the african languages of south africa with special reference to xhosa. *Southern African Linguistics and Applied Language Studies*, 21:223–237.

Alrabiah, M., Alhelewh, N., Al-Salman, A., and Atwell, E. (2014). An empirical study on the holy quran based on a large classical arabic corpus. *International Journal of Computational Linguistics (IJCL)*, 5(1):1–13.

Anderson, W. N. and Petronella, M. K. (2006). Finite state tokenisation of an orthographical disjunctive agglutinative language: The verbal segment of northern sotho. In *5th Edition of International Conference on Language Resources and Evaluations. Genoa, Italy, May 22–28, 2006*.

Andrew, A. M. (2000). An introduction to support vector machines and other kernel-based learning methods by nello christianini and john shawe-taylor, cambridge university press, cambridge, 2000, xiii+ 189 pp., isbn 0-521-78019-5 (hbk,£ 27.50).

Ani, K. J. (2012). Unesco prediction on the extinction of igbo language in 2025: Analyzing societal violence and new transformative strategies. *Historical Research Letter*, 4:11–20.

Apps, A. (2015). Speak and translate - free live voice and text translator with speech and dictionary. https://itunes.apple.com/gb/app/speak-translate-free-live/id804641004?mt=8.

Artstein, R. and Massimo, P. (2008). *Inter-coder agreement for computational linguistics*. MIT Press, (34)4:555–596.

Arvi, H. (2004). Tagset of swatwol a two-level morphological dictionary of kiswahili. http://www.aakkl.helsinki.fi/cameel/corpus/swatags.pdf. Accessed: 2016-03-22.

Attia, M. A. (2008). *Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation*. PhD thesis, University of Manchester.

Atwell, E. (2008). Development of tag sets for part-of-speech tagging. *An international handbook. Corpus Linguistics: Mouton de Gruyter.*, pages 501–526.

Atwell, E., Al-Sulaiti, L., Al-Osaimi, S., and Abu Shawar, B. (2004). A review of arabic corpus analysis tools. In *in Proceedings of TALN04: XI Conference sur le Traitement Automatique des Langues Naturelles*, pages 229–234. ATALA.

AYOGU, I. I., ADETUNMBI, A. O., and KAMMELU, N. C. (2013). Finite state concatenative morphotactics: The treatment of igbo verbs. *International Journal of Computing and ICT Research*, 7(1).

Bamba Dione, C. M., Kuhn, J., and Zarrieß, S. (2010). Design and development of part-of-speech-tagging resources for wolof (niger-congo, spoken in senegal). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). Valletta, Malta*. European Language Resources Association (ELRA).

Baroni, M. and Kilgarriff, A. (2006). Large linguistically-processed web corpora for multiple languages. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters and Demonstrations. Trento, Italy*, pages 87–90. Association for Computational Linguistics.

Berg, A., Pretorius, R., and Pretorius, L. (2012). Exploring the treatment of selected typological characteristics of tswana in lfg. In *Proceedings of the 17th International Lexical Functional Grammar Conference (LFG 2012)*, pages 85–98. CSLI Publications.

Bigi, B. (2011). A multilingual text normalization approach. In *In 2nd Less-Resourced Languages workshop, 5th Language and Technology Conference, Poznàn (Poland)*, pages 515–526.

Björn, G., Fredrik, O., Atelach, A. A., and Lars, A. (2009). Methods for amharic part-of-speech tagging. In *Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages AfLaT 2009*, pages 104–111. TEHNOGRAFIA DIGITAL PRESS 7 Ektoros Street, Athens, Greece.

Björn, G. and Lars, A. (2009). Experiences with developing language processing tools and corpora for amharic. In *IST-Africa 2010 Conference Proceedings*, pages 1–8. Paul Cunningham and Miriam Cunningham (Eds) IIMC International Information Management Corporation, 2010.

Bosch, S. E., Pretorius, L., , and Fleisch, A. (2008). Experimental bootstrapping of morphological analysers for nguni languages. *Nordic Journal of African Studies.*, 17.

Brants, T. (1999). Tnt– statistical part-of-speech tagger. `http://www.coli.uni-saarland.de/~thorsten/publications/Brants-TR-TnT.pdf`. Accessed: 2016-01-18.

Brants, T. (2000a). Inter-annotator agreement for a german newspaper corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000). Athens, Greece.*

Brants, T. (2000b). Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Brill, E. (1995a). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Comput. Linguist. MIT Press. Cambridge, MA, USA*, 21(4):543–565.

Brill, E. (1995b). Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the third workshop on very large corpora*, volume 30, pages 1–13. Somerset, New Jersey: Association for Computational Linguistics.

Brill, E. and Marcus, M. (1992). Tagging an unfamiliar text with minimal human supervision. In *Proceedings of the Fall Symposium on Probabilistic Approaches to Natural Language . American Association for Arti cial Intelligence (AAAI), 1992*, pages 10–16.

Calzolari, N., Bel, N., Choukri, K., Mariani, J., Monachini, M., Odijk, J., Piperidis, S., Quochi, V., and Soria, C. (2011). Final flarenet deliverable language resources for the future–the future of language resources. *The Strategic Language Resource Agenda. FLaReNet project.*

Childs, G. (2005). *An introduction to African languages.* Amsterdam: John Benjamins Publishing Company.

Clark, M. M. (1990). *The Tonal System of Igbo.* Dordrecht: Foris Publications Holland.

CorpusLinguistics (2016). African language corpora. `https://corplinguistics.wordpress.com/2012/02/08/african-language-corpora/`. Accessed: 2016-01-10.

Crane, P. A. (2006). Texture in text: A discourse analysis of a news article using halliday and hasan's model of cohesion. *Nagoya University of Foreign Studies Foreign Studies Bulletin*, pages 131–156.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics.

Daelemans, W., Zavrel, J., Berck, P., and Gillis, S. (1996). Mbt: A memory-based part of speech tagger-generator. In *Proceedings of the Workshop on Very Large Corpora, Copenhagen, Denmark.*

Daoud, A. M. (2010). Morphological analysis and diacritical arabic text compression. *Computer Journal of the International Journal of ACM Jordan*, 1(1):41–47.

De Pauw, G. and De Schryver, G.-M. (2009). African language technology: the data-driven perspective. In *Proceedings of the Second Colloquium on Lesser Used Languages and Computer Linguistics, Bozen-Bolzano, 13th-14th November 2008*, pages 79 – 96. European Academy.

De Pauwy, G., de Schryverz, G.-M., and de Looy, J. v. (2012). Resource-light bantu part-of-speech tagging. In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages, SALTMIL 8-AFLAT 2012*, pages 85–92. European Language Resources Association (ELRA).

Demuth, K., Faraclas, N., and Marchese, L. (1986). Niger-congo noun class and agreement systems in language acquisition and historical change. In *Proceeding of a Symposium, Eugene, Ore., 1983*, volume 7, page 453. John Benjamins Publishing Co. Amsterdam/Philadelphia.

Economist (2011). Africa rising: After decades of slow growth, africa has a real chance to follow in the footsteps of asia. `http://www.economist.com/node/21541015`. Accessed: 2016-01-10.

Elworthy, D. (1995). Tagset design and inflected languages. In *Proceedings of the ACL SIGDAT Workshop, Dublin, (also available as cmp-lg archive 9504002).*

Emenanjo, N. E. (1978). *Elements of Modern Igbo Grammar: A Descriptive Approach.* Ibadan Oxford University Press.

Factbook, C. W. (2016). Igbo 23%" out of a population of 177 million (2014 estimate). `https://www.cia.gov/library/publications/the-world-factbook/geos/ni.html`. Accessed: 2016-01-09.

Fernández, R. (2011). *Assessing the Reliability of an Annotation Scheme for Indefinites Measuring Inter-annotator Agreement.* Institute for Logic, Language and Computation University of Amsterdam.

Gebre, B. G. (2010). *Part of speech tagging for Amharic.* PhD thesis, University of Wolverhampton Wolverhampton.

Gelu, S. (2010). Pos tagset design for sherpa text. Technical report, Central Department of Linguistics Tribhuvan University, Kathmandu.

Geoffrey, L. (2004). Developing linguistic corpora: a guide to good practice adding linguistic annotation. Geoffrey Leech, Lancaster University. Available from: `https://ota.ox.ac.uk/documents/creating/dlc/chapter2.htm`. Accessed: Feb 16, 2017.

Gertrud, F., Ulrich, H., Elsabé, T., and Danie, P. (2009). Part-of-speech tagging of northern sotho: disambiguating polysemous function words. In *AfLaT '09 Proceedings of the First Workshop on Language Technologies for African Languages*, pages 38–45. Association for Computational Linguistics Stroudsburg, PA, USA.

Giesbrecht, E. and Evert, S. (2009). Is part-of-speech tagging a solved task? an evaluation of pos taggers for the german web as corpus. In *Proceedings of the fifth Web as Corpus workshop, San Sebastian*, pages 27–35.

Girma, A. D. and Mesfin, G. (2006). Manual annotation of amharic news items with part-of- speech tags and its challenges. *Ethiopian Languages Research Center Working Papers*, 2:1–16.

Girma, A. D. and Mesfin, G. (2010). Fast development of basic nlp tools: Towards a lexicon and a pos tagger for kurmanji kurdish. In *International Conference on Lexis and Grammar, Belgrade : Serbia (2010)*, page 0.

Goldsmith, J. (1979). *Autosegmental Phonology*. PhD thesis, M.I.T.

Gordon, R. (2005). *Languages of the World, Fifteenth Edition*. Ethnologue Dallas: SIL International.

Green, A. M. (1977). *Kappa Statistics for Multiple Raters Using Categorical Classifications*, volume 2. Proceedings of the Twenty-Second Annual SAS Users Group International Conference, San, Diego, CA.

Halácsy, P., Kornai, A., and Oravecz, C. (2007). Hunpos: an open source trigram tagger. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 209–212. Association for Computational Linguistics.

Halácsy, P., Kornai, A., Oravecz, C., Vikto, T., and Varga, D. (2006). Using a morphological analyzer in high precision pos tagging of hungarian. In *Proceedings of LREC. ELRA*, page 2245–2248.

Hardie, A. (2003). *The Computational Analysis of Morphosyntactic Categories in Urdu*. PhD thesis, University of Lancaster.

Hardie, A., Lohani, R., Regmi, B., and Yadava, Y. (2005). Categorisation for automated morphosyntactic analysis of nepali: introducing the nelralec tagset (nt-01). Technical report, Nelralec/Bhasha Sanchar Working Paper 2.

Heid, U., Taljard, E., , and Prinsloo, D. J. (2006). Grammar-based tools for the creation of tagging resources for an unresourced language: the case of northern sotho. In *5th Edition of International Conference on Language Resources and Evaluations*.

Helgadóttir, S., Loftsson, H., and Rögnvaldsson, E. (2012). Correcting errors in a new gold standard for tagging icelandic text. In *LREC'14: 2944-2948*.

Hepple, M. (2000). Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 278–277. Association for Computational Linguistics.

Hurskainen, A. (2004). Swahili language manager: A storehouse for developing multiple computational applications. *Nordic Journal of African Studies*, 13(3):363–397.

Hyman, L. M. (2003). Why describe african languages? In *World Congress of African Linguistics 4/Annual Conferences on African Linguistics*.

IgboGuide.org (2016). Igbo grammar. `http://www.igboguide.org/HT-igbogrammar.htm`. Accessed: 2016-01-10.

Ikegwuonu, C. N. (2011). Tense as an element of infl phrase in igbo. *Journal of Igbo Language and Linguistics (JILL)*, 3:112–121.

Ikekeonwu, C. (1999). *"Igbo", Handbook of the International Phonetic Association*. Cambridge University Press.

InternetWorldStats (2015). usage and population statistics: Africa. `http://www.internetworldstats.com/africa.htm`. Accessed: 2016-01-10.

Jurafsky, D. and Martin, J. H. (2007). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall.

Jurafsky, D. and Martin, J. H. (2014). Part of speech tagging. Speech and language processing, draft of February 19, 2015. `https://web.stanford.edu/~jurafsky/slp3/9.pdf`.

Kaneda, T. and Bietsch, K. (2016). 2013 world population data sheet" (pdf). www.prb.org. population reference bureau. retrieved 3 february 2017.

Karlsson, F. (1995). Designing a parser for unrestricted text. *In F. Karlsson, A. Voutilainen, J. Heikkilä, and A. Anttila, eds., Constraint Grammar — A Language-Independent System for Parsing Unrestricted Text*, pages 1–40.

Krauwer, S. (2003). The basic language resource kit (blark) as the first milestone for the language resources roadmap. *Proceedings of SPECOM 2003*, pages 8–15.

Krippendorff, K. (1980). *Content Analysis: An Introduction to Its Methodology*. Sage, Beverly Hills, CA.

Kumar, D. and Josan, G. S. (2012). Developing a tagset for machine learning based pos tagging in punjabi. *International Journal of Applied Research on Information Technology and Computing*, 3:132–143.

Kupiec, J. (1992). Robust part-of-speech tagging using a hidden markov model. *Computer Speech and Language*, 6(3):225–242.

Landis, R. J. and Koch, G. G. (1977). *The measurement of observer agreement for categorical data*. biometrics, 159–174, JSTOR.

Leech, G. (1997). *Introducing Corpus Annotation*. Addison Wesley Longman, London.

Leech, G., Garside, R., and Atwell, E. S. (1983). The automatic grammatical tagging of the lob corpus. *ICAME Journal: International Computer Archive of Modern and Medieval English Journal*, 7:13–33.

Leech, G. and Wilson, A. (1999). Standards for tagsets. In *Syntactic wordclass tagging*, pages 55–80. Springer.

Lewis, M. P., Gary, F. S., and Charles, D. F. (2015). Ethnologue: Languages of the world, eighteenth edition. dallas, texas: Sil international. `http://www.ethnologue.com`. Accessed: 2016-01-09.

Lezius, W. (2000). Morphy-german morphology, part-of-speech tagging and applications. In *Proceedings of the Ninth EURALEX International Congress, EURALEX 2000: Stuttgart, Germany, August 8th-12th, 2000*, pages 619–623.

Light, M. (1996). Morphological cues for lexical semantics. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 25–31. Association for Computational Linguistics.

Loftsson, H. (2007). *Tagging and Parsing Icelandic Text*. PhD thesis, University of Sheffield.

Loftsson, H. (2009). Correcting a pos-tagged corpus using three complementary methods. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–531. Association for Computational Linguistics.

Louwrens, L. J. and Poulos, G. (2006). The status of the word in selected conventional writing systems—the case of disjunctive writing. *Southern African Linguistics and Applied Language Studies. Taylor and Francis*, 24(3):389–401.

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *In Alexander Gelbukh (ed.), Computational Linguistics and Intelligent Text Processing, 12th International Conference, CICLing 2011, Proceedings, Part I.*, pages 171–189. Springer.

Mariya, K. (2012). *Towards Adaptation of NLP Tools for Closely-Related Bantu Languages: Building a Part-of-Speech Tagger for Zulu*. PhD thesis, Saarland University.

Milne, R. (1986). Resolving lexical ambiguity in a deterministic parser. *Computational Linguistics. MIT Press*, 12(1):1–12.

Murphy, K. P. (2006). Naive bayes classifiers. *University of British Columbia*.

Nelralec (2006). A part-of-speech tagger for nepali. http://www.lancaster.ac.uk/staff/hardiea/nepali/postag.php#tagset. Accessed: 2016-01-10.

Ngai, G. and Florian, R. (2001). Transformation-based learning in the fast lane. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Nwachukwu, P. (1987). The argument structure of igbo verbs, lexican project working papers in linguistics. Technical report, Massachusetts Institute of Technology, Cambridge Mass. U.S.A.

Nweke, J. A. (2011). A review of the impact of the minimalist programme on igbo noun phrase. Knowledge Review Volume 23. globalacademicgroup.com.

Omniglot (2016). The online encyclopedia of writing systems and languages: Igbo. `http://www.omniglot.com/writing/igbo.htm`. Accessed: 2016-01-10.

Oraka, L. N. (1983). *The foundations of Igbo studies*. Onitsha: University Publishing Company.

Pavel, K. and Karel, O. (2002). (semi-) automatic detection of errors in pos-tagged corpora. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1. Taipei, Taiwan*, pages 1–7. Association for Computational Linguistics.

Petrov, S., Das, D., and McDonald, R. (2011). A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Poulos, G. and Louwrens, L. (1994). *A Linguistic Analysis of Northern Sotho*. Pretoria: Via Afrika Limited.

Pretorius, R., Berg, A., Pretorius, L., and Viljoen, B. (2009). Setswana tokenisation and computational verb morphology: Facing the challenge of a disjunctive orthography. In *In AfLaT2009. Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages*, pages 66–73. European Association for Computer Linguistics. Athens, Greece.

Pritchett, F. W. (2014). A history of the igbo language compiled by frances w. pritchett. `http://www.columbia.edu/itc/mealac/pritchett/00fwp/igbo/igbohistory.html`. Accessed: 2016-01-10.

Pustejovsky, J. and Stubbs, A. (2012). *Natural Language Annotation for Machine Learning*. O'Reilly Media, Inc. Sebastopol, CA 95472.

Ratnaparkhi, A. et al. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142. Philadelphia, USA.

Resnik, P. (1999). Mining the web for bilingual text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. June 20-26, 1999, College Park, Maryland*, pages 527–534. Association for Computational Linguistics.

Resnik, P., Olsen, M., and Diab, M. (1999). The bible as a parallel corpus: Annotating the 'book of 2000 tongues. *Computers and the Humanities. Springer*, 33:29–153.

Robin (2009). what is corpus? `http://language.worldofcomputing.net/linguistics/introduction/what-is-corpus.html`. Accessed: 2016-01-10.

Rowbory, J. A. (2009). The history and impact of igbo bible, 1840-1920. `http://negstor.rowbory.co.uk/wp-content/uploads/2009/03/the-history-and-impact-of-the-igbo-bible-1840-1920.pdf`. Accessed: 2017-01-28.

Samuelsson, C. (1993). Morphological tagging based entirely on bayesian inference. In *9th Nordic conference on computational linguistics. NODALIDA-93, Stockholm University, Stockholm, Sweden*, pages 225–238.

Sang, Y. L. (2005). Sherpa orthography. Technical report, Korea Research Institute for Languages and Culture.

Sapna, K., Ravishankar, M., and Sanjeev, K. S. (2011). Pos tagging of punjabi language using hidden markov model. *An International Journal of Engineering Sciences*, 2.

Sawalha, M. and Atwell, E. (2009). Linguistically informed and corpus informed morphological analysis of arabic. In *Proceedings of the 5th Corpus Linguistics Conference*. Lancaster University Centre for Computer Corpus Research on Language.

Sawalha, M. and Atwell, E. (2010). Fine-grain morphological analyzer and part-of-speech tagger for arabic text. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1258–1265. European Language Resources Association (ELRA).

Scannell, K. P. (2007). The crúbadán project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, volume 4, pages 5–15. Presses univ. de Louvain.

Schmidt, R. (1999). *Urdu: an essential grammar.* London: Routledge.

Sisay, F. A. (2005). Part of speech tagging for amharic using conditional random fields. In *Workshop on Computational Approaches to Semitic Languages.*, pages 47–54. ACL (2005).

Sornlertlamvanich, V., Takahashi, N., and Isahara, H. (1999). Building a thai part-of-speech tagged corpus (orchid). In *J Acoust Soc Japan*.

Spiegler, S., van der, S. A., and Flach, P. A. (2010). Additional material for the ukwabelana zulu corpus. Technical report, Intelligent Systems Group University of Bristol.

Tachbelie, M. Y., Abate, S. T., and Besacier, L. (2011). Part-of-speech tagging for under-resourced and morphologically rich languages — the case of amharic. *HLTD*, pages 50–55.

Taljard, E., Faaß, G., Heid, U., and Prinsloo, D. J. (2008). On the development of a tagset for northern sotho with special reference to the issue of standardisation. *Literator: Journal of Literary Criticism, Comparative Linguistics and Literary Studies. AOSIS*, 29(1):111–137.

Tapas, K. and Philip, R. (1999). The bible, truth, and multilingual ocr evaluation. In *in Proc. of SPIE Conf. on Document Recognition and Retrieval*, pages 86–96.

Thede, S. M. and Harper, M. (1997). Analysis of unknown lexical items using morphological and syntactic information with the timit corpus. In *In Proceedings of the Fifth Workshop on Very Large Corpora. Beijing, China*, pages 261–272.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. May 27-June 01, 2003, Edmonton, Canada*, pages 173–180. Association for Computational Linguistics.

Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13. October 07-08, 2000, Hong Kong*, pages 63–70. Association for Computational Linguistics.

Trushkina, J. (2006). The north-west university bible corpus: a multilingual parallel corpus for south african languages. *Language Matters: Studies in the Languages of Southern Africa. UNISA Press*, 37(2):227–245.

Tseng, H., Jurafsky, D., and Manning, C. (2005). Morphological features help pos tagging of unknown words across language varieties. In *Proceedings of the fourth SIGHAN workshop on Chinese language processing*, pages 32–39.

Uchechukwu, C. (2005). The representation of igbo with the appropriate keyboard. In Ikekeonwu, C. and Nwadike, I., editors, *Igbo Language Development: The Metalanguage Perspective*, pages 26–38. CIDJAP Enugu.

Uchechukwu, C. (2006). Igbo language and computer linguistics: Problems and prospects. In *Proceedings of the Lesser Used Languages and Computer Linguistics Conference*. European Academy (EURAC).

Uchechukwu, C. (2008). African language data processing: The example of the igbo language. In *10th International pragmatics conference, Data processing in African languages*.

UCLA (2014). Language materials project: Igbo. `http://www.lmp.ucla.edu/Profile.aspx?menu=004&LangID=13`. Accessed: 2016-01-10.

Van Valin, R. D. (2001). *An Introduction to Syntax*. Syndicate of the University of Cambridge.

Welmers, W. and Welmers, B. (1968). Igbo: A learner's manual, privately published by the author. *University of California, Los Angeles*.

Widjaja, M. (2013). Igbo grammar. http://www.igboguide.org/HT-igbogrammar.htm.

Williamson, K. (1971). Igbo dictionaries. Paper presented at the Seminar on the problems of the Igbo Language and Literature. University of Nigeria, Nsukka.

Wynne, M., Arts, and Service, H. D. (2005). *Developing linguistic corpora: A guide to good practice*, volume 92. Oxbow Books Oxford.

Ọnwụ Committee (1961). The official igbo orthography. http://www.columbia.edu/itc/mealac/pritchett/00fwp/igbo/txt_onwu_1961.pdf.

# Appendix A

# Full Description of Igbo Tagset And Taggers Performance Scores

## A.1 The Developmental Stages of Igbo Tagset

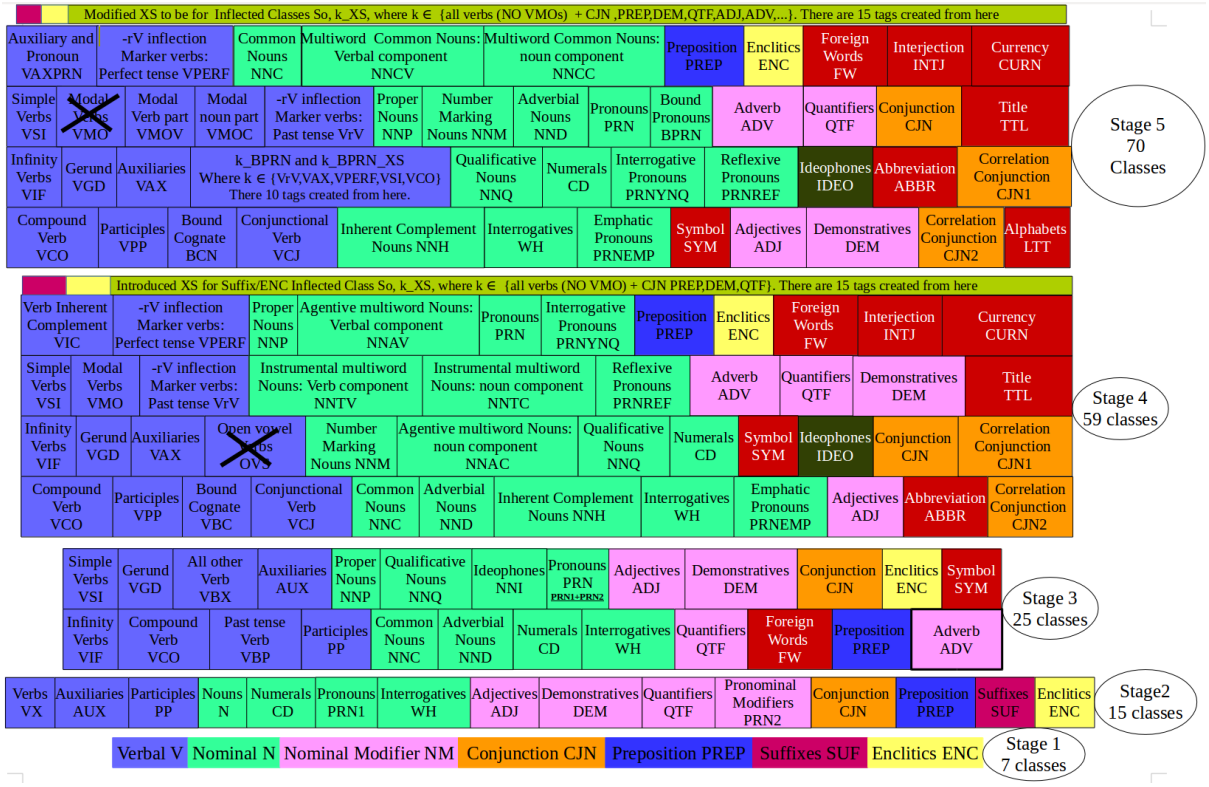



Figure A.1: The developmental stages of Igbo tagset. Red buttons indicate new tags added that are independent of other the core tags. Other colours show the decomposition steps of the core tags.

### A.1.1 The Extensional Affixes Part of Igbo Tagset

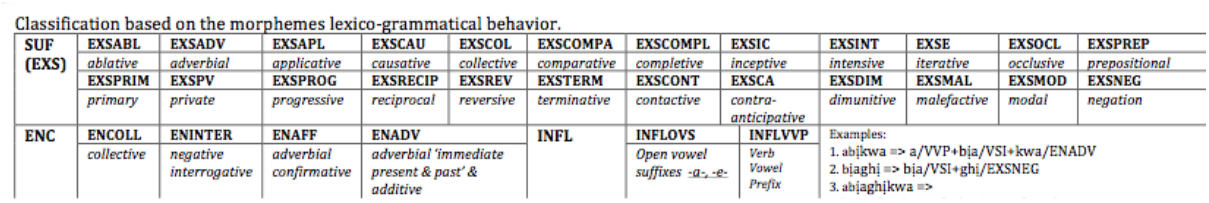



Figure A.2: The developmental stages of Igbo tagset: decomposing of the extensional suffix marker *XS* into various morph-tags according to grammatical functions.

## A.2 Igbo Tagset (IgbTS) Descriptions

This section describes the applications of the tagset schema on a prepared corpus data. It also define tags and the relationship that exist between them.

| **Noun Class: 8** | | | |
|---|---|---|---|
| **Tag** | **Description and Example** | | **Illustration** |
| NNP | Noun Proper | *Chineke* "God", Onyeka, Okonkwo, Osita, Izrel | (1) Na mmalite, **Chineke** kele eluigwe na ụwa.<br><br>"In [the] beginning , 'God' created heaven and earth." |
| | | All the words that begin with a capital letter that are within a sentence. | *. . . mekwaa ka ọ na-acha ọcha n'ime ọbara* ***Nwa Atụrụ*** *ahụ.* ". . . made them white in the blood of the **Lamb**" |
| NNC | Noun Common | *ọkụ* 'fire', *ụwa* "earth", *osisi* "tree, stick", *ala* "ground", *eluigwe* "sky, heaven" | (2) Na mmalite ,Chineke kele **eluigwe** na **ụwa**.<br><br>"In [the] beginning , God created 'heaven' and 'earth'." |
| **Special Noun Class** | | | |
| NNM | Number Marking Noun | Igbo nouns are not inflected in marking singular and plural.<br>Rather, there are words that when preceding a noun modify it to singular or plural. e.g. *ndị* 'people' (pl.), *nwa* 'child' (sg.), *ụmụ* 'children' (pl.).<br>*ndị* is classified as a common noun with an attached phrase of "thing/person associated with" Emenanjo (1978)<br>*ndị* before a noun (NNP or NNC) without any intervening word marks plurality of that noun.<br>*nwa* 'sg.' marks nouns for singular.<br>Clarity between marking NNC and NNM based on sentential meaning. | (3) **Ndị** British. " 'People' British", that is, "The people of Britain."<br><br>(4)***Ndị*** *obodo m.* " 'people' town me", i.e. "The people of my town."<br><br>(5) ***ndị Farisii*** *jụrụ ya.* "people Pharisees asked him", i.e. "The Pharisees asked him."<br><br>"nwa" can be marked NNC also.<br>(6) *Ọ mụrụ* ***nwa*** *nwoke.* "He begot 'a' son" or ***nwa*** *nwoke* "man"<br>(7) *...kama ọ na-anọru* ***nwa*** *oge* "...but continues for 'a' time"<br>(8) *Nwa Devid* "child Devid", i.e. "Devid's child or son of Devid"<br>In (6) and (7) "nwa" is acting as a singular modifier of nouns "nwoke" and "oge". While (8)'s "nwa" is a common noun NNC. |
| | | *ụmụ* 'pl.' associates with a thing/person to indicate plurality. It can be marked as NNC when it is not acting as a plural modifier. "ụmụ" in (9) and (10) are NNM while (11) and (12) are NNC. | (9) *Unu* ***ụmụ*** *ajụala* "You offspring of vipers." (10) *ụmụ nwoke* "men". (11) *Chineke nwere ike ime ka Ebraham nweta* ***ụmụ*** *site na nkume ndị a* "God is able to raise up 'children' to Abraham from these stones." (12) **ụmụ** ya "his/her 'children' " or **ụmụ** Pauḷina "Pauḷina's 'children' ". |

| | | | |
|---|---|---|---|
| NNQ | Qualificative noun | These are nouns that are inherently semantically descriptive. They have been frequently called adjectives but don't have full properties of an adjective, e.g. *ogologo* [height, long, tall]. To identify NNQ: (1) They can only be used after verb **-dị**. (2) May be head of Noun phrase (NP), i.e. found in a noun slot in NP. (3) Often used together with other nouns as their qualifiers(esp. proper, common and numerals). | (12) *Ọ dị* ***ogologo***. "It be long" → "It is long." (13) ***Ogologo*** *osisi* "long tree" or "long stick/tall tree." (14) *Ọ dị* ***obosara***. "It is wide." See appendix .1 item number 8 for more illustrations. |
| NND | Adverbial noun | Words in these lexical class always function to modify verbals. They may be used immediately after **-bụ, -ji and -dị**, never found as indirect object and frequently used as Head of NP (Emenanjo 1978). They may be found in the noun slots of NP, adverbial slots or elsewhere in the sentence. | (15) *Ọ dị* ***nwayọọ*** "He is quiet" (16) *Ọ ji* ***nwayọọ*** *eri nri ya* "He holds slowly eats food his.", that is, "He eats his food slowly." See appendix .1 item number 9 for more examples. |
| NNH | Inherent Complement | The typical Igbo verb has a [verb + NP/PP] structure. The NP/PP serves as the complement to the verb and always cooccurrs with the verb to complete its sense, even though at times quite distant from the verb. | (17) *ịgụ* ***egwu*** "to sing song" → "to sing" (18) *igwu* ***ji*** "to dig yam" → "harvest yam" (19) *iti* ***ịgba*** "to beat drum" → "to drum" (20) *ịkọwa* ***okwu*** "to explain word" → "interprete..." **Note:** NNH can occur without complementing a verb, in that case it should be marked as NNC. See second appendix figure A.3 and first appendix item number 10 for more illustrations. |
| NNCV and NNCC (2 tags. See note.) | Multword nouns | Multiword nouns formed by verb nominalization. Verbal (V) and inherent (C) components marked with complementary tags NNCV and NNCC. Compare (17) with (21) and (19) with (22). | (21) *ọgụ/**NNCV** egwu/**NNCC*** "singer" (22) *oti/**NNCV** ịgba*/**NNCC** "drummer". (23) *ntachi/**NNCV** obi/**NNCC*** "persistence or determination". (24) *ntabi/**NNCV** anya/**NNCC*** "envy". See appendix .1 item number 11 for more examples |
| **NOTE:** We introduced link indicators V and C in NNC. Where V and C stand for verbal and Complementary respectively. So, NNCV indicates derivation from the verbal component of the inherent complement verb and NNCC is the inherent complement . See examples 21 to 24. | | | |
| **Verb Class: 10** | | | |
| **NOTE:** Igbo verb is made of three mutually obligatory and complementary parts diagrammed thus **V±CP±BCN**, where V is verb, CP is complementary and BCN is Bound Cognate Noun. Igbo verbs co-exits with either one or both parts (Emenanjo 1978). | | | |
| VIF | Infinitive | Marked through the addition of the vowel [i] or [ị] to the verb root. This is governed by the vowel harmony rule whereby the inifinitive marker must come from the vowel group to which verb's Root Vowel belongs. The two vowel groups are [aịọụ] and [eiou]. | (1) *ri* 'eat' → [i + ri ] → *iri* 'to eat' (2) *nye* 'give' → [i + nye ] → *inye* 'to give'. ( 3) *pụ* 'go out' → [ị + pụ] → *ịpụ* 'to go out' (4) *ba* "enter" → [ị + ba ] → *ịba* 'to enter' (5) *tụ* 'throw' → [ị + tụ] → *ịtụ* 'to throw' |
| VSI | Simple verb | Has only one verb root. | *ri* 'eat', *nye* 'give', *ba* 'enter', *tụ* 'throw' |
| VCO | Compound Verb | Involves a combination of two verb roots. | (6) *ịtụ* 'to throw' + *ịba* 'to enter' → *ịtụba* 'to throw in' (7) *ịba* 'to enter' + *inye* 'to give' → *ịbanye* 'to enter into' |
| VMO (2 tags. See note.) | Modal Verb | Structurally this is made up of inherent complement verbs and simple verbs. In addition, the few modal verbs of the language are supplemented by modal suffixes. [See the section on suffixes] | (10) *ikwesi* 'to should' → 'should'. (11) *inwe ike* "to have strength" → 'can'. Note that the word 'ike' also functions as a common noun in other context. See appendix .1 item number 7. |
| **NOTE:** We introduced link indicators V and C to denote simple verb and inherent complement components of the modal. So, we have VMOV and VMOC (used only where there is inherent complement). For example, **inwe/VMOV ike/VMOC**. In the case of only modal verbs suplemented by modal suffixes we use VMO. For example, **ikwesi/VMO** | | | |
| VAX | Auxiliary Verb | ga [Future marking], na [progressive] | (12) *Obi ga - eri nri* "Obi AUX(Fut.) eat food." → "Obi shall eat." (13) *Obi ga - atụ egwu* "Obi AUX(Fut.) throw fear" → "Obi shall be afraid" (14) *Obi na - eri nri* "Obi AUX(Prog.) eat food." → "Obi is eating." (15) *Obi na - atụ egwu* "Obi AUX(Prog.) throw fear" → "Obi is afraid." |

| | | | |
|---|---|---|---|
| VPP | Participle | Always occurs after the auxiliary, and is formed through the additon of the harmonizing vowel **e/a** to the verb root. Note the structures after the auxiliaries in sentences (12) to (15). Moves from **Verb → Participle** | (16) *ri* 'eat' → *eri* 'eat' (17) *nye* 'give' → *enye* 'give' (18) *ba* 'enter' → *aba* 'enter' (19) *tụ* 'throw' → *atụ* 'throw' |
| VCJ | Conjunctional Verb | A verb that has a conjuntional meaning, especially in narratives. | (20) *Obi banye-re n' ụlọ* ***wee*** *hụ nne ya.* "Obi enter-rv(PAST) PREP house and then saw mother his" → "Obi entered the house and saw his mother." |
| BCN | Bound Cognate Noun | Formed through the addition of the harmonizing suffix **a/e** to the verb root. Looks like the participle but always occurs after the participle in the same sentence as the verb from which it is formed. Can be formed from every verb. | ( 21) *Obi ga - eri nri ahụ eri* "Obi AUX(Fut.) eat food DET BCN" → "Obi will surely eat that food." |
| VGD | Gerund | Formed through a form of reduplication of the verb root with the addition of a harmonizing vowel o/ọ. Internal vowel changes can also occur as in example (23) and (24) | (22) *ri* 'eat' [o + ri +ri ] → *oriri* 'eating' (23) *nye* 'give' [ o + nye + nye] → *onyinye* 'giving' (24) *ba* 'enter' [ọ + bụ + ba ] → *ọbụba* 'entering' (25) *tụ* 'throw' [ọ + tụ +tụ] → *ọtụtụ* 'throwing' |
| **Other part-of-speech tags: 23** | | | |
| ADJ | Adjective | To be understood as the traditional part of speech 'adjective' this qualifies a noun. They modify the meaning of nomonal they co-occur and can never be used after the verb "dị". There are five of them *ajọ or ọjọọ, ọma, ọcha, ojii, ukwu or ukwuu*, four of which divide up neatly into two pairs of antonyms (Emenanjo 1978). | Their syntactic features are 1) Except *ajọ* which comes before the nominals they modifies, others *ọjọọ, ọma, ọcha, ojii and ukwu* comes after the nominals they modifies. Examples 1) *mmụọ* ***ọjọọ*** "spirit bad", that is, "bad spirit". 2) ***ajọ mmụọ*** "bad spirit". 3) *onye* ***ukwu*** "big person" 4) *nke* ***ọma*** associated with "something good" 5) *nwoke* ***ọcha*** "fair man" 6) *uwe* ***ojii*** "black cloth". **They normally follow or precedes their nominals without any intercepting words.** |
| PRN | Pronoun | The three persons realised as: First person pronoun (sing + pl), Second person Pronoun (sing + pl), Third person Pronoun (sing + pl) | |
| PRNREF | Reflexive Pronoun | This involves the combination of the personal pronouns with the noun onwe 'self'. The combination is fixed but not written together like the English reflexive. | (2) First person reflexive pronoun: Singular: *onwe m* Plural: *onwe anyị* (3) Second person reflexive pronoun Singular: *onwe gị* Plural: *onwe unu* (4) Third person reflexive pronoun Singular: *onwe ya* Plural *onwe ha*. **Whether to tag the structure [onwe + pronoun] together as the reflexive pronoun, regardless of the orthographic space between them?** |
| PRNEMP | Emphatic pronoun | This involves the structure [pronoun+onwe+pronoun] and is to be handled together like the reflexive pronoun | (5) First person emphatic pronoun: Singular: *mụ onwe m* Plural: *anyị onwe anyị* (6) Second person reflexive pronoun Singular: *gị onwe gị* Plural: *unu onwe unu* (7) Third person reflexive pronoun Singular: *ya onwe ya* Plural: *ha onwe ha* |
| PRNYNQ | Pronoun question. | Questions that return YES or NO answer. It begins with a pronoun marked with low tone and terminates with a question mark '**?**' | E.g. *m̀, à, hà, ò, ọ̀, ùnu, ...* See appendix 1 item number 6 for examples. |
| BPRN | Bound pronouns | Any prounoun tied to the vowel prefixes *a,e* attached to a verb. In the examples 1 through 4, *a,e* prefixes in words *ana, ekwuru, enyewo or enyego* are bounded to pronouns *m, ha.* | E.g. 1) ***Ana/VAX_BPRN m/BPRN*** *akwado ka e gbuo ha.* "I am getting ready to kill them." 2) ***Ekwuru/VrV_BPRN m/BPRN*** *okwu banyere ya.* "I talked about him/her." 3) *Lee,* ***enyewo/VPERF_BPRN m/BPRN*** *unu ike.* or *Lee,* ***enyego/VPERF_BPRN m/BPRN*** *unu ike.* "Look, I have give you (people) power" 4) ***Ha/BPRN ana/VAX_BPRN****-ada n'ugwu gilboa.* "They, bowing down in the mountain of gilboa". See second appendix figure A.4 for more illustrations. |

| | | | |
|---|---|---|---|
| ADV | Adverb | Though there are few of them in Igbo, they should be tagged with the typical abbreviation ADV | *nnọọ* 'just' (8) *Enyi m nwoke, I kwu-te-re ya nnọọ* "Friend me man you speak-DIR-rv(PAST) it just" → "My friend, you just spoke the right thing!" |
| CJN (3 tags. See note) | Conjunction | The most unproblematic, except for complex ones. Morphologically, we could distinguish between complex and simple conjunctions, while in line with there grammatical functions one could distinguish between co-rodinators, sub-ordinators and correlatives. | Co-ordinator: *na* 'and' (9) Emeka *na* Mary... "Emeka and Mary..." Sub-ordinator: *mgbe* 'when' (10) *Emeka na-eri nri mgbe Mary batara* "Emeka AUX-eat food CONJ Mary enter- rV(PAST)" → "Emeka was eating when Mary entered." Correlative: *ma ..... ma .....* "both .... and..." (11) *Ma ndị nwoke ma ndị nwaanyị....* "CONJ PL man CONJ PL woman" → "Both the men and the women ..."' |
| **Note:** We introduced link indicators CJN1 and CJN2 for "correlative CJN", where CJN1 is the first CJN and CJN2 is the second. For example, *ma/CJN1... ma/CJN2*. Note: CJN1 and CJN2 must occur together. They might occur at a close or far distance to each other in a sentence. | | | |
| PREP | Preposition | There are few of them in Igbo. The preposition *na* is realised as *n'* if the modified word begins with a vowel, as in example (21). Also there are other words that have grammatical function as preposition *site, banyere,ruo, ...* | *na* 'in/at/by' (12) Okey *nọ na be nna ya.* "Okey be PREP place father his." → "Okey is in his father's place" (13) *Okey bi n' ụlọ nna ya.* "Okey live PREP house father his." → "Okey lives in his father's house." |
| QTF | Quantifiers | This can be found either after or before their nominals in the NP structure to express or measure the quantity of the nominal. | *dum* 'all', *niile*, 'all', *ọtọtọ* 'many', *naanị* 'only', *naabọ*, 'only two' |
| DEM | Demonstratives | This is made up of only two deictics and always used after their nominals. | *a* 'this', *ahụ* 'that' |
| INTJ | Interjection | | *Ee chei!* |
| FW | Borrowed words | Foreign words found in Igbo texts. | amen |
| SYM | Punctuations | It includes all symbols. | |
| CD | Numbers | This includes all digits 1,2,3, ... and *otu, mbụ, abụa, atọ, ...* | |
| WH | Interrogatives | Questions that return useful data through explanation. | *Ònye, gịnị, olee, ...* |
| IDEO | Ideophones | This is used for sound-symbolic realization of various lexico-grammatical function. It can be variously realised as ideophones of colour, taste, sound, etc. | E.g. *nịganịga, mụrịị, kọị, etc.* |
| LTT | alphabets | All (both) graphemes that represent a character in Igbo, which occur in the text. | *gb, gw, kp, nw, ...* |
| TTL | TITLE | Includes foreign and Igbo titles. | E.g. *Maazị, Mz. , Ma. .* |
| CURN | CURRENCY | | *Naịra, dọla* |
| ABBR | ABBREVIATION | | OAU, ISSN, etc. |
| **NOTE:** 'inflection' and other additional morpho-syntactic-cum-semantic modification of or additions to the verb (e.g. 'tense' and 'aspect') are through suffixes called 'inflectional suffixes'. For this reason, 'inflectional suffixes' shall form another tagset group, with the grammatical attributes as the sub-types. | | | |
| **Inflectional Class: 2** This is usually attached to the verb to express various forms of temporal relations of an event as either presently happening, already happened or still to happen. | | | |
| VrV | rV | If attached to an active verb, it expresses the simple past; but expressive a stative meaning when attached to a stative verb. It can also vary according to dialect and as such can be realised as lV. The 'V' stands generally for 'any vowel' attached to the root consonant. NOTE: VrV is to mark both active and stative verbs where $-rv$ occurred. | **Active Verb:** *ịgba egwu* 'to dance' (1) *Emeka gba-ra egwu* "Emeka dance-rV(PAST) dance" → "Emeka danced." (2) *Emeka gba-lụ egwu* "Emeka dance-lV(PAST) dance" "Emeka danced." **Stative Verb:** *ịma mma* "to be beautiful" (3) *Ada ma-ra mma* "Ada beauty-rV(STATIVE) beautiful" [literal: 'Ada beauties beautiful'] → "Ada is beautiful." (4) *Ada ma-lụ mma* "Ada beauty-rV(PAST) beautiful" [literal: 'Ada beauties beautiful'] "Ada is beautiful." |

| | | | |
|---|---|---|---|
| VPERF | Perfect | Generally described as the form of the 'perfect tense'. While the form *=la/=le* obeys the vowel harmony rule, the variant *=go* does not obey the harmony rule. NOTE: VPERF is to mark all *–la/–le, –go* verbs where *–rv* occurred. | **Verbs:** (a) *ịgba egwu* 'to dance' (b) *iri nri* 'to eat (food)' (5) *Emeka a-gbaa-la egwu* "Emeka PREF-dance-PERF dance" → "Emeka has danced." (6) *Emeka e-rie-le nri* "Emeka PREF-eat-PERF food" → "Emeka has eaten." Compare the above with the two below where the same verbs are involved but without any changes in the suffix: (7)*Emeka a-gba-go egwu* "Emeka PREF-dance-PERF dance" → "Emeka has danced." (8)*Emeka e-ri-go nri* "Emeka PREF-eat-PERF food" → "Emeka has eaten." |
| **Enclitics:** This class includes all enclitics found not attached to any token. The enclitics co-occur with verbs and other parts of speech. For their orthographic realization, the rule is to write them separately when they co-occur with other parts of speech, but write them together with the verb when they co-occur with a verb. Note the pattern of realization in the examples below. Above all, this needs to be borne in mind because even some of the Igbo authors do not observe this rule and it then creates the problem of how to establish what the actual word form is. 1 tag. | | | |
| ENC | **Collective**. | *cha, sị nụ, kọ* – means all, totality forming a whole or aggregate. | E.g. *Ndị a cha_ENC bịa-ra oriri* instead of *Ndị a bịa-cha_ENC-ra oriri.* "People DET all come-rV(Past) feasting" → "All these people came to the feasting" The second example will be assigned _XS as any type of suffix |
| | **Negative Interrogative**. | *dị, rị, dụ* – indicates scorn or disrespect and are mainly used in Rhetorical Interrogatives. | |
| | **Adverbial 'Immediate present and past'**. | *fọ/hụ* – it indicates action that is just/has just taking/taken place. *rịị* – indicates that an action/event has long taken place | |
| | **Adverbial 'Additive'**. | *kwa (kwọ), kwu* – mean 'also', 'in addition to', 'denoting', 'repetition or emphasis'. | |
| | **Adverbial 'Confirmative'**. | *nọọ (nọọ; nnọọ)* – this means really or quite. | |
| **Special Tags: 26** | | | |
| *α*_XS (15 tags) | any POS tag with affixes. | for *α* ∈ {VIF, VSI, VCO, VPP, VGD, VAX, CJN, WH, VPERF, VrV, PREP, DEM, QTF, ADJ,ADV}. See verb, other POS, inflectional classes. Affixes includes Verbal Vowel Prefixes, Open Vowel Suffixes, Enclitics when attached to any tokens especially verbs, and suffixes [SUF] | E.g. ịkpagharịsi, bịakwasịkwa, bịaghịkwa, abọchabeghị, oririkwa, erighịkwa, rie, pụọ,... Please refer to XS class for all affixes. |
| *α*_BPRN *α*_BPRN_XS (10 tags) | Vowel prefix *a,e* of a verb bound to pronouns | any verb whose prefix *a,e* is bound to the pronoun it is preceding or following. for *α* ∈ {VrV, VAX, VCO, VPERF, VSI}. In the examples 1 through 4, *a,e* prefixes in words *ana, ekwuru, enyewo or enyego* are bounded to pronouns *m, ha.* **Note**, if you rephrase the sentences, they will still be grammatical correct conveying the same sense. E.g., *M si Sheffield abị.* So, verbs *enyewo, ekwuru* in 2 and 3 are not VPP (participles). | E.g. 1) *Esi/**VIS_BPRN** m Sheffield abịa.* "I am coming from Sheffield." 2) *Ekwuru/**VrV_BPRN** m okwu banyere ya.* "I talked about him/her." 3) *Lee, enyewo/VPERF_BPRN m unu ike.* or *Lee, enyego/**VPERF_BPRN** m unu ike.* "Look, I have give you (people) power". In 1, 2 and 3 examples the prefixes *a* and *e* are bound to pronoun *m*. That's why you can rewrite any say 1 as *M/**PRN** na-/**VAX** akwado ka e gbuo ha.* "I am getting ready to kill them." See second appendix figure A.4 for more illustrations. |
| VAXPRN | Auxilliary and pronoun | Auxilliary with dependent pronoun for subject and non attributable subject | "ịna-" and "ana-" |
| **XS class** In Emenanjo (1978), "the term 'extensional' is used in African linguistics for referring to elements, usually affixes, which function principally as meaning-modifiers, i.e. extending the meaning of the word with which they are used." Arguably, enclitics would be regarded as part of extensional suffixes based on the above assertion. Suffixes and enclitics found so far: ba , be , bo , bu , bụ , bụ , cha , chi , chu , chị , de , debe , di , dide , do , dị , dụ , fụ , ga , gba , gbado , gbe , ge , gha , gharị , gheri , ghi , ghị , gide , go , godu , gwa , gwo , gọ , ha , haa , he , hu , hube , huwe , hịa , hụ , hụka , karị , kata , kebe , keli , kene , keta , kiri , kpọ , kpụ , kwa , kwasị , kwu , kwọ , kọ , kụ , la , lahụ , le , leru , li , lụ , nahị , nahụ , nalụ , nanị , narị , nnọọ , nu , nwu , nwụ , nya , nye , nọọ, nụ , pịa , pọ , pụ , ra , re , ri , riri , ro , ru , rị , rịrị , rịta , rịị , rọ , rụ , sa , si , sie , sisi , sị, sịa , sịsị , ta , te , tu , tọ , tụ , vọ , wa , we , wo , za , ze , zi , zu , zị , zọ. If found attached to any of verb or other word, modifies its original meaning and hence, the tags *XS* in this section. Are the followings valid standard Igbo suffixes **wo, ro,pụ, zị,zi, re,ru,chị, zu, sịa, gha, nahụ, sie, rọ**? | | | |

Table A.1: POS tags description and usage

## A.2.1 More illustrative examples of some complex words and POS tags

1. Agbawo m **ọsọ** ahụ ...
   Here **ọsọ** is the inherent complement to the inflected verb "agbawo", which in standard Igbo should be "agbago". Its verb root is**gba ọsọ** but is prefixed 'a' because of the position of the 'm' pronoun in the sentence. Therefore, it should tag 'VSI_BPRN_XS' to indicate pronoun bound 'BPRN' and inflection 'XS'. **ọsọ** will be tag 'NNH'.

2. E jirikwa m **ọsọ** gbaga n'ụzọ.
   Here it functions as part of a serial verb construction. It is simply NNC. The verb "gbaga" is made up of the two verbs "gba" and "ga" which is a compound verb structure. The relationship between "gba" and **ọsọ** is that of a collocation. You can also say the following: "E jirikwa m **ngwangwa** gbaga n'ụzọ." Here you can see that you can replace **ọsọ** with another common noun.

3. Compare the following sentences:
   3. Ka **ọsọ** m na-agba ugbu a ...
   3a. Ka **egbe** m na-gba ugbu a ...
   3b. Ka **mmiri** m na-agba ugbu a ...
   What do all these sentences have in common? The verb "gba" collocates with all the highlighted words which also function as its complement. In other words, we have here NNH.

4. 4. Agbaghị m **ọsọ** ahụ n'efu.
   The inflected verb is "gba" which is negated through the negative suffix ghị. It is complemented by **ọsọ**, its inherent complement NNH.

5. More on pronouns for YES or NO questions, PRNYNQ
   For example, (a) ***Ì*** *riwo mkpụrụ si n'osisi ahụ m nyere gị iwu ka ị ghara iri?* "Have you eaten from the tree that I commanded you not to eat from?" Ì is PRNYNQ.
   (b) *Ọ gaghị enye ya agwọ, ka* ***ọ̀*** *ga-enye ya?* "He will not give him a serpent, will he?" Here, the first 'ọ' functions as PRN, while the second with a low tone mark 'ọ̀' is PRNYNQ. Also, notice that the phrase "ọ̀ ga-enye ya" ends with a question mark '?'. This should be the case of all other pronouns marked with low tones and sentence ends with '?'.
   Note: the answer to any question under this category is either a YES or NO.

6. More examples on qualifactive nouns NNQ and ADJ
   (a) N'ihi na e ji ha bịa n'ihu Jehova, nke mere ha ji di **nsọ**.
   (b) Unu enyela nkịta ihe dị **nsọ**.
   (c) Ndị nchụàjà bụ́ ndị nọ n'ụlọ **nsọ**, na-eme ụbọchị izu ike dị ka ihe na-adịghị **nsọ**.
   (d) A ga-ewere ha rụọ **mbadamba** ọla ndị dị fere fere ...
   The key features of NNQ by Emenanjo (1978); nouns in this lexical class includes words that (1) are inherently semantically descriptive, (2) can only occur after the verb "dị", (3) are found in noun slots in the noun phrase NP and (4) can co-occur with the nominals they qualify. So, "nsọ" in (a), (b) and the second "nsọ" in (c) are NNQ. Emenajo (1978) points out that adjectives can never be used after the verb "dị" and can occur before or after the nominal they modify. For example, you can have ọla ọcha, ọla ojii and so on. Note: Any other words acting to modify nominals, which fails to fit into the five classes of ADJ and their syntactic features should be classified as NNQ.

7. Examples on NND and ADV for clarity purpose.
   (a) O si na mmiri ahụ bilie **ozugbo** "He rose up from that water immediately".
   (b) **Ozugbo** ahụ , ha hapụrụ ụgbụ ha. "Immediately, they left their boat".
   (c) **Ozugbo** a chụpụrụ ìgwè mmadụ ahụ n' èzí .... "Immediately that crowd was drove out, ..."
   (d) Ma **ozugbo** ha pụtara n'ụgbọ .... "And immediately they came out of the boat ..."
   (e) Ọ bụ **ozugbo** ahụ ka ọ bịara. "It is that immediately that he came", that is, "He came immediately (something happened)"
   (f) O ji **nwayọọ** eri nri ya. "He eats his food slowly".
   (g) Ọ dị **nwayọọ**. "It is quiet."
   **Ozugbo** in (a) and (d) are adverbs ADV since they lay emphasize on their verbs 'bilie' (how did he rise?) and 'pụtara' (how did they come out?). **Ozugbo** in (b) and (c) are also emphasizers but they satisfy Emenanjo's adverbial nouns syntactic feature of "Head of the NP", hence they shall be tagged NND. In (e), **Ozugbo** is preceded by the verb 'bụ', which makes it to function as adverbial noun NND and **Ozugbo** does not emphasize on any verb, rather it marks a coincidence. **nwayọọ** in (f) emphasizes how he eats "eri" his food "nri", but since it is preceded by the verb 'ji', which makes it to be found in the noun slot, we shall tag it NND (adverbial noun). Also notice that any other noun can be used in place of **nwayọọ** in the sentence (f) and will still be grammatically correct. For example, *O ji* ***nkazi*** *eri nri ya* "He eats his food with spoon", *O ji* ***efere*** *eri nri ya* "He eats his food with plate", .... Lastly, the verb 'dị' that precedes **nwayọọ** in (g) marks it adverbial noun NND. All other cases of reduplication shall be tagged ADV. For example, *osiiso osiiso, ugboro ugboro, ozigbo ozigbo, ...*

8. Verbs and inherent complements NNH
   These two components form the verbal complex structure (Nwachukwu) in Igbo language. The NNH completes

the meaning of the verb its complementing. It usually occurs on the right sides of the verb and rarely on the left sides. NNH can also be NNC if its not complementing any verb. Verbal components can be either VIF, VSI, VrV, VCO and so on. See item number 1 of this appendix and second appendix for more illustrations. In figure A.3 of the second appendix, using the first example, *ịtụ n'anya*
(a) *Ọ ga-**atụ** gị **n'anya**.* "You will be surprised"
(b) *Ọ bụ ihe **ịtụ n'anya**.* "It is a thing of surprise"
(c) *Ọ **tụrụ** m **n'anya**.* "it surprised me"
(d) *Ọ **tụrụ egwu**.* "He feared"
(e) *Ọ bụ onye **egwu**.* "He is a person of fear"
**anya and egwu** in (a)-(d) are inherent complement component of the verbal complex completing the sense of the verbs **atụ, ịtụ, and tụrụ**. We shall tag them NNH. While **egwu** in sentence (e) is NNC since its not supporting any verb.
**Note:** Always check if a verb makes sence without the complementing noun, that is, check the senses of a verb by changing the complementing noun with other nouns and if grammatically correct, the complementing noun should be tagged NNH.

9. Common and multiwords nouns
(a) *Ọ bịara n'ebe **ịchụàjà** ahụ.* "He came to that alter place"
(b) *ịchụ àjà.* "to sacrifice"
(c) *Onye **nchụ àjà**.* "person who sacrifices"
(d) *Ọgha mkpụrụ.* "person who cultivate"
(e) *ụbọchị ọgba aghara.* "day of confusion"
"ebe **ịchụàjà**" is referring to "alter place", which is a common NNC. When in form of (b), the first part is infinitive verb VIF, and the second part is inherent complement NNH. "Onye" in (c) is pointing to the person that uses the sacrificial instrument **nchụ àjà**. So, **nchụ àjà** is multiword noun having NNCV and NNCC respectively. "Ọgha mkpụrụ" and "ọgba aghara" in (d) and (e) are multiword nouns having NNCV and NNCC respectively. Note that multiword nouns are usually nominalized with pronouns **Ọ, O** and a consonant **n** as in "ikwu okwu" to "okwu okwu", "ịkọwa okwu" to "ọkọwa okwu", "ịgha mkpụrụ" to "ọgha mkpụrụ", "ịgha mmiri" to "ọgha mmiri", "igwu ji" to "ngwu ji", "ịchụ àjà" to "nchụ àjà" and so on. If written together as in "nchụ àjà" or "ọkọwa okwu", it shall be tagged NNC only. Please note that NNCV and NNCC must go together.

10. **Mgbe and nke** are to be classified as common nouns (Emenanjo 1978).
**Mgbe** means "Time, timing or moment" and should be assigned NNC in all cases where it functions as "time, timing or moment". Also, **nke** in all cases where it functions as "thing associated with" should be tagged NNC.

11. **ọ bụla** should be rewritten as **ọbụla** and tagged as QTF since it is found after its nominals it is quantifying. Examples, ihe **ọbụla**, onye **ọbụla**, nwoke **ọbụla**, ...

### A.2.2 The verbal complex structure showing verbs and their inherent complements NNH

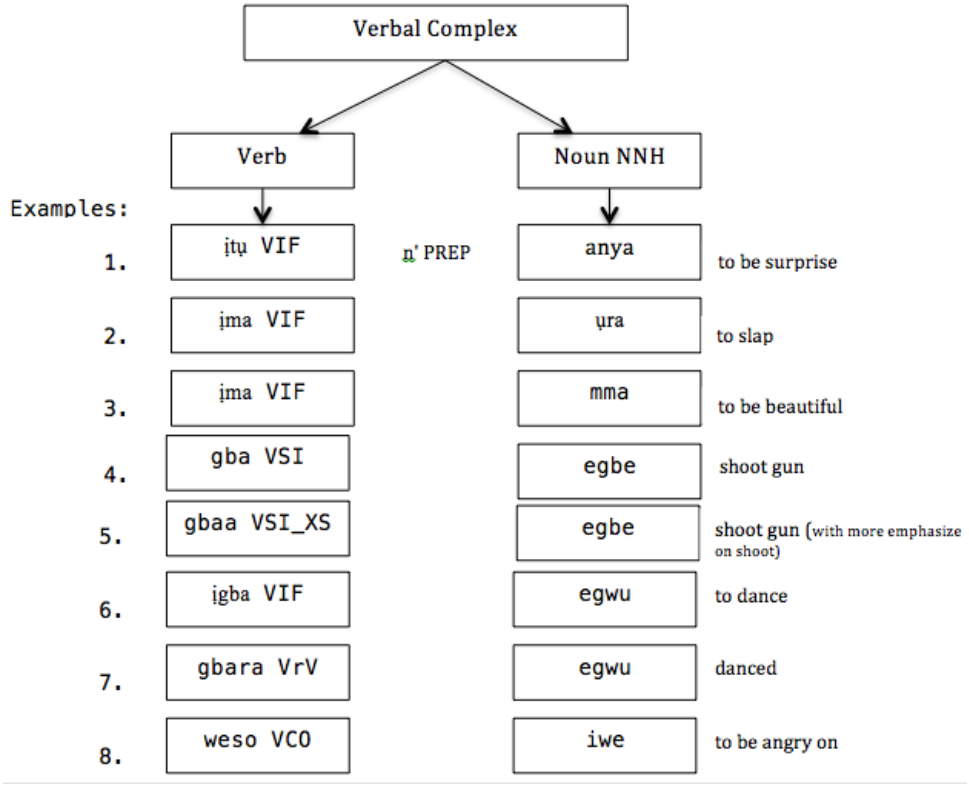



Figure A.3: The verbal complex structure


Figure A.4: The bound pronoun (BPRN) structure. Here, we want to show that the prefix 'E' attached to the simple verb 'si' is as a result of the position of pronoun 'm' in the sentence. It can be rewritten as "M/PRN si/VSI Sheffield abịa"

## A.3 Tables Showing POS Taggers Performances on Precision, Recall, Fmeasure and Averages

Tables in this section display precision, recall, fmeasure and their micro- and macro-averages of each tagger used in chapter 8. Igbo tagged corpus (IgbTC) is a combination of IgbTNT- religious texts and IgbTMT- modern texts.

| Tag | TP | FP | FN | Actual | PREC | RECALL | Fmeasure |
|---|---|---|---|---|---|---|---|
| VAX_XS | 47.7 | 5.2 | 5.2 | 52.9 | 90.091 | 90.228 | |
| BCN | 101.7 | 10.0 | 12.4 | 114.1 | 89.262 | 91.216 | |
| IDEO | 13.4 | 0.7 | 2.8 | 16.2 | 83.975 | 95.336 | |
| VrV | 973.1 | 2.7 | 8.9 | 982.0 | 99.097 | 99.723 | |
| WH_XS | 6.4 | 0.0 | 0.5 | 6.9 | 90.500 | 100.000 | |
| FW | 11.8 | 0.6 | 3.9 | 15.7 | 79.662 | 96.361 | |
| NNCC | 35.5 | 0.9 | 3.8 | 39.3 | 90.105 | 97.705 | |
| VrV_BPRN | 7.6 | 1.6 | 2.1 | 9.7 | 78.853 | 81.526 | |
| WH | 61.6 | 2.4 | 0.0 | 61.6 | 100.000 | 96.120 | |
| VAX_BPRN_XS | 1.0 | 0.7 | 1.5 | 2.5 | 43.667 | 48.333 | |
| DEM | 618.1 | 21.5 | 11.9 | 630.0 | 98.108 | 96.632 | |
| VrV_BPRN_XS | 7.9 | 1.2 | 1.6 | 9.5 | 83.184 | 86.900 | |
| CURN | 2.4 | 0.0 | 0.1 | 2.5 | 96.667 | 100.000 | |
| NNCV | 35.8 | 1.2 | 3.6 | 39.4 | 90.660 | 96.989 | |
| CJN1 | 0.2 | 0.0 | 4.2 | 4.4 | 3.750 | 20.000 | |
| VSI_BPRN_XS | 9.0 | 3.1 | 5.6 | 14.6 | 58.831 | 71.913 | |
| CJN2 | 1.3 | 0.2 | 3.1 | 4.4 | 27.262 | 72.500 | |
| VPP | 677.5 | 18.0 | 21.2 | 698.7 | 96.957 | 97.418 | |
| NNH | 348.6 | 73.9 | 104.2 | 452.8 | 77.034 | 82.480 | |
| ADJ | 103.1 | 1.5 | 5.9 | 109.0 | 94.594 | 98.671 | |
| PRN | 3949.1 | 25.9 | 25.2 | 3974.3 | 99.365 | 99.349 | |
| NNC | 6338.0 | 227.5 | 147.0 | 6485.0 | 97.734 | 96.534 | |
| VPERF_BPRN_XS | 0.2 | 0.2 | 1.5 | 1.7 | 20.000 | 20.000 | |
| QTF | 381.8 | 0.4 | 2.7 | 384.5 | 99.303 | 99.890 | |
| PRNREF | 58.1 | 0.3 | 0.5 | 58.6 | 99.152 | 99.486 | |
| QTF_XS | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| VSI_XS | 1100.8 | 62.1 | 42.1 | 1142.9 | 96.317 | 94.657 | |
| TTL | 5.9 | 0.1 | 0.7 | 6.6 | 90.536 | 99.000 | |
| VAXPRN | 1.4 | 0.0 | 0.7 | 2.1 | 55.833 | 70.000 | |
| VPERF | 49.6 | 7.2 | 5.0 | 54.6 | 91.372 | 87.499 | |
| NNQ | 391.2 | 27.3 | 18.4 | 409.6 | 95.509 | 93.479 | |
| NNP | 836.3 | 4.0 | 4.1 | 840.4 | 99.512 | 99.527 | |
| ADV | 48.8 | 5.3 | 36.5 | 85.3 | 57.011 | 90.026 | |
| ENC | 14.5 | 0.0 | 0.9 | 15.4 | 94.656 | 100.000 | |
| CJN | 1759.3 | 88.5 | 26.3 | 1785.6 | 98.527 | 95.212 | |
| CJN_XS | 44.6 | 0.4 | 0.1 | 44.7 | 99.697 | 99.118 | |
| VPP_XS | 553.6 | 28.3 | 38.9 | 592.5 | 93.430 | 95.140 | |
| VCJ | 266.1 | 0.0 | 0.1 | 266.2 | 99.963 | 100.000 | |
| NNM | 383.0 | 26.9 | 36.6 | 419.6 | 91.320 | 93.448 | |
| INTJ | 20.1 | 0.3 | 1.2 | 21.3 | 93.387 | 98.648 | |
| PRNYNQ | 51.1 | 1.8 | 2.6 | 53.7 | 95.168 | 96.521 | |
| PRNEMP | 34.6 | 0.4 | 0.2 | 34.8 | 99.420 | 98.866 | |
| VAX | 1092.3 | 1.4 | 3.4 | 1095.7 | 99.689 | 99.873 | |
| NND | 66.6 | 12.9 | 15.2 | 81.8 | 81.636 | 83.689 | |
| PREP_XS | 1.8 | 0.0 | 0.4 | 2.2 | 75.833 | 90.000 | |
| ADV_XS | 8.5 | 2.2 | 1.5 | 10.0 | 84.892 | 81.157 | |
| VGD | 21.1 | 1.9 | 5.0 | 26.1 | 80.831 | 91.898 | |
| BPRN | 82.1 | 4.8 | 10.4 | 92.5 | 88.539 | 94.300 | |
| VMOC | 10.8 | 0.1 | 0.9 | 11.7 | 92.187 | 99.231 | |
| LTT | 0.8 | 0.0 | 0.7 | 1.5 | 35.000 | 40.000 | |
| VIF | 246.4 | 5.8 | 20.1 | 266.5 | 92.447 | 97.695 | |
| VCO | 37.8 | 1.1 | 1.6 | 39.4 | 96.106 | 97.092 | |
| VrV_XS | 609.9 | 19.9 | 10.6 | 620.5 | 98.300 | 96.842 | |
| VPERF_XS | 18.7 | 3.4 | 4.8 | 23.5 | 79.636 | 85.495 | |
| VIF_XS | 123.2 | 3.6 | 2.7 | 125.9 | 97.805 | 97.100 | |
| SYM | 3832.4 | 0.0 | 0.4 | 3832.8 | 99.990 | 100.000 | |
| VCO_XS | 4.6 | 0.1 | 1.7 | 6.3 | 71.889 | 98.333 | |
| CD | 1118.3 | 7.5 | 1.4 | 1119.7 | 99.874 | 99.332 | |
| VMOV | 10.8 | 0.1 | 0.9 | 11.7 | 92.187 | 99.231 | |
| VSI_BPRN | 7.4 | 1.6 | 2.2 | 9.6 | 76.688 | 81.112 | |
| DEM_XS | 7.2 | 0.2 | 0.0 | 7.2 | 100.000 | 97.321 | |
| VSI | 1372.9 | 7.1 | 22.4 | 1395.3 | 98.397 | 99.486 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| VAX_BPRN | 31.2 | 0.4 | 0.0 | 31.2 | 100.000 | 98.777 | |
| PREP | 1575.1 | 18.2 | 44.4 | 1619.5 | 97.260 | 98.861 | |
| VPERF_BPRN | 4.1 | 1.2 | 0.7 | 4.8 | 89.683 | 78.548 | |
| ABBR | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| # of tags: 66 | 29635.8 | 745.8 | 745.8 | 30381.6 | | | |
| Macroaverages | | | | | 82.854 | 87.149 | 84.947 |
| Microaverages | | | | | 97.545 | 97.545 | 97.545 |

Table A.2: SLLT POS tagger On IgbTC: Igbo Tagged Texts

| Tag | TP | FP | FN | Actual | PREC | RECALL | Fmeasure |
|---|---|---|---|---|---|---|---|
| VAX_XS | 50.4 | 8.4 | 2.5 | 52.9 | 95.244 | 85.856 | |
| BCN | 96.6 | 13.9 | 17.5 | 114.1 | 84.834 | 87.536 | |
| IDEO | 13.7 | 1.5 | 2.5 | 16.2 | 85.032 | 91.029 | |
| VrV | 973.3 | 4.5 | 8.7 | 982.0 | 99.114 | 99.539 | |
| WH_XS | 6.6 | 2.0 | 0.3 | 6.9 | 95.500 | 76.433 | |
| FW | 13.1 | 1.4 | 2.6 | 15.7 | 84.811 | 88.471 | |
| NNCC | 36.4 | 2.5 | 2.9 | 39.3 | 92.636 | 93.664 | |
| VrV_BPRN | 8.3 | 3.8 | 1.4 | 9.7 | 84.888 | 68.840 | |
| WH | 60.8 | 2.6 | 0.8 | 61.6 | 98.716 | 95.706 | |
| VAX_BPRN_XS | 1.8 | 1.6 | 0.7 | 2.5 | 79.000 | 54.167 | |
| DEM | 623.5 | 92.1 | 6.5 | 630.0 | 98.952 | 87.100 | |
| VrV_BPRN_XS | 7.3 | 1.6 | 2.2 | 9.5 | 75.665 | 83.255 | |
| CURN | 2.4 | 0.0 | 0.1 | 2.5 | 96.667 | 100.000 | |
| NNCV | 36.5 | 2.8 | 2.9 | 39.4 | 92.657 | 92.983 | |
| CJN1 | 1.1 | 0.6 | 3.3 | 4.4 | 21.429 | 44.667 | |
| VSI_BPRN_XS | 9.8 | 3.6 | 4.8 | 14.6 | 65.053 | 71.838 | |
| CJN2 | 1.3 | 0.5 | 3.1 | 4.4 | 26.429 | 59.667 | |
| VPP | 675.1 | 32.3 | 23.6 | 698.7 | 96.604 | 95.434 | |
| NNH | 357.0 | 166.0 | 95.8 | 452.8 | 78.881 | 68.268 | |
| ADJ | 101.3 | 3.5 | 7.7 | 109.0 | 92.932 | 96.705 | |
| PRN | 3877.1 | 14.2 | 97.2 | 3974.3 | 97.552 | 99.636 | |
| NNC | 6158.5 | 221.4 | 326.5 | 6485.0 | 94.967 | 96.530 | |
| VPERF_BPRN_XS | 0.2 | 0.5 | 1.5 | 1.7 | 12.000 | 15.000 | |
| QTF | 381.9 | 1.6 | 2.6 | 384.5 | 99.321 | 99.576 | |
| PRNREF | 58.3 | 0.3 | 0.3 | 58.6 | 99.503 | 99.493 | |
| QTF_XS | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| VSI_XS | 1074.8 | 66.1 | 68.1 | 1142.9 | 94.046 | 94.203 | |
| TTL | 6.1 | 0.1 | 0.5 | 6.6 | 93.710 | 99.000 | |
| VAXPRN | 1.9 | 0.1 | 0.2 | 2.1 | 86.667 | 90.000 | |
| VPERF | 49.2 | 8.3 | 5.4 | 54.6 | 90.308 | 85.816 | |
| NNQ | 378.3 | 39.8 | 31.3 | 409.6 | 92.345 | 90.482 | |
| NNP | 836.9 | 5.1 | 3.5 | 840.4 | 99.591 | 99.396 | |
| ADV | 46.2 | 15.4 | 39.1 | 85.3 | 54.018 | 74.744 | |
| ENC | 14.5 | 0.1 | 0.9 | 15.4 | 94.656 | 99.375 | |
| CJN | 1753.4 | 136.4 | 32.2 | 1785.6 | 98.196 | 92.786 | |
| CJN_XS | 44.5 | 0.4 | 0.2 | 44.7 | 99.639 | 99.115 | |
| VPP_XS | 549.0 | 41.0 | 43.5 | 592.5 | 92.654 | 93.051 | |
| VCJ | 266.2 | 0.0 | 0.0 | 266.2 | 100.000 | 100.000 | |
| NNM | 391.1 | 53.5 | 28.5 | 419.6 | 93.204 | 87.935 | |
| INTJ | 20.4 | 0.3 | 0.9 | 21.3 | 95.063 | 98.547 | |
| PRNYNQ | 51.2 | 2.7 | 2.5 | 53.7 | 95.281 | 94.603 | |
| PRNEMP | 34.6 | 0.2 | 0.2 | 34.8 | 99.420 | 99.377 | |
| VAX | 1092.1 | 0.3 | 3.6 | 1095.7 | 99.670 | 99.973 | |
| NND | 69.1 | 35.5 | 12.7 | 81.8 | 84.437 | 65.882 | |
| PREP_XS | 1.9 | 0.3 | 0.3 | 2.2 | 79.167 | 80.000 | |
| ADV_XS | 8.5 | 4.0 | 1.5 | 10.0 | 85.499 | 69.740 | |
| VGD | 21.9 | 3.6 | 4.2 | 26.1 | 83.839 | 86.532 | |
| BPRN | 88.2 | 7.9 | 4.3 | 92.5 | 95.336 | 91.779 | |
| VMOC | 11.5 | 0.6 | 0.2 | 11.7 | 98.516 | 95.064 | |
| LTT | 1.1 | 0.1 | 0.4 | 1.5 | 42.000 | 48.000 | |
| VIF | 229.0 | 11.7 | 37.5 | 266.5 | 85.896 | 95.161 | |
| VCO | 36.7 | 2.6 | 2.7 | 39.4 | 93.350 | 93.281 | |
| VrV_XS | 606.1 | 27.4 | 14.4 | 620.5 | 97.686 | 95.670 | |
| VPERF_XS | 17.5 | 4.0 | 6.0 | 23.5 | 74.076 | 81.766 | |
| VIF_XS | 110.4 | 9.5 | 15.5 | 125.9 | 87.593 | 92.190 | |
| SYM | 3832.3 | 0.0 | 0.5 | 3832.8 | 99.987 | 100.000 | |
| VCO_XS | 4.6 | 0.2 | 1.7 | 6.3 | 71.889 | 95.000 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CD | 1114.9 | 12.7 | 4.8 | 1119.7 | 99.571 | 98.874 | |
| VMOV | 11.5 | 0.2 | 0.2 | 11.7 | 98.516 | 98.516 | |
| VSI_BPRN | 7.8 | 2.8 | 1.8 | 9.6 | 79.242 | 71.305 | |
| DEM_XS | 7.2 | 0.1 | 0.0 | 7.2 | 100.000 | 98.750 | |
| VSI | 1366.2 | 4.2 | 29.1 | 1395.3 | 97.915 | 99.694 | |
| VAX_BPRN | 31.2 | 0.4 | 0.0 | 31.2 | 100.000 | 98.801 | |
| PREP | 1534.2 | 20.3 | 85.3 | 1619.5 | 94.735 | 98.697 | |
| VPERF_BPRN | 4.1 | 1.6 | 0.7 | 4.8 | 87.683 | 71.000 | |
| ABBR | 0.2 | 0.1 | 0.1 | 0.3 | 15.000 | 20.000 | |
| # of tags: 66 | 29278.8 | 1102.8 | 1102.8 | 30381.6 | | | |
| Macroaverages | | | | | 84.527 | 84.780 | 84.653 |
| Microaverages | | | | | 96.370 | 96.370 | 96.370 |

Table A.3: TnT POS tagger On IgbTC

| Tag | TP | FP | FN | Actual | PREC | RECALL | Fmeasure |
|---|---|---|---|---|---|---|---|
| VAX_XS | 50.5 | 7.4 | 2.4 | 52.9 | 95.435 | 87.327 | |
| BCN | 99.6 | 16.8 | 14.5 | 114.1 | 87.340 | 85.599 | |
| IDEO | 13.3 | 1.4 | 2.9 | 16.2 | 82.500 | 91.341 | |
| VrV | 973.5 | 5.3 | 8.5 | 982.0 | 99.135 | 99.459 | |
| WH_XS | 6.5 | 0.1 | 0.4 | 6.9 | 94.071 | 98.000 | |
| FW | 12.9 | 1.5 | 2.8 | 15.7 | 84.097 | 86.744 | |
| NNCC | 36.3 | 1.5 | 3.0 | 39.3 | 92.348 | 96.128 | |
| VrV_BPRN | 7.4 | 1.6 | 2.3 | 9.7 | 74.834 | 83.515 | |
| WH | 61.1 | 2.3 | 0.5 | 61.6 | 99.151 | 96.272 | |
| VAX_BPRN_XS | 1.5 | 1.1 | 1.0 | 2.5 | 63.667 | 50.833 | |
| DEM | 609.2 | 59.6 | 20.8 | 630.0 | 96.702 | 91.072 | |
| VrV_BPRN_XS | 6.5 | 0.9 | 3.0 | 9.5 | 67.784 | 90.194 | |
| CURN | 2.4 | 0.0 | 0.1 | 2.5 | 96.667 | 100.000 | |
| NNCV | 36.5 | 1.6 | 2.9 | 39.4 | 92.702 | 95.902 | |
| CJN1 | 0.8 | 1.0 | 3.6 | 4.4 | 18.750 | 38.333 | |
| VSI_BPRN_XS | 9.1 | 2.1 | 5.5 | 14.6 | 59.655 | 78.035 | |
| CJN2 | 1.3 | 0.5 | 3.1 | 4.4 | 26.667 | 48.571 | |
| VPP | 673.0 | 28.3 | 25.7 | 698.7 | 96.304 | 95.965 | |
| NNH | 368.9 | 158.4 | 83.9 | 452.8 | 81.504 | 69.944 | |
| ADJ | 102.9 | 2.4 | 6.1 | 109.0 | 94.391 | 97.796 | |
| PRN | 3908.9 | 29.2 | 65.4 | 3974.3 | 98.354 | 99.260 | |
| NNC | 6175.3 | 192.7 | 309.7 | 6485.0 | 95.226 | 96.974 | |
| VPERF_BPRN_XS | 0.1 | 0.3 | 1.6 | 1.7 | 10.000 | 10.000 | |
| QTF | 381.9 | 0.9 | 2.6 | 384.5 | 99.321 | 99.761 | |
| PRNREF | 58.2 | 0.3 | 0.4 | 58.6 | 99.328 | 99.490 | |
| QTF_XS | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| VSI_XS | 1084.8 | 75.8 | 58.1 | 1142.9 | 94.922 | 93.467 | |
| TTL | 6.5 | 0.1 | 0.1 | 6.6 | 98.571 | 99.000 | |
| VAXPRN | 2.0 | 0.1 | 0.1 | 2.1 | 96.667 | 95.000 | |
| VPERF | 47.6 | 7.1 | 7.0 | 54.6 | 87.335 | 87.124 | |
| NNQ | 383.0 | 36.9 | 26.6 | 409.6 | 93.500 | 91.204 | |
| NNP | 837.4 | 8.1 | 3.0 | 840.4 | 99.646 | 99.045 | |
| ADV | 51.5 | 12.6 | 33.8 | 85.3 | 60.097 | 80.074 | |
| ENC | 14.5 | 0.2 | 0.9 | 15.4 | 94.656 | 99.005 | |
| CJN | 1756.9 | 111.6 | 28.7 | 1785.6 | 98.395 | 94.029 | |
| CJN_XS | 44.5 | 0.4 | 0.2 | 44.7 | 99.639 | 99.115 | |
| VPP_XS | 555.4 | 42.4 | 37.1 | 592.5 | 93.739 | 92.913 | |
| VCJ | 266.2 | 0.0 | 0.0 | 266.2 | 100.000 | 100.000 | |
| NNM | 382.1 | 46.0 | 37.5 | 419.6 | 91.092 | 89.230 | |
| INTJ | 20.1 | 0.0 | 1.2 | 21.3 | 93.475 | 100.000 | |
| PRNYNQ | 51.4 | 2.4 | 2.3 | 53.7 | 95.621 | 95.196 | |
| PRNEMP | 34.6 | 0.3 | 0.2 | 34.8 | 99.420 | 99.115 | |
| VAX | 1092.9 | 3.0 | 2.8 | 1095.7 | 99.743 | 99.724 | |
| NND | 68.7 | 27.5 | 13.1 | 81.8 | 84.021 | 71.211 | |
| PREP_XS | 1.8 | 0.3 | 0.4 | 2.2 | 75.833 | 80.000 | |
| ADV_XS | 6.9 | 1.6 | 3.1 | 10.0 | 68.232 | 80.031 | |
| VGD | 21.1 | 4.5 | 5.0 | 26.1 | 80.893 | 83.350 | |
| BPRN | 87.1 | 7.6 | 5.4 | 92.5 | 94.315 | 91.954 | |
| VMOC | 11.4 | 0.4 | 0.3 | 11.7 | 97.747 | 96.619 | |
| LTT | 1.0 | 0.0 | 0.5 | 1.5 | 40.000 | 40.000 | |
| VIF | 241.5 | 14.0 | 25.0 | 266.5 | 90.603 | 94.555 | |
| VCO | 36.6 | 1.6 | 2.8 | 39.4 | 93.228 | 95.786 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| VrV_XS | 603.2 | 27.3 | 17.3 | 620.5 | 97.227 | 95.675 | |
| VPERF_XS | 17.4 | 3.5 | 6.1 | 23.5 | 73.637 | 83.171 | |
| VIF_XS | 108.8 | 9.4 | 17.1 | 125.9 | 86.334 | 92.110 | |
| SYM | 3832.3 | 0.1 | 0.5 | 3832.8 | 99.987 | 99.997 | |
| VCO_XS | 4.6 | 0.3 | 1.7 | 6.3 | 71.889 | 93.810 | |
| CD | 1116.4 | 7.5 | 3.3 | 1119.7 | 99.704 | 99.331 | |
| VMOV | 11.4 | 0.2 | 0.3 | 11.7 | 97.747 | 98.452 | |
| VSI_BPRN | 7.4 | 1.5 | 2.2 | 9.6 | 75.093 | 81.611 | |
| DEM_XS | 7.2 | 0.1 | 0.0 | 7.2 | 100.000 | 98.750 | |
| VSI | 1369.4 | 12.5 | 25.9 | 1395.3 | 98.144 | 99.102 | |
| VAX_BPRN | 31.2 | 0.2 | 0.0 | 31.2 | 100.000 | 99.393 | |
| PREP | 1551.4 | 26.3 | 68.1 | 1619.5 | 95.799 | 98.336 | |
| VPERF_BPRN | 3.9 | 1.3 | 0.9 | 4.8 | 84.921 | 73.869 | |
| ABBR | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| # of tags: 66 | 29369.7 | 1011.9 | 1011.9 | 30381.6 | | | |
| Macroaverages | | | | | 83.452 | 85.559 | 84.492 |
| Microaverages | | | | | 96.669 | 96.669 | 96.669 |

Table A.4: HunPOS tagger On IgbTC

| Tag | TP | FP | FN | Actual | PREC | RECALL | Fmeasure |
|---|---|---|---|---|---|---|---|
| VAX_XS | 47.4 | 7.2 | 5.5 | 52.9 | 89.734 | 86.990 | |
| BCN | 94.6 | 12.4 | 19.5 | 114.1 | 83.005 | 88.538 | |
| IDEO | 12.8 | 0.6 | 3.4 | 16.2 | 80.294 | 96.086 | |
| VrV | 972.5 | 2.5 | 9.5 | 982.0 | 99.033 | 99.744 | |
| WH_XS | 6.2 | 0.0 | 0.7 | 6.9 | 87.071 | 100.000 | |
| FW | 11.4 | 0.1 | 4.3 | 15.7 | 77.947 | 99.231 | |
| NNCC | 34.8 | 3.3 | 4.5 | 39.3 | 88.479 | 91.651 | |
| VrV_BPRN | 6.6 | 0.9 | 3.1 | 9.7 | 67.312 | 88.667 | |
| WH | 61.6 | 2.6 | 0.0 | 61.6 | 100.000 | 95.843 | |
| VAX_BPRN_XS | 1.3 | 1.0 | 1.2 | 2.5 | 55.667 | 51.667 | |
| DEM | 608.9 | 22.5 | 21.1 | 630.0 | 96.643 | 96.431 | |
| VrV_BPRN_XS | 5.7 | 0.3 | 3.8 | 9.5 | 58.598 | 94.405 | |
| CURN | 2.4 | 0.0 | 0.1 | 2.5 | 96.667 | 100.000 | |
| NNCV | 35.9 | 2.3 | 3.5 | 39.4 | 91.053 | 94.408 | |
| CJN1 | 1.0 | 0.7 | 3.4 | 4.4 | 23.929 | 59.000 | |
| VSI_BPRN_XS | 7.7 | 2.3 | 6.9 | 14.6 | 52.249 | 75.572 | |
| CJN2 | 0.9 | 0.4 | 3.5 | 4.4 | 23.929 | 50.000 | |
| VPP | 671.3 | 93.0 | 27.4 | 698.7 | 96.067 | 87.858 | |
| NNH | 322.9 | 74.1 | 129.9 | 452.8 | 71.350 | 81.294 | |
| ADJ | 102.7 | 1.9 | 6.3 | 109.0 | 94.214 | 98.333 | |
| PRN | 3950.7 | 38.8 | 23.6 | 3974.3 | 99.406 | 99.029 | |
| NNC | 6321.6 | 546.9 | 163.4 | 6485.0 | 97.480 | 92.038 | |
| VPERF_BPRN_XS | 0.0 | 0.2 | 1.7 | 1.7 | 0.000 | 0.000 | |
| QTF | 381.6 | 0.4 | 2.9 | 384.5 | 99.250 | 99.889 | |
| PRNREF | 58.0 | 0.4 | 0.6 | 58.6 | 98.974 | 99.294 | |
| QTF_XS | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| VSI_XS | 1003.5 | 30.6 | 139.4 | 1142.9 | 87.808 | 97.045 | |
| TTL | 6.5 | 0.1 | 0.1 | 6.6 | 98.571 | 99.000 | |
| VAXPRN | 1.4 | 0.0 | 0.7 | 2.1 | 55.833 | 70.000 | |
| VPERF | 37.1 | 2.0 | 17.5 | 54.6 | 68.510 | 95.100 | |
| NNQ | 394.4 | 35.8 | 15.2 | 409.6 | 96.284 | 91.679 | |
| NNP | 807.9 | 0.1 | 32.5 | 840.4 | 96.127 | 99.988 | |
| ADV | 48.7 | 6.5 | 36.6 | 85.3 | 56.895 | 88.046 | |
| ENC | 14.5 | 0.0 | 0.9 | 15.4 | 94.656 | 100.000 | |
| CJN | 1759.4 | 92.1 | 26.2 | 1785.6 | 98.532 | 95.030 | |
| CJN_XS | 44.7 | 0.4 | 0.0 | 44.7 | 100.000 | 99.118 | |
| VPP_XS | 491.2 | 37.5 | 101.3 | 592.5 | 82.908 | 92.906 | |
| VCJ | 266.1 | 0.0 | 0.1 | 266.2 | 99.963 | 100.000 | |
| NNM | 392.2 | 27.9 | 27.4 | 419.6 | 93.444 | 93.348 | |
| INTJ | 19.9 | 0.3 | 1.4 | 21.3 | 92.690 | 97.980 | |
| PRNYNQ | 51.4 | 2.2 | 2.3 | 53.7 | 95.801 | 95.550 | |
| PRNEMP | 34.5 | 0.5 | 0.3 | 34.8 | 99.176 | 98.595 | |
| VAX | 1092.1 | 0.2 | 3.6 | 1095.7 | 99.670 | 99.982 | |
| NND | 59.6 | 12.1 | 22.2 | 81.8 | 72.919 | 82.962 | |
| PREP_XS | 1.8 | 0.2 | 0.4 | 2.2 | 75.833 | 85.000 | |
| ADV_XS | 9.5 | 2.2 | 0.5 | 10.0 | 94.766 | 82.438 | |
| VGD | 22.2 | 2.8 | 3.9 | 26.1 | 84.934 | 88.896 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| BPRN | 78.7 | 4.5 | 13.8 | 92.5 | 85.076 | 94.410 | |
| VMOC | 11.6 | 0.2 | 0.1 | 11.7 | 99.286 | 98.571 | |
| LTT | 0.8 | 0.0 | 0.7 | 1.5 | 35.000 | 40.000 | |
| VIF | 242.4 | 12.5 | 24.1 | 266.5 | 90.946 | 95.118 | |
| VCO | 38.0 | 2.3 | 1.4 | 39.4 | 96.577 | 94.247 | |
| VrV_XS | 536.6 | 9.9 | 83.9 | 620.5 | 86.471 | 98.191 | |
| VPERF_XS | 12.8 | 0.5 | 10.7 | 23.5 | 54.085 | 96.310 | |
| VIF_XS | 97.0 | 0.9 | 28.9 | 125.9 | 76.961 | 99.038 | |
| SYM | 3832.3 | 0.0 | 0.5 | 3832.8 | 99.987 | 100.000 | |
| VCO_XS | 4.6 | 0.1 | 1.7 | 6.3 | 71.889 | 98.333 | |
| CD | 1116.2 | 5.4 | 3.5 | 1119.7 | 99.686 | 99.518 | |
| VMOV | 11.6 | 0.2 | 0.1 | 11.7 | 99.286 | 98.571 | |
| VSI_BPRN | 7.5 | 2.4 | 2.1 | 9.6 | 78.038 | 73.648 | |
| DEM_XS | 7.2 | 0.2 | 0.0 | 7.2 | 100.000 | 97.321 | |
| VSI | 1367.2 | 3.8 | 28.1 | 1395.3 | 97.987 | 99.722 | |
| VAX_BPRN | 31.1 | 0.4 | 0.1 | 31.2 | 99.667 | 98.766 | |
| PREP | 1574.1 | 17.3 | 45.4 | 1619.5 | 97.196 | 98.916 | |
| VPERF_BPRN | 3.0 | 0.9 | 1.8 | 4.8 | 71.952 | 78.500 | |
| ABBR | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| # of tags: 66 | 29251.8 | 1129.8 | 1129.8 | 30381.6 | | | |
| Macroaverages | | | | | 80.512 | 86.937 | 83.601 |
| Microaverages | | | | | 96.281 | 96.281 | 96.281 |

Table A.5: FnTBL tagger On IgbTC

| Tag | TP | FP | FN | Actual | PREC | RECALL | Fmeasure |
|---|---|---|---|---|---|---|---|
| VAX_XS | 36.8 | 10.6 | 16.1 | 52.9 | 69.542 | 78.424 | |
| BCN | 59.3 | 18.4 | 54.8 | 114.1 | 51.945 | 76.366 | |
| IDEO | 11.8 | 1.9 | 4.4 | 16.2 | 72.625 | 85.418 | |
| VrV | 972.4 | 12.3 | 9.6 | 982.0 | 99.023 | 98.751 | |
| WH_XS | 6.2 | 0.0 | 0.7 | 6.9 | 87.071 | 100.000 | |
| FW | 11.4 | 0.1 | 4.3 | 15.7 | 77.947 | 99.231 | |
| NNCC | 1.5 | 1.8 | 37.8 | 39.3 | 3.648 | 53.024 | |
| VrV_BPRN | 6.3 | 1.1 | 3.4 | 9.7 | 64.494 | 84.626 | |
| WH | 61.6 | 2.6 | 0.0 | 61.6 | 100.000 | 95.843 | |
| VAX_BPRN_XS | 0.7 | 0.1 | 1.8 | 2.5 | 25.333 | 46.667 | |
| DEM | 630.0 | 223.9 | 0.0 | 630.0 | 100.000 | 73.739 | |
| VrV_BPRN_XS | 5.7 | 0.3 | 3.8 | 9.5 | 58.598 | 94.405 | |
| CURN | 2.4 | 0.0 | 0.1 | 2.5 | 96.667 | 100.000 | |
| NNCV | 34.3 | 2.6 | 5.1 | 39.4 | 86.949 | 93.307 | |
| CJN1 | 0.0 | 0.0 | 4.4 | 4.4 | 0.000 | 0.000 | |
| VSI_BPRN_XS | 2.8 | 1.0 | 11.8 | 14.6 | 17.525 | 62.310 | |
| CJN2 | 0.0 | 0.0 | 4.4 | 4.4 | 0.000 | 0.000 | |
| VPP | 590.8 | 84.8 | 107.9 | 698.7 | 84.544 | 87.457 | |
| NNH | 236.8 | 115.4 | 216.0 | 452.8 | 52.315 | 67.220 | |
| ADJ | 93.6 | 7.9 | 15.4 | 109.0 | 85.837 | 92.351 | |
| PRN | 3773.8 | 100.3 | 200.5 | 3974.3 | 94.953 | 97.413 | |
| NNC | 6249.6 | 1109.2 | 235.4 | 6485.0 | 96.369 | 84.926 | |
| VPERF_BPRN_XS | 0.0 | 0.2 | 1.7 | 1.7 | 0.000 | 0.000 | |
| QTF | 381.6 | 0.4 | 2.9 | 384.5 | 99.250 | 99.889 | |
| PRNREF | 58.5 | 38.6 | 0.1 | 58.6 | 99.821 | 60.209 | |
| QTF_XS | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| VSI_XS | 968.1 | 53.1 | 174.8 | 1142.9 | 84.713 | 94.803 | |
| TTL | 5.2 | 0.0 | 1.4 | 6.6 | 77.480 | 100.000 | |
| VAXPRN | 1.4 | 0.0 | 0.7 | 2.1 | 55.833 | 70.000 | |
| VPERF | 36.7 | 2.0 | 17.9 | 54.6 | 67.759 | 95.066 | |
| NNQ | 384.5 | 67.1 | 25.1 | 409.6 | 93.858 | 85.156 | |
| NNP | 807.9 | 0.1 | 32.5 | 840.4 | 96.127 | 99.988 | |
| ADV | 36.2 | 10.3 | 49.1 | 85.3 | 42.228 | 77.921 | |
| ENC | 14.5 | 0.0 | 0.9 | 15.4 | 94.656 | 100.000 | |
| CJN | 1756.6 | 140.2 | 29.0 | 1785.6 | 98.376 | 92.614 | |
| CJN_XS | 44.7 | 0.4 | 0.0 | 44.7 | 100.000 | 99.118 | |
| VPP_XS | 461.8 | 85.3 | 130.7 | 592.5 | 77.944 | 84.406 | |
| VCJ | 266.1 | 0.0 | 0.1 | 266.2 | 99.963 | 100.000 | |
| NNM | 154.3 | 31.7 | 265.3 | 419.6 | 36.787 | 82.945 | |
| INTJ | 17.0 | 0.3 | 4.3 | 21.3 | 78.466 | 97.556 | |
| PRNYNQ | 50.6 | 1.3 | 3.1 | 53.7 | 94.271 | 97.275 | |
| DEM_XS | 7.2 | 2.0 | 0.0 | 7.2 | 100.000 | 81.065 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| VAX | 1092.1 | 0.2 | 3.6 | 1095.7 | 99.670 | 99.982 | |
| NND | 39.5 | 6.8 | 42.3 | 81.8 | 48.036 | 85.183 | |
| PREP_XS | 1.8 | 0.2 | 0.4 | 2.2 | 75.833 | 85.000 | |
| ADV_XS | 10.0 | 5.0 | 0.0 | 10.0 | 100.000 | 67.523 | |
| VGD | 21.7 | 3.0 | 4.4 | 26.1 | 82.612 | 88.248 | |
| BPRN | 0.0 | 0.0 | 92.5 | 92.5 | 0.000 | 0.000 | |
| VMOC | 0.0 | 0.0 | 11.7 | 11.7 | 0.000 | 0.000 | |
| LTT | 0.8 | 0.0 | 0.7 | 1.5 | 35.000 | 40.000 | |
| VIF | 209.6 | 9.1 | 56.9 | 266.5 | 78.626 | 95.837 | |
| VCO | 38.0 | 2.7 | 1.4 | 39.4 | 96.577 | 93.355 | |
| VrV_XS | 526.9 | 8.3 | 93.6 | 620.5 | 84.910 | 98.449 | |
| VPERF_XS | 12.6 | 0.5 | 10.9 | 23.5 | 53.224 | 96.310 | |
| VIF_XS | 97.0 | 0.9 | 28.9 | 125.9 | 76.961 | 99.038 | |
| SYM | 3832.3 | 0.0 | 0.5 | 3832.8 | 99.987 | 100.000 | |
| VCO_XS | 4.6 | 0.1 | 1.7 | 6.3 | 71.889 | 98.333 | |
| CD | 1117.1 | 30.7 | 2.6 | 1119.7 | 99.767 | 97.325 | |
| VMOV | 0.0 | 0.0 | 11.7 | 11.7 | 0.000 | 0.000 | |
| VSI_BPRN | 4.9 | 1.6 | 4.7 | 9.6 | 48.877 | 71.567 | |
| PRNEMP | 0.0 | 0.0 | 34.8 | 34.8 | 0.000 | 0.000 | |
| VSI | 1364.9 | 2.2 | 30.4 | 1395.3 | 97.821 | 99.840 | |
| VAX_BPRN | 30.3 | 0.4 | 0.9 | 31.2 | 97.172 | 98.736 | |
| PREP | 1523.8 | 10.7 | 95.7 | 1619.5 | 94.093 | 99.304 | |
| VPERF_BPRN | 2.6 | 0.7 | 2.2 | 4.8 | 63.619 | 80.000 | |
| ABBR | 0.0 | 0.0 | 0.3 | 0.3 | 0.000 | 0.000 | |
| # of tags: 66 | 28171.2 | 2210.4 | 2210.4 | 30381.6 | | | |
| Macroaverages | | | | | 67.085 | 75.508 | 71.048 |
| Microaverages | | | | | 92.725 | 92.725 | 92.725 |

Table A.6: MBT tagger On IgbTC