Towards An Effective Igbo Part-of-Speech Tagger

IKECHUKWU E ONYENWE*, University of Sheffield, UK
MARK HEPPLE, University of Sheffield, UK
UCHECHUKWU CHINEDU, Nnamdi Azikiwe University, Nigeria
IGNATIUS EZEANI, University of Sheffield, UK

Part-of-Speech (POS) tagging is a well established technology for most West European languages, and a few other world languages, but it has not been evaluated on Igbo, an agglutinative African language. This article presents POS tagging experiments conducted using an Igbo corpus as a test bed for identifying the POS taggers and the Machine Learning (ML) methods that can achieve a good performance with the small data set available for the language. Experiments have been conducted using different well-known POS taggers developed for English or European languages, and different training data styles and sizes. Igbo has a number of language-specific characteristics that present a challenge for effective POS tagging. One interesting case is the wide use of verbs (and nominalisations thereof) which have an *inherent noun complement*, which form 'linked pairs' in the POS tagging scheme, but which may appear discontinuously. Another issue is Igbo's highly productive agglutinative morphology, which can produce many variant word forms from a given root. This productivity is a key cause of the out-of-vocabulary (OOV) words observed during Igbo tagging. We report results of experiments on a promising direction for improving tagging performance on such morphologically-inflected OOV words.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; *Natural language processing*; Machine Learning; Supervised learning; Supervised learning by classification; Rule learning; Boosting; Language resources; Part of Speech Tagging;

Additional Key Words and Phrases: Natural Language Processing (NLP), Language technology, Corpus annotation, Part-of-speech (POS) tagging, POS tagger, Text Processing, African language, Igbo, Corpora, Morphological Analysis, Machine Learning, Tagset

ACM Reference Format:
Ikechukwu E Onyenwe, Mark Hepple, Uchechukwu Chinedu, and Ignatius Ezeani. 0. Towards An Effective Igbo Part-of-Speech Tagger. *ACM Trans. Web* 0, 0, Article 0 (0), 26 pages. https://doi.org/-

1 INTRODUCTION

Part of speech (POS) tagging is a prerequisite step for many advanced Natural Language Processing (NLP) tasks, and one of the most fundamental steps for processing any new language in NLP. It is the process of assigning the most probable grammatical class (tag) to each word (or token) in a

*This is the corresponding author

Authors' addresses: Ikechukwu E Onyenwe, University of Sheffield, 211 Portobello, Regent Court, Sheffield, South Yorkshire, S1 4DP, UK, ie.onyenwe@unizik.edu.ng; Mark Hepple, University of Sheffield, 211 Portobello, Regent Court, Sheffield, South Yorkshire, UK, m.hepple@sheffield.ac.uk; Uchechukwu Chinedu, Nnamdi Azikiwe University, PMB 5025 , Awka - Nigeria, Awka, Anambra, Nigeria, neduchi@yahoo.com; Ignatius Ezeani, University of Sheffield, 211 Portobello, Regent Court, Sheffield, South Yorkshire, UK, ignatius.ezeani@sheffield.ac.uk.

1559-1131/0/0-ART0 $15.00
https://doi.org/-

text. The majority of African Languages[1] lack the resources and tools (e.g. POS taggers) of a *Basic LAnguage Resource Kit* (BLARK) [18] required for further research and development in the field of Language Technology. These languages are commonly referred to as under-resourced languages [4]. Language technology tools for these languages are developed using datasets collected by researchers which are usually small compared to the technologically favoured languages. Therefore, these under-resourced and lesser-studied languages can be used as an interesting test-bed for the NLP techniques already developed for the well-studied and technologically favoured languages [3].

A review of the literature shows that there has been no previous work on the development of a POS tagger for Igbo. This is significant, as POS tagging gives important information about words and their neighbours which is useful in a broad range of higher NLP tasks such as parsing, chunking, clustering, semantic analysis, machine translation, etc. In this article, we present experiments conducted towards the development of an effective POS tagger for Igbo. This is a part of the Igbo*NLP*[2] project which aims to achieve the development of a Igbo BLARK, so as to facilitate the advancement of NLP for Igbo. The project will also contribute to closing the huge gap in NLP research between Igbo and most European languages. In the LREC (Language Resources and Evaluation Conference) map, English is the most researched language, followed by French and then German[9], and the resources available reflect this level of attention.

To illustrate the broad potential of this work, the paper identifies the existing POS taggers and the Machine Learning (ML) techniques that can achieve a good performance with the small dataset available for Igbo. Also, it identifies the effective methods that work best for the language by using the language available resources, especially on previously unseen words that are morphologically-inflected. We have observed that the majority of the out-of-vocabulary (OOV) words in Igbo tagging are morphologically-inflected. This is because new words are formulated in the language via agglutination.

2 THE IGBO LANGUAGE AND THE CHALLENGES OF DEVELOPING RESOURCES

Igbo is one of the under-resourced languages of the African continent. It is the native language for a subset of Nigerians called Igbo who live in the eastern part of the country. It is a Kwa sub-group language of the Niger-Congo family[3], and one of the most spoken languages of West Africa[4] with its speakers forming about 3% of African and 18% of Nigerian populations[5]. The following sub sections highlight the special features and POS tags specific to the language.

2.1 Language Properties

Igbo has a number of phenomena that are challenging for effective POS taqgging. We consider two key cases relating to the verbal and morphologically-complex structures of the language.

The Igbo verb is typically made up of two components, a verbal and a nominal component called *noun inherent complement(s)* (NIC), and can be represented as *V + NIC* (e.g, *gba (V) egwu (NIC)* 'dance'). NIC is a noun that completes the meaning of a verb. The composite structure has been described as 'verbal complex' in Igbo language studies. These components can occur adjacently, but can also occur with one or more words in between. These complexes can also undergo nominalization, where the verb is converted to a noun, but which retains the NIC noun

[1]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.
[2]It started as a PhD research in 2013 at the University of Sheffield. It is a project partly sponsored by Tertiary Eduction Trust Fund, and Nnamdi Azikiwe University, both in Nigeria
[3]https://www.ethnologue.com/language/ibo
[4]http://www.igboguide.org/HT-igbogrammar.htm
[5]https://www.cia.gov/library/publications/the-world-factbook/geos/ni.html

as its complement. For instance, the English verbs 'dance' and 'sing' are realized in Igbo through the combination of the words *-gba, -gụ* (verbs) and *egwu*, both of which give rise to *-gba egwu* 'dance' and *-gụ egwu* 'sing'. The infinitive form involves the addition of the morpheme 'ị' to get *ịgba egwu* 'to dance' and *ịgụ egwu* 'to sing'. For its nominalization, the morpheme 'ọ' is added, as in *ọgba egwu* 'dancer' and *ọgụ egwu* 'singer'. Hence, the English nominalization suffix is realized in Igbo as a prefix. Also, the modal verb 'can' is written as *nwe ike* in Igbo. The infinitive form of it is *inwe ike* 'to have strength'. It is inflected as a stative verb through addition of the suffix *-re* as in *O nwere ike ịbịa* 'She/he can come'. In a different context, this expression *nwere ike* can mean 'have strength/power'. In general, both the verb and noun of such verb complexes can appear elsewhere without its 'partner', i.e. as just a simple verb or noun, and these different uses are distinguished in our POS tagging scheme. As such, this phenomenon presents a case of ambiguity that may be a challenge for accurate tagging, especially when verb and noun appear no-adjacently.

Igbo has many suffixes and prefixes [23], which can be subclassified into a number of functional classes. Drawing on a number of sources, and from our own corpus work, we have collated a list of 145 Igbo affixes, which is used by our system. Some affixes change the grammatical class (i.e. POS) of a word, e.g. adding the prefix 'a-' to the verb stem 'bia' creates the present participle 'abia'. However, many suffixes do not change the grammatical class, but instead just contribute an additional component to the meaning, in which case there is typically considerable freedom as to the order in which the affixes are attached to a verb stem, so that large sets of morphological variants such as the following may be observed: *abịakwa 'a-bịa-kwa', bịakwaghị 'bịa-kwa-ghị', bịaghịkwa 'bịa-ghị-kwa', bịaghachiri 'bịa-gha-chi-ri', bịachighara 'bịa-chi-gha-ra', bịaghachiriri 'bịa-gha-chi-rịrị'*, etc. This productivity is a key cause of the out-of-vocabulary (OOV) words observed during Igbo tagging, including the case where an unseen form differs only from a previously seen form in the order in which its affixes appear.

3 PART-OF-SPEECH TAGGER AND TAGGING TECHNIQUES

NLP methods for assigning POS tags to words in a sentence can be divided into rule-based and probabilistic approaches. The former approach assigns tags based on rules; rules that can be hand-crafted-based [17] or corpus-based [6]. The latter assigns tags based on probability models [25]. For both approaches, models may be learned by either supervised or unsupervised methods, where supervised learning requires manually POS tagged corpus-data, whilst unsupervised methods do not [7]. The following are widely used and well evaluated supervised POS taggers that have been applied on most European, and a few other world languages.

- *Baseline Tagger*: Unigram in computational linguistics and probability refers to a single word. Therefore, a unigram tagger assign tag based on the 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 used as a default tag for classifying unseen words in the training corpus. Although unigram tagger is a context-independent type of tagger, it can achieve an acceptable results on a large training corpus-data. The result it achieves are normally used as a baseline for more sophisticated taggers.
- *Hidden Markov Model (HMM) Tagger*: HMM consists of a set of states that are connected together by a set of transition probabilities, which indicate the probability of moving between two given states. A process starts with a state, then moves to a new state according to the direction of the transition probabilities. As the process enters each state, a set of output symbol is emitted which is determined by the probability distribution of that state. The exact

sequence of states that the process generates is unknown, hence the name 'hidden'. When using an HMM to perform POS tagging, the goal is to determine the most likely sequence of tags (states) that generates the words in the sentence (sequence of output symbols). A HMM tagger generally chooses a tag sequence for a given sentence rather than for a single word. For instance, given a sentence $w_1 \ldots w_n$, 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 $P(t_1 \ldots t_n)$. Therefore many different taggers have been proposed to simplify this probability computation.

There are *first-order* and *second-order* HMM. In POS tagging, *first-order* HMM is called a bigram tagger. This model works reasonably well in tagging tasks, but captures a more limited amount of the contextual information than is available. While *second-order* HMM taggers use a trigram model, which replaces the bigram transition probability $P_{ij} = P(t_j|t_i)$ with a trigram probability $P_{ijk} = P(t_k \ |t_j, \ t_i)$ [29]. P_{ij} is the probability that tag t_j follows t_i, which can be estimated using the training corpus data.

An example of such a tagger is TnT [5], which is one of the most commonly used HMM based tagger. It uses second order Markov model to simplify the computation; it also assumes that the tag of a word is determined by the POS tags of the previous two words.

- *Maximum Entropy Markov Model (MEMM) Based Tagger*: Unigram and HMM taggers compute 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 a new feature is added, the conditional probability gets harder leading to computational complications. According to [25], MEMM-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 MEMM-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). MEMM-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 [30].

- *Transformation-based Error-Driven Tagger*: This method utilizes rules generated from the training corpus commonly called transformations. These transformations are used to automatically extract linguistic information directly from the training data. The training data is a manually and correctly tagged corpus. The corpus size is usually small, and it serves as input to the initial annotator. Transformation-Based 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 word in an untagged corpus resulting in a temporarily tagged corpus. The unigram tagger derives information for choosing the most probable tag for each word from the tagged corpus called the truth. Iteratively, the temporarily 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 the temporarily tagged corpus resembles or is close to the truth. At the end, this process produces an ordered list of transformations to be applied on the test data. This was originally developed by [6] and subsequently improved both in speed and performance by [19] and [15].

- *Similarity-Based Reasoning Tagger*: Similarity-based reasoning is a method in intelligent systems that draws conclusions by finding similarity between entities. [10] introduce a memory-based supervised learning technique 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.
- Hybrid POS tagging techniques for agglutinative and less-resourced languages. In literature, numerous researches for resource constrained and agglutinative languages have proposed a two stage method for dealing with POS tagging challenges. The first stage involves the use of any or more of the above models for performing full morphosyntactic tagging, while the second one is for identifying morphologically-inflected *word tag* pairs with the help of morphological analysis [8, 24, 26, 28]. For, example, in developing POS tagger for Assamese Text, an agglutinative Indic language, [26] use HMM and simple morphological analysis to determine probable tags for previously unseen words. [8] use a morphological analyzer to improve the performance of the tagger for Bengali developed using HMM and MEMM.

4 METRICS FOR MEASURING TAGGERS PERFORMANCE AND CORPUS PROPERTY

The goal of evaluation in POS tagging is to understand how well a tagger performs on a specific language, either for comparison with other taggers or for understanding where improvement is needed for the language. The standard and generally used evaluation methods are instantiated as follows:

Metrics for Taggers Performance: The followings are the formulas we used to calculate the performance of the taggers:

$$Accuracy = \frac{number\ of\ correct\ tags\ produced\ by\ tagger}{total\ number\ of\ Tokens/tags\ in\ the\ truth} \tag{1}$$

We calculate *precision, recall* and *f-measure* for tag class t using

$$precision_t = \frac{TP_t}{TP_t + FP_t} \tag{2}$$

$$recall_t = \frac{TP_t}{TP_t + FN_t} \tag{3}$$

$$fmeasure = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

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, *micro-average precision=micro-average recall=accuracy.*

Metrics for Corpus Property: For measuring corpus property such as *ambiguity rate* and *ambiguous type*, we used the following formulas:
Ambiguity rate is the average number of tags per token in a corpus, and it is calculated as follows:

$$\frac{Total\ number\ of\ unique\ tags\ per\ word\text{-}type}{Total\ number\ of\ word\text{-}types} \tag{5}$$

Ambiguous type is the average number of identical tokens with more than one tag in a corpus, and it is calculated as follows:

$$\frac{Total\ number\ of\ types\ with\ tag > 1}{Total\ number\ of\ types} \tag{6}$$

5 EXPERIMENTAL RESOURCES

This section describes the resources used in our experiments, including the tagset and corpus.

5.1 The Igbo Corpus and Tagset

The Igbo corpus and tagset developed in [21] PHD research as part of IgboNLP project has been used. The corpus contains several text styles such as essay, news, poem, story, novel, and religious writings. The purpose of using different text styles in our research is to enable the testing of POS taggers on different text styles (i.e., in and out-of-domain testing). As a resource constrained language, the size of this corpus is moderate and can not be compared to the resource rich languages. The corpus represents an exemplar of high-quality writing in Igbo.

Apart from the obvious aim of developing a tagset that will capture the key linguistic features of the language, according to [21] and [23], the tagset designed also focused on capturing the distinction between morphologically-inflected and non-inflected words, to facilitate subsequent investigation of Igbo morphology.

The Igbo tagset has 70 tags. It is collapsible to coarse grain tags of the language, which is 15 [21, 23]. The 70-tag tagset divides into 43 tags indicating tokens that are not inflected and 27 tags for tokens that are morphologically-inflected. Of these 27 tags, 25 have the form α_*XS*, where α is a tag from the group of 43 tags and *XS* indicates the presence of affixes. From Table 1, *VrV* and *VPERF* are tags used to indicate a suffix that is attached to a verb to express various temporal relations of an event that is presently happening, already happened or still to happen. These tags can further be extended to have *XS* to indicate the presence of morphology that is not due to temporal relations (see appendix A). Thus, *VrV* and *VPERF* tags are included as morphologically-inflected tags whether they are *XS* inflected or not. When *XS* is stripped from α_*XS*, the set of 70-tags collapses down to 45. The aim of this division is to capture all words with and without morphology in the Igbo corpus.

Table 1. A selection of some distinctive tags of Igbo tagset. See table in appendix A for full description of tags

NNM	Number marking nouns	BPRN	Bound Pronoun
NNQ	Qualificative nouns	VrV	$-rV$ implies suffix to express simple past if active verb, or stative meaning if stative verb.
NND	Adverbial nouns	VCJ	Conjunctional verbs
NNH	Inherent complement nouns	α_XS	any POS tag with morphology $\alpha \in$ {*CJN,VrV,VPERF,...*}
NNCV	Verb part of multiword noun	NNCC	Noun complement part of multiword noun
VMOV	Verb part of multiword verb	VMOC	Noun complement part of multiword verb
CJN1	Correlative conjunction 1	CJN2	Correlative conjunction 2
VAXPRN	Auxilliary and Pronoun	VPERF	$-PERF$ implies suffix to express perfect tense.

Table 1 lists some of the special tags in the tagset that are peculiar to the language. There are 8 noun classes in the language, viz. the 6 in this table plus common (NNC) and proper (NNP) nouns.

The NNCV is a common noun formed through verb nominalization, VMOV is a modal verb, and the others (NNCC and VMOC) are the inherent complement (C) nouns required to complete the sence of NNCV and VMOV. The CJN 1 and 2 are used to indicate 'correlative conjuction'. They may occur at a close or far distance to each other in a sentence. These tags are used to represent multiword expressions that occur as link pairs (see the language properties in section 2.1). Bound Pronoun (BPRN) represents any pronoun that is tied to a vowel *a/e* prefixed to a verb.

The verb class is made up of 10 tags without *XS* marker, viz. VIF (infinitive), VSI (simple verb), VCO (compound), VMO (Modal), VMOV (modal with complement), VPP (participle), VCJ (conjunctional verb), BCN (Bound Cognate Noun), VGD (gerund), VAX (auxiliary); 2 inflectional classes; and 21 tags with *XS* marker. Out of the 21 tags, 10 are verbs (VrV, VAX, VCO, VPERF, VSI) that can be represented in any of this form: *α_BPRN or α_BPRN_XS*. It means that the prefix *a/e* found in a word with those tags is bound to a pronoun that is preceeding or following it. Full description of this tagset is given in table 12 of appendix A. Refer to [21, 23] for full details of the Igbo tagset and corpus developments.

5.2 The Tagged Igbo Corpora

The Tagged Igbo Corpora (IgbTC) was produced in [21] using the tagset and corpus discussed in the section above. It has nearly 300,000 annotated tokens in total, and contains 67 tags of the 70-tag tagset. The three tags not found, however, are cases of morphologically-inflected tags (i.e. of the form *α*_XS) whose corresponding base form (*α*) is found in the corpus. That means, *VrV* or *VrV_BPRN* may be found in the corpus but *VrV_BPRN_XS* may not be found.

Table 2. Basic statistics of the Tagged Igbo Corpora (IgbTC)

Name	Number of Sentence	Number of Token
IgbTNT	8219	263856
IgbTMT	2032	39960
ESSAY	139	2921
NEWS	17	407
POEM	36	584
STORY	15	248
Total IgbTC	10458	307976

Table 2 shows the basic statistics of IgbTC which comprises six different tagged subcorpora. They are IgbTNT (Igbo Tagged New Testament of the IgbTC represented by Jehovah's witnesses New World Translation Bible), IgbTMT (Igbo Tagged Modern Texts of the IgbTC represented by a novel written in 2013), ESSAY, NEWS, POEM, and STORY. The first and second subcorpora are used for the tagger's development and in-domain testing, while the rest are used for performing out-of-domain testing of the tagger.

5.3 Corpus Property

This section describes the properties (such as tag/token ambiguity) of the *corpus data* we used in the tagger's development experiment. 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 3 shows the general properties of the IgbTNT, IgbTMT, and IgbTNMT (IgbTNT + IgbTMT) that represent the the *corpus data*, while Table 4 and bar charts in Figure 1 show the most ambiguous

tokens and the log-scaled frequency distribution tags. The followings are observed from the figure and tables:

Table 3. IgbTNT, IgbTMT, and IgbTNMT subcorpora general statistics

Properties	IgbTNT	IgbTMT	IgbTNMT
Word size	263,856	39,960	303,816
Sentence size	8,219	2,032	10,253
Type size	6,424	3,122	8,020
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 (AT)	29.73%	34.88%	36.65%
% of AT that are inflected	16.54%	5.58%	15.94%
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%

Table 4. 10 most tag ambiguous tokens in IgbTNMT

Word	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	4,067	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	1,174	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	1,025	NNC=908 NNH=52 BCN=47 VPP=12 NNCC=6

- Word-type size increases as corpus size increases (e.g. from Table 3, 8,020 types used in IgbTNMT > 6,424 and 3,122 used in IgbTNT and IgbTMT).
- *Ambiguity rate* from Table 3 shows that tag/words 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.
- The percentage of *ambiguous words* from Table 3 shows that taggers will disambiguate 29.71% words in IgbTNT, 34.87% words in IgbTMT and 36.65% words in IgbTNMT. This implies that taggers won't struggle to classify the remaining words (e.g. 70.29% words in IgbTNT) since they only get one tag.
- Table 4 shows that the frequent ambiguous words are mainly non-inflected words. This is also quantified by the proportion of the *ambiguous tokens* that are inflected in table 3. For example, in IgbTNT corpus, the percentage of *ambiguous tokens* that are inflected is 16.54%, which implies that there are 83.46% of non-inflected *ambiguous tokens* in the corpus. In Arabic, the

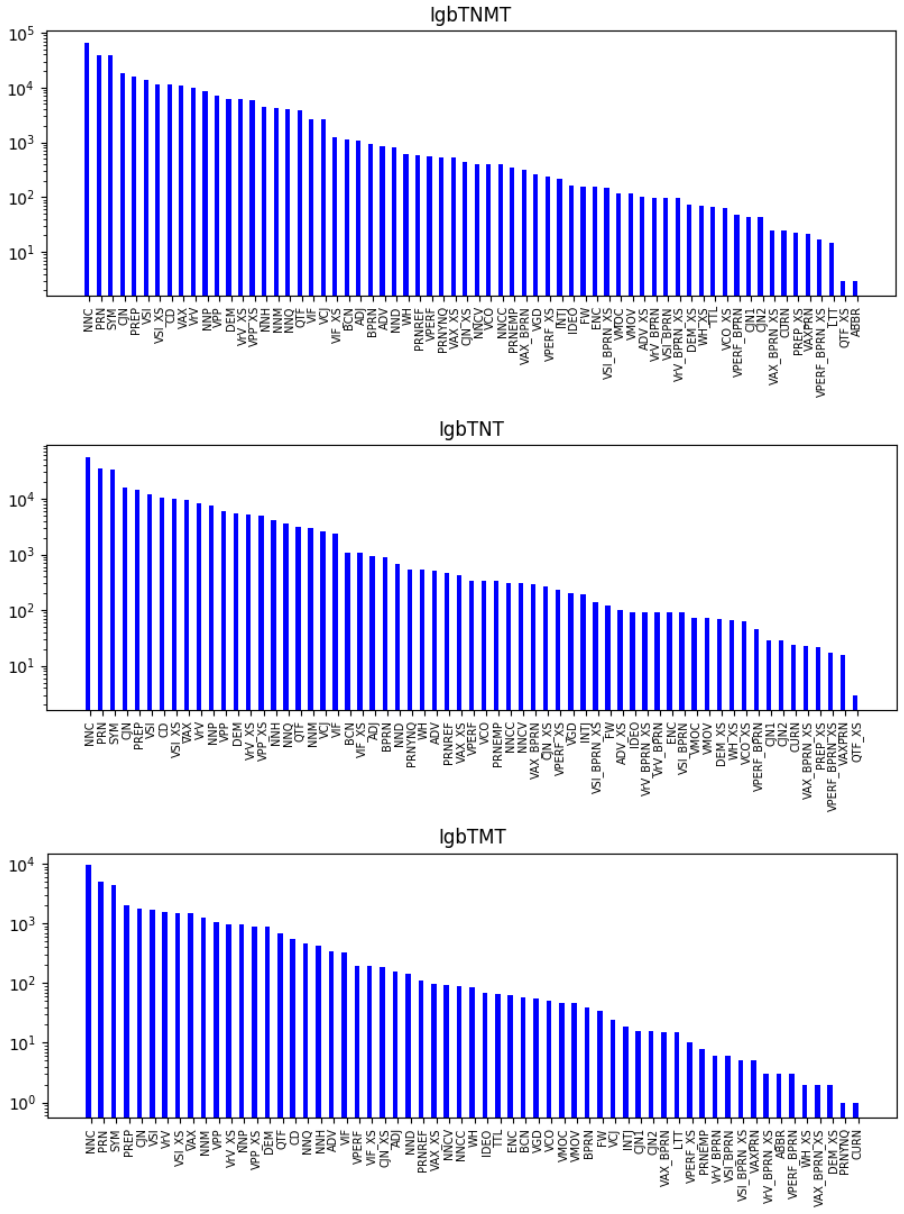


Fig. 1. The log-scaled frequency distribution of tags

highest rate of ambiguity appeared at the stem level, but decreases with inflection, and even decreases further when clitics are added [1]. Also, [14] reveal that the most frequent and ambiguous words in Northern Sotho are not morphologically-inflected. This indicates that the more ambiguous a word is, the more likely it is to have few/no suffixes, and to be more frequent.

- 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*[6] in IgbTNT, so that it 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 words.
- The majority of word types in Igbo are *not inflected* while the majority of word types are *inflected*. Table 3 shows that in IgbTNT *inflected words* account for only 11.89% of *tokens*, whilst accounting for 65.63% of *types*. Within IgbTNT, the *non-inflected* words account for 88.11% of tokens, but only 34.36% of types. Figure 1 shows that tags with *XS* extension, for *inflected words*, have low frequency, and so are skewed on the right. These observations indicate that *inflected words* are one of the major constituents of the rare word set in Igbo.
- Table 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). We used this information for automatic tagging analysis on identical words with more than 4 unique tags in section 7.2.

5.4 POS Tagger Selection and Implementation

Igbo is a language in which a single stem can combine with affixes in multiple different orders to produce variant word forms. The Igbo tagset used in this paper captured words that are morphologically-inflected and non-inflected in the corpus. Taking cognizance of these facts, we chose tagging tools with the following criteria: taggers that are commonly used, have done well on tagging generally, and have parameters for word feature extractions.

Some existing taggers use starting and ending n length of letter sequences of each word as predictive features of words [5, 27]. 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. [30] uses *variable 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. However, it is uncertain how well they will perform in Igbo if not through testing them on the language corpus.

Table 5 shows five selected tagging models used in this paper, and the tools that implement them. The unigram tagger is used to set the baseline accuracy scores which other taggers have to achieve. They are all supervised taggers and represent five types of taggers discussed in section 3. Since they are supervised taggers, a pre-tagged training corpus is required for the taggers' development. We used IgbTNT, IgbTMT, and IgbTNMT corpora for this purpose. To improve the word analysis powers of the taggers, word feature extraction length was set to n=5 because the longest suffixes in Igbo are 5 in length. Furthermore, the taggers achieve best performance at this length[7]

6 EXPERIMENT

Apart from testing the effectiveness of the existing tagging techniques on the Igbo corpus, we also justify why inflected words have their own tagset in the corpus. Given the aim of the Igbo tagset defined in section 5.1, we sought to develop a tagger that can better identify the distinctions

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

[7] 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.

Table 5. Chosen POS taggers

Tagger	Type	Tool
Baseline	Unigram	Self Coded in Python
SLT[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 [30]
[b]Trigrams'n'Tags by [5]
[c]Hungarian Part-of-Speech Tagger is a reimplementation of TnT by [13]
[d]Transformation-based learning in the fast lane. Brill's TBL [6] reimplemented by [19]
[e]A memory-based part of speech tagger-generator by [10]
[f]Tilburg Memory-Based Learner

between morphologically-inflected and non-inflected words in Igbo, which is a valuable step towards automated morphological segmentation of Igbo. Also, we investigate the justification of inflected words having their own tagset by removing morphologically-inflected marker *XS* from tags and training taggers on that set. Then we compare the results with the initial when tags have *XS*.

Furthermore, we performed both in- and out-of-domain testing of the trained taggers on the IgbTC corpus. The corpus was shared into two, we used the first part for the development and in-domain testing of taggers, and the second for out-of-domain testing. This is to practically discuss the performance of the trained taggers tagging similar and dissimilar texts in Igbo. Also, we want to find out how well taggers can identify morphologically-inflected words on a wide range of texts of the language given that new words are mostly formed through agglutination.

6.1 Experimental Setup

We used cross-validation to estimate how accurately the developed taggers will perform in practice. Therefore in order to determine the average accuracies of each, we perform 10-fold cross validation on each of the IgbTNT, IgbTMT, and IgbTNMT subcorpora of the IgbTC corpus in Table 2. We used nine of the ten (90%) as a known tagged texts on which training was run, while the remaining 10% is an unknown but similar texts against which the 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. The remaining subcorpora (ESSAY, POEM, STORY and NEWS) in Table 2 are used later in this paper for out-of-domain testing.

Table 6 shows the average sizes of training and testing data, and the ratios of unknown words. Unknown words arise as a result of words found in the training data that are not in the testing data due to cross-validation. A size of IgbTNT comparable to IgbTMT gives 3.38% unknown word ratio, which indicates that the IgbTMT unknown word ratio is due to its size.

7 PERFORMANCE EVALUATIONS

Using the metrics in section 4 for calculating the taggers performance scores, we observe from the bar charts of Figure 2 (*SLT_s*, *SLT_sp*, and *SLT_** are discussed in section 7.1) that all taggers could achieve an accuracy of 93.17% to 98.11% on the overall words, 7.13% to 83.95% on unknown words,

Table 6. Average statistics of the subcorpora used for the taggers development

Corpus	Train size	Test size	Unknown word ratio	Proportion of unknown words that are morph-inflected
IgbTNT	237,470	26,386	1.19%	77.42%
IgbTMT	35,965	3,996	4.90%	68.37%
IgbTNMT	273,434	30,382	1.39%	74.91%

and 88.02% to 97.46% on morph-inflected words. Unigram and MBT scored zero on the unknown morph-inflected words, while other taggers could achieve accuracy of 78.16% to 87.26%.

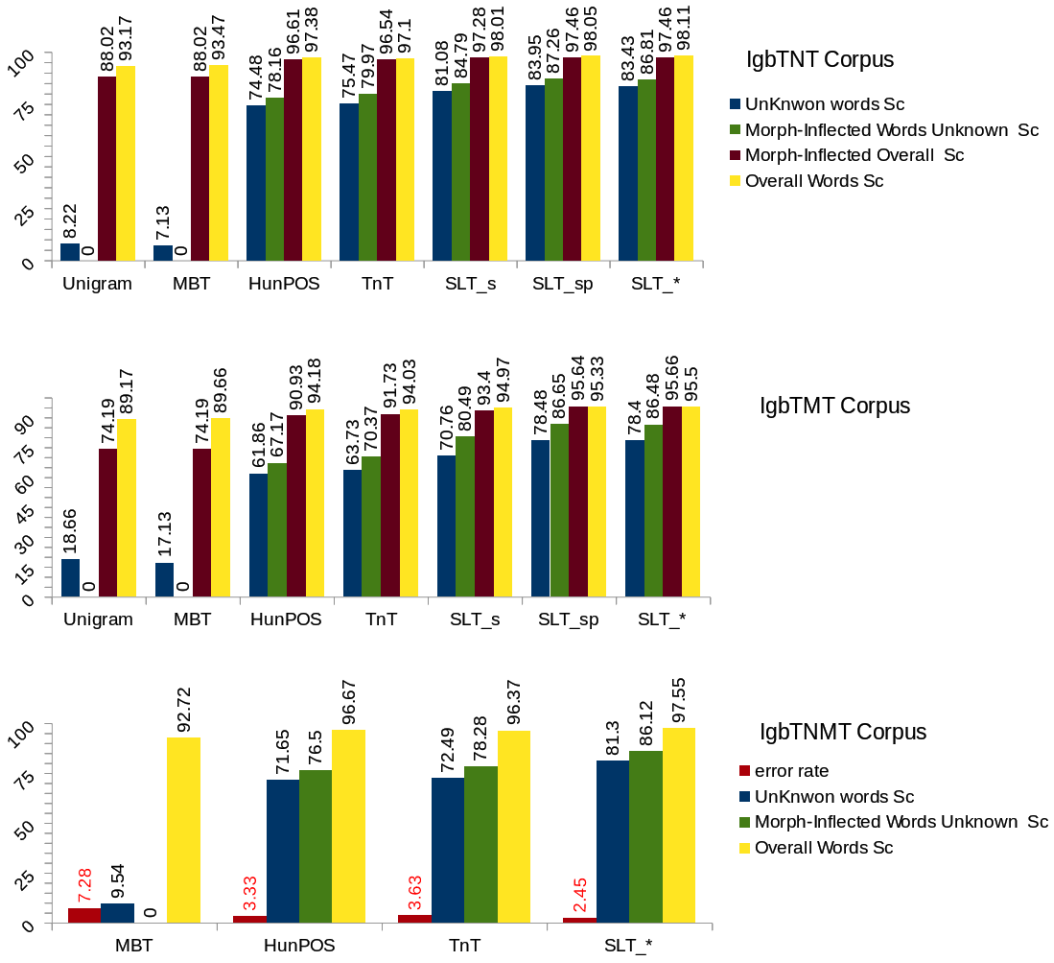


Fig. 2. Accuracy Scores of Taggers.

Ambiguous tokens in Table 3 and the accuracy scores in Figure 2 reveal that out of 29.73% *ambiguous tokens* in IgbTNT, *MBT* tagger correctly classified 22.90%, which is added to the 70.27%

tokens with only one tag, which all the taggers will get for free, to make the overall accuracy score. Also compare other taggers performance on disambiguating the *ambiguous tokens*. Accuracy scores in Figure 2 reveal that the taggers overall performance scores are commendable but not good enough on the unknown words, especially those ones that are morpho-inflected. Generally, the overall scores are good despite the low performance of the taggers on the unknown words, which can be credited to the small size of the unknown words in the subcorpora.

7.1 Tagging Using Different Word Feature Settings

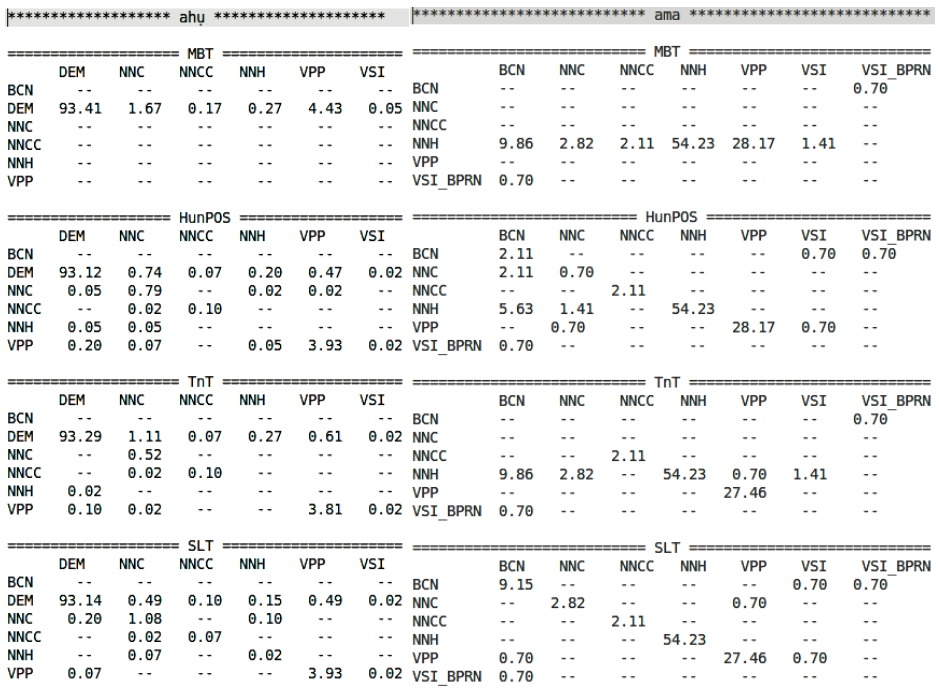
We split SLT tagging experiment into three variations, viz; *SLT_s* means only suffix feature added, *SLT_sp* means suffix and prefix features added, and *SLT_** means suffix, prefix and other features, such as *word shapes*[8] added. From Figure 2, performance scores reveal that SLT performed best on the overall words when in *SLT_** configuration, but performed best on the morph-inflected unknown words and unknown words when in *SLT_sp* configuration. *SLT_s* configuration negatively affects the general performance of the tagger. [30] empirically observe that the prefix features for rare words have a net negative effect on the accuracies, and that its removal considerably increased the unknown and overall words accuracies in the Penn TreeBank English corpus. Conversely, SLT tagger's results using *SLT_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 in Igbo is a good predictive element despite the fact that it is a single character long. We observe that morph-inflected words with a prefix constitute 4.60% of IgbTNMT corpus.

7.2 Evaluating Taggers on Word Level Accuracy

We look at the performance of taggers based on the tags assigned to words with a high number of unique tags. We evaluated this using two most frequent words with a high number of unique tags. From Table 4, we selected top two words *ama* and *ahụ*, and compute the confusion matrices of how the taggers classified them according to their unique tags.

Figure 3 shows the resultant matrices, on top of each matrix are the truth tags while on the left side are the tags assigned by taggers. The word *ama* has seven unique tags and occurs less frequently compared with *ahụ*. It is 54.23% NNH (an inherent noun complementing a verb) and 0.70% VSI_BPRN (a simple verb that is pronoun bound). While *ahụ* is 93.41% DEM and remaining 6.59% is distributed over other tags (NNC, NNCC[9], NNH, VPP[10], VSI[11]). 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%[12], while in other labels, R and P are 0. Across all labels, R=P=A(accuracy)=93.41%. For SLT classifying "ahụ" as DEM, R is 99.71%[13] and P is 98.67%[14] respectively, and across all labels P=R=A=98.24%[15]. We observe from the computed matrices that tags that occur less frequently with *ama* and *ahụ* are difficult to classify by taggers. None of the taggers correctly classified "ahụ" as VSI since it occurs less frequently as

[8]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 [16].
[9]Second pair of a multiwords noun complementing the first pair NNCV.
[10]Participle.
[11]Simple verb.
[12]Using equation 2: $\frac{93.41}{93.41+6.59}$, where 93.41 is true positive (TP) and 6.59 is false positive (FP).
[13]using equation 3: $\frac{93.14}{93.14+0.27}$, where 93.14 is TP and 0.27 is false negative (FN).
[14]Using equation 2: $\frac{93.14}{93.14+1.25}$, where 93.14 is TP and 1.25 is FP.
[15]Sum of diagonals in SLT matrix: 93.14+1.08+0.07+0.02+3.93+0.



ahụ

MBT

	DEM	NNC	NNCC	NNH	VPP	VSI
BCN	--	--	--	--	--	--
DEM	93.41	1.67	0.17	0.27	4.43	0.05
NNC	--	--	--	--	--	--
NNCC	--	--	--	--	--	--
NNH	--	--	--	--	--	--
VPP	--	--	--	--	--	--

HunPOS

	DEM	NNC	NNCC	NNH	VPP	VSI
BCN	--	--	--	--	--	--
DEM	93.12	0.74	0.07	0.20	0.47	0.02
NNC	0.05	0.79	--	0.02	0.02	--
NNCC	--	0.02	0.10	--	--	--
NNH	0.05	0.05	--	--	--	--
VPP	0.20	0.07	--	0.05	3.93	0.02

TnT

	DEM	NNC	NNCC	NNH	VPP	VSI
BCN	--	--	--	--	--	--
DEM	93.29	1.11	0.07	0.27	0.61	0.02
NNC	--	0.52	--	--	--	--
NNCC	--	0.02	0.10	--	--	--
NNH	0.02	--	--	--	--	--
VPP	0.10	0.02	--	--	3.81	0.02

SLT

	DEM	NNC	NNCC	NNH	VPP	VSI
BCN	--	--	--	--	--	--
DEM	93.14	0.49	0.10	0.15	0.49	0.02
NNC	0.20	1.08	--	0.10	--	--
NNCC	--	0.02	0.07	--	--	--
NNH	--	0.07	--	0.02	--	--
VPP	0.07	--	--	--	3.93	0.02

ama

MBT

	BCN	NNC	NNCC	NNH	VPP	VSI	VSI_BPRN
BCN	--	--	--	--	--	--	0.70
NNC	--	--	--	--	--	--	--
NNCC	--	--	--	--	--	--	--
NNH	9.86	2.82	2.11	54.23	28.17	1.41	--
VPP	--	--	--	--	--	--	--
VSI_BPRN	0.70	--	--	--	--	--	--

HunPOS

	BCN	NNC	NNCC	NNH	VPP	VSI	VSI_BPRN
BCN	2.11	--	--	--	--	0.70	0.70
NNC	2.11	0.70	--	--	--	--	--
NNCC	--	--	2.11	--	--	--	--
NNH	5.63	1.41	--	54.23	--	--	--
VPP	--	0.70	--	--	28.17	0.70	--
VSI_BPRN	0.70	--	--	--	--	--	--

TnT

	BCN	NNC	NNCC	NNH	VPP	VSI	VSI_BPRN
BCN	--	--	--	--	--	--	0.70
NNC	--	--	--	--	--	--	--
NNCC	--	--	2.11	--	--	--	--
NNH	9.86	2.82	--	54.23	0.70	1.41	--
VPP	--	--	--	--	27.46	--	--
VSI_BPRN	0.70	--	--	--	--	--	--

SLT

	BCN	NNC	NNCC	NNH	VPP	VSI	VSI_BPRN
BCN	9.15	--	--	--	--	0.70	0.70
NNC	--	2.82	--	--	0.70	--	--
NNCC	--	--	2.11	--	--	--	--
NNH	--	--	--	54.23	--	--	--
VPP	0.70	--	--	--	27.46	0.70	--
VSI_BPRN	0.70	--	--	--	--	--	--

Fig. 3. Confusion matrices of tagging errors made by the taggers on some words with high number of unique tags. SLT in this figure is *SLT_**

VSI. SLT tagger's overall performance is better than other taggers except where HunPOS scored 28.17%[16] in classifying *ama* as VPP, while MBT is least performing tagger.

7.3 Most Frequent Tagging Errors and Language-Specific Challenges

Taggers are evaluated by *tag type error* that occurs when tag t_1 is proposed by tagger but t_0 is the correct tag. Illustrating this with IgbTNMT test corpus, the total number tags where SLT tagger proposed t_1 instead of t_0 is 7,458, which when divided by total number of words in IgbTNMT is 2.45% (compare with error rate in Figure 2). Table 7 shows the distributions of the most common tagging errors by the selected taggers.

Observe that "NNC>NNH" (mistagging NNC as NNH) or "NNH>NNC" (mistagging NNH as NNC) are the top rank error types in most of the lists of different tagging errors from Table 7. This is caused by the verbal complex structure (VCS) of the language property we have discussed in section 2.1. The VCS of the language is made up 2 parts, namely verbal and noun complement components. The noun complement completes the meaning of the verb that it complements, and can be found immediately adjacent to the verb or several words away. Also, the noun complement can function as other noun classes (mostly NNC) if found on it's own. We observe that most of the tagging errors are due to language specific challenges which are discussed in the following sections.

7.3.1 NNC Tag Type Error. The most common error is related to the distinction between Common Nouns (NNC) and Noun Inherent Complement (NNH), Noun Number Marking (NNM), Noun

[16]That is 100% recall for MBT and HunPOS.

Table 7. Top most frequent tagging error made by taggers

SLT		TnT		HunPOS	
$t_0 > t_1$[a]	Error	$t_0 > t_1$[b]	Error	$t_0 > t_1$[c]	Error
NNC Tag Type Error					
NNC>NNH	0.219%	NNC>NNH	0.498%	NNC>NNH	0.479%
NNC>NNM	0.088%	NNC>NNM	0.176%	NNC>NNM	0.151%
NNC>CJN	0.037%	NNC>NND	0.076%	NNC>CJN	0.064%
Total error[d]	0.484%		1.075%		1.019%
NNH and PRN Tags Type Error					
NNH>NNC	0.313%	PRN>DEM	0.276%	NNH>NNC	0.232%
NNH>NNQ	0.015%	PRN>BPRN	0.026%	NNH>NNQ	0.014%
NNH>NND	0.008%	PRN>PRNYNQ	0.009%	NNH>NND	0.012%
Total error	0.343%		0.320%		0.276%
NNH and PREP Tags Type Error					
PREP>CJN	0.127%	NNH>NNC	0.261%	PREP>CJN	0.183%
PREP>VSI_XS	0.013%	NNH>NND	0.016%	PREP>VSI_XS	0.031%
PREP>VrV_XS	0.003%	NNH>NNQ	0.015%	PREP>VrV_XS	0.008%
Total error	0.146%		0.315%		0.224%
Overall total[e]	2.45%		3.63%		3.33%

[a]the total number tags where SLT tagger proposed t_1 instead of t_0 is 7,458.
[b]the total number tags where TnT tagger proposed t_1 instead of t_0 is 11,028.
[c]the total number tags where HunPOS tagger proposed t_1 instead of t_0 is 10,119.
[d]Total errors: tag t_1 is proposed by tagger instead of correct tag NNC.
[e]Summation of total errors. This is equivalent to $error\ rate = 1 - Accuracy * 100$.

Adverbial (NND), and Conjunction (CJN). This error accounts for 0.484% of 2.45% tagging errors made by SLT, 1.075% of 3.63% tagging errors made by TnT, and 1.019% of 3.33% tagging errors made by HunPOS (see Table 7).

A noun is regarded as NNC if

- it is explicitly marked as such in the lexicon: e.g., *ụwa* 'earth', *ala* 'ground', *eluigwe* 'heaven/sky',
- it is marked as NNC when it is not acting as a verb complement: e.g., *egwu* 'dance/fear', *ọsọ* 'run', etc.
- it is a nominalized noun which involves the formation of nouns from verbs and its sense is complete without requiring an inherent noun complement [23]: e.g., *ikesa* 'to separate' nominalized as *nkesa* 'separation', *icheta* 'to remember' nominalized as *ncheta* 'remembrance', etc.
- it is not acting to make a noun to become plural or singular in form or number: e.g., *ndị* 'people', *nwa* 'son', *ụmụ* 'children', etc.

A noun is regarded as NNH if

- it is marked as NNH because it completes the sense of a verb (verb complement): e.g., *ịgụ/Verb egwu/NNH* 'to sing', *ịtụ/Verb egwu/NNH* 'to fear', etc.

A noun is regarded as NNM if

- it is acting to singularize or pluralize a noun. Igbo nouns are not inflected for numbers. Rather, there are words that when preceding a noun modify it to singular or plural [23]: e.g., *ndị Nigeria* 'Nigerians', *nwa* nwoke 'a son', *ụmụ* nwoke 'sons', etc.

A noun is regarded as NND if

- it is found in the noun slot of a noun phrase and may be used immediately after these verbs *bụ*, *ji* and *dị* [12, 23] and are not found in the adverbial slots or elsewhere in the sentence: e.g., *nwayọọ* 'slowly', etc.

7.3.2 NNH Tag Type Error. This is another common error that is related to the mistagging of NNH to be NNC, Qualificative Noun (NNQ), and NND. This error accounts for 0.343% of 2.45% tagging errors made by SLT, 0.276% of 3.33% tagging errors made by HunPOS, and 0.315% of 3.63% tagging errors made by TnT (see Table 7). The properties of NNH and NND tags have been discussed.

A noun is regarded as NNQ if

- used after the verb *dị*. These are nouns that are inherently semantically descriptive. They have been frequently called adjectives but don't have full properties of an adjective [12, 23]: *ogologo* 'tall/height/long', *obosara* 'wide/spread', etc.

7.3.3 PRN Tag Type Error. Another common error found is related to the mistagging of pronoun (PRN) to be demonstrative (DEM), Bound Pronoun (BPRN), and Pronoun Yes/No Question (PRNYNQ). This error accounts for 0.320% of 3.63% tagging errors made by TnT.

A word is regarded as PRN if

- it represents first, second and third person: *a/e* 'impersonal pronoun', *i/ị* 'you', *o/ọ* 'she/he', *m/mụ* 'I', etc.
- it is not bounded to the prefixes *a/e* attached to a verb: *m/PRN na-abịa* 'I am coming'.

A word is regarded as DEM if

- it is used after the nominals. There are only two deictics: *a* 'this' and *ahụ* 'that'.

A word is regarded as BPRN if

- it is a pronoun that is bound to the vowel prefixes *a/e* attached to a verb: *Ana m/BPRN abịa* 'I am coming'. Here *m* is bound to the *a* in *Ana*.

A word is regarded as PRNYNQ if

- it is a question that returns YES or NO answer and the sentence ends with '?': *m̀/PRNYNQ ga-abịa?* 'will I come?', *ò nwere ike iso m wee bịa?* 'can she/he come with me?', etc.

7.3.4 PREP Tag Type Error. This type of error is related to the mistagging of preposition (PREP) to be conjunction (CJN), Simple Verb that is morpologically-inflected (VSI_XS), and active or stative verbs that are morpologically-inflected (VrV_XS). This error accounts for 0.146% of 2.45% tagging errors made by SLT and 0.224% of 3.33% tagging errors made by TnT.

A word is regarded as PREP if

- it takes a noun or pronoun as its nominal complement and not acting as a verb: e.g., *na/n'* 'in, on, under, over', *banyere/gbasara* 'about/concerning', *site* 'through', *tupu* 'before/until', *maka* 'for'. For example, *o kwuru okwu banyere ya* 'she/he spoke about her/him', *ọ bịara tupu ozi ya erute ebe a* 'He came before his message got here', etc.

A word is regarded as CJN if

- it functions as a co-rodinator or sub-ordinator or correlative: e.g., *na* 'and', *nakwa* 'and also', *mgbe* 'when', *ma ... ma ...* 'both ... and ...' (*ma nwoke ma nwanyị* 'both man and woman'), etc.

A word is regarded as VSI_XS if

- it is a simple verb that is morpologically-inflected: e.g., *gbakwa egwu* 'dance also', *bịakwa* 'come also', etc.

A word is regarded as VrV_XS if

- it is an active or stative verb that is morpologically-inflected. *rV* means letter 'r' and vowel. When it is attached to an active verb, it expresses simple past or attached to a stative verb

to express stative meaning: e.g., *banyere* 'entered' (*nwoke ahụ banyere n'ụlọ ya* 'that man entered his house'), *mara mma* 'is beautiful', etc.

This evaluation discusion can be viewed as a sort of guiding principles that are valuable for further development. The most common errors can be viewed as a target-oriented way for future developers investing their time in a maximally effective manner.

7.4 Justification of Morph-Inflected Words XS Tagset

The tags used in the Tagged Igbo Corpus (IgbTC) are in two parts, viz; tags with morphologically-inflected (morph-inflected) marker *XS* to identify morph-inflected words and tags without *XS* to identify words not morph-inflected. To justify the use of *XS* marker, we performed tagging on IgbTNT subcorpus of IgbTC without the tags having *XS* using the SLT tagger , then compared the results with when the tags have *XS*. IgbTNT contains 63 tags with 21 of them having the form *t*_XS, where *t* is the tag and *XS* is maker to indicate morph-inflected. The SLT tagger was set to use suffix, prefix and other features, such as word shapes in this tagging experiments (see *SLT_** in section 7.1).

There are four variations of experiment we conducted in this section. The following itemized points and figure 4 explain the variations and their results.

- **+XS** means that *SLT* tagger was trained, tested, and evaluated on the subcorpus without removing *XS* from the tags. The accuracy score of **+XS** in figure 4 is the same as the accuracy score of SLT in figure 2 on IgbTNT.
- **-XS1** means that we removed *XS* from the tags in IgbTNT subcorpus. This reduced the 63 tags of the subcorpus to 42 tags. Then, SLT tagger was trained and tested on the subcorpus based on 90%:10% cross validation, the evaluation was carried out on the SLT's test results. Results on Figure 4 show that the ratio of tags per word reduced by 0.05 over the ambiguous class, 0.02 on the overall, and word ambiguity percentage ratio reduced by 0.12%. The accuracy scores on the unknown and overall words generally increased. For example, the SLT's accuracy scores increased by 5.23% and 0.09% on the unknown and overall words.
- **-XS2** means that we trained and tested SLT tagger on the subcorpus, then removed the *XS* from the tags in the SLT's test result. There are 63 tags in the test result before removing *XS* and 42 tags after removing *XS*. We carried out evaluation on the test result with 42 tags, and then compared the accuracy scores on *-XS2* and *+XS*. We observed that the accuracy scores increases when *XS* is removed from the SLT's test result, which is *almost* equivalent to *-XS1* when *XS* is removed from the corpus before training and testing SLT on the subcorpus. Therefore, we can decide to ignore morph-inflected words having their own *XS* tagset or to retain it, since both approaches deliver good accuracy scores. But *+XS* gives extra information about the morphological parts of the language which makes the Igbo corpus to be more informative.
- In **Coarse Grain**, we mapped down the 63 tags used in the subcorpus to 15 course-grain tags designed for Igbo. Then, we trained, tested and evaluated SLT on the subcorpus with 15 tags (no *XS*). Results on Figure 4 show that the ratio of tags per word reduced by 0.25 and 0.08 over the ambiguous and overall classes, and the 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 SLT's accuracy scores increased by 11.46% (unknown words), 1.08% (known words) and 1.20% (overall) 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 smaller the tagset the more accurate is the tagging performance of the taggers [2, 11]. That means that there are fewer cases of ambiguous

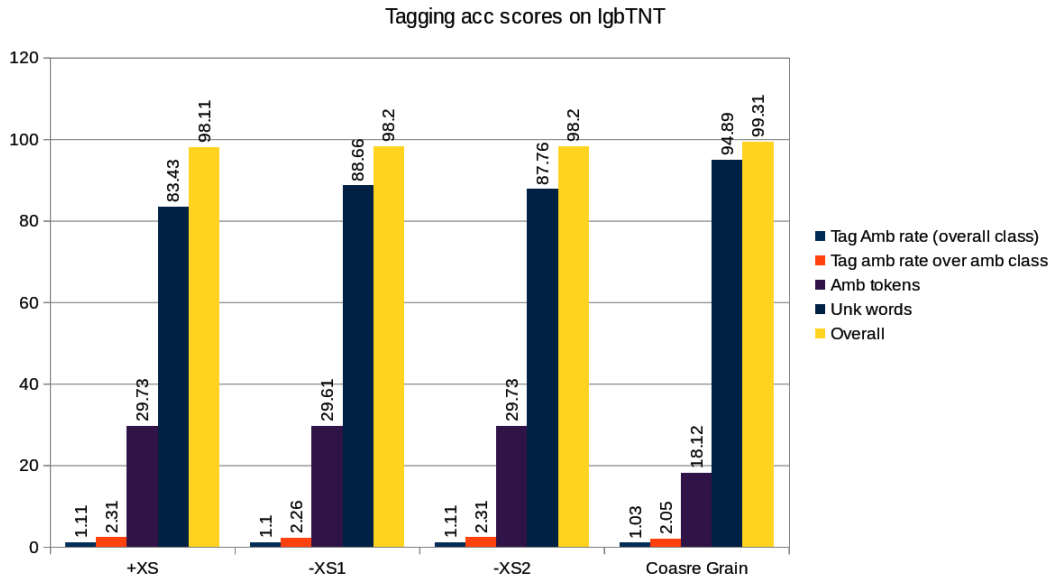


Fig. 4. Accuracy Scores of Taggers.

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 in a sentence. For example, if we decide to do away with the *XS*, the goal of capturing the morph-inflected words, which also are part of the Igbo grammar will be defeated. Capturing morph-inflected words is one of the key points towards performing full-scale computational morphology in Igbo.

7.5 Tagging Morphologically-Complex Words

One of the major aims of this work is to develop a tagger that will capture a good number of morphologically-inflected words in Igbo, as a step towards further morphological analysis. Regarding the standard POS taggers that we have investigated, Figure 2 shows their poor performance on the previously unseen words, and Table 6 shows that morphologically-complex words are dominant amongst the previously unseen words, e.g. with 74.91% of the previously unseen words in IgbTNMT being morphologically-inflected. This is because in Igbo, a stem can combine with affixes in multiple different orders to form different word variants. For example, the word *bụ* occurred 3794 times as a word and 2579 times as a morph-inflected words (e.g., *bụrụ-1008, bụkwa-124, bụrụkwa-108, abụkwa-27, abụrụkwa-2, bụrụkwanụ-2,* etc). This productivity is a key cause of the OOV words encountered during tagging, including cases where an unseen form differs only from a previously seen form in the order in which its affixes appear. Therefore, the automatic handling of such words is an important and challenging task for POS tagging in Igbo.

To address this issue, we investigate the possibility of exploiting knowledge of Igbo morphology to improve performance, as opposed to the approach taken by many standard taggers, of using the initial/final character strings of words as a proxy for linguistically-real affixes. This section introduces a hybrid tagging approach that includes the use of a probablistic tagger, rule-based

tagger and morphological analyzer components. The morphological analyzer component uses a simple lexicon-based method to extract and analyze morpologically-inflected words from the Igbo corpus in the tagging system architecture. The lexicon is a list of 145 Igbo suffixes we collected from the Igbo grammar book by [12] and the Igbo corpus. Our aim is to use more linguistic knowledge to more effectively identify and analyse morphologically-complex handle Igbo words.

Although, the tagging framework builds on an existing POS tagging algorithm, it is extended with a morphological analyzer components that is dependent on the use of the actual linguistic suffixes to fit the agglutinative language. The tagging system process involves the use of the following algorithms.

Algorithm 1 Algorithm for morphologically-inflected words used on train data

1: **Input**: train data
2: suffixDict ← 145 Igbo suffixes
3: **while** there is word *w* in input **do**
4: check if *w* string combinations exist in *suffixDict*
5: suffixlist ← store valid strings combinations of *w* if 4 is true
6: trainCandidates ← store *w* against its *suffixlist*

Algorithm 2 Algorithm for morphologically-complex words used on test data

1: **Input**: test data
2: suffixDict ← 145 Igbo suffixes
3: **while** there is word *w* in input **do**
4: check if *w* strings combinations exist in *suffixDict*
5: suffixlist ← store valid strings combinations of *w* if 6 is true
6: **if** *w is not in trainCandidates of algorithm 1* **then**
7: testCandidates ← store *w* against its *suffixlist*

Algorithm 1 is applied to the training data to extract all the morphologically-inflected words, which are stored as *trainCandidates*. Algorithm 2 extracts the morphologically-complex words from the test data, which are stored as *testCandidates*. This set excludes any words that are also found in *trainCandidates*. A rule-based tagger is trained over *trainCandidates* to generate the rules of its model, which is then applied to *testCandidates*. The end result is a tagged list of morphologically-complex words. In both algorithms, *suffixDict* is a dictionary of Igbo suffixes. Table 8 illustrates the format of data in *trainCandidates* and *testCandidates*.

Table 8. Format of data in *trainCandidates* & *testCandidates* of Algorithms 1 & 2

Morph-Inflected Word *w*	Morphological Parts *suffixlist*	Initial State	Truth State
ahụtụbeghị	a	PREFIX	PREFIX
	hụ	VrV	VPP_XS
	tụ	SUFFIX	SUFFIX
	be	SUFFIX	SUFFIX
	ghị	SUFFIX	SUFFIX

Observe in Table 8 that each affix or stem has *two* alternative labels: an *initial* label and a *true* label. This is because the rule-based tagger we use for this task employs the *transformation-based error-driven learning* (TBL) algorithm of [6]. TBL requires a *truth state* representation of the data, i.e. showing the correct label for each item. TBL also creates an *initial state labelling* of the data,

typically using a simple method, such as assigning each item its most common label. The initial state will contain many errors. TBL then proceeds to learn a series of *transformation rules*, that correct errors in the initial state, so that it better approximates the truth state. These rules are context-dependent, i.e. can apply to replace label X with Y provided the context meets some requirement, e.g. that the item to the left is some specific w or the label to the right is some specific t. At run-time, TBL labels unseen data by creating its initial state, and then applying the sequence of transformation rules learned during training.

As shown in Table 8, each affix is labelled as either PREFIX or SUFFIX, and the initial and true values are the same. The stem, however, is marked with a POS tag, suited to a full word. This is because we want to use the morphological class clues to predict the true tag of the stem. which is returned then as the tag of the full word. The initial state label for each stem is assigned based on the most common errors made by the standard taggers. For the example *ahụtụbeghị* ('I have never seen') in Table 8, the initial tag VrV tag is assigned, as this is the most common tag erroneously assigned to it by the taggers.

As differences between the initial and truth states (i.e. errors) arise only for stems, TBL will only learn rules to modify stem tags. Examples of rules generated using the information in Table 8 are rules that: (1) change *VrV* to *VSI_XS* (simple verb that is morph-inflected) due to the suffixes; then (2) change *VSI_XS* to *VPP_XS* (morph-inflected participle) due to the prefix 'a'. In Igbo, if a prefix 'a/e' precedes a stem, it is highly probable that that word is a participle. Related work of this kind has been reported in [22] and [20].

7.5.1 Setup, Experiment, and Performance Evaluation. In this experiment, we used FnTBL [19], a re-implementation of [6] TBL, to implement the rule-based approach for handling morph-complex unknown words; SLT, TnT, and HunPOS taggers to represent the probabilistic taggers that have parameters for processing unknown words; and the IgbTNT corpus as our experimental data (see Table 6 for statistics).

SLT, TnT, and HunPOS taggers were trained and tested on the entire corpus while FnTBL was trained only on the morph-inflected words (*trainCandidates*) of the train data; and tested only on the morph-complex unknown words (*testCandidates*) of the test data, acquired using Algorithms 1 and 2. We compare the outputs of all the taggers on the morph-complex unknown words.

Figure 5 shows tagging results on the morphologically-complex unknown words (top chart), unknown words (middle) and words overall (bottom). Here, TBL is the FnTBL tagger using morphological clues to tag morphologically-complex words, while *SLT+TBL, TnT+TBL and HunPOS+TBL* are when both taggers are used together, with the TBL predicted tags for morph-complex unknown words replace those predicted by the other taggers. There is an impressive improvement of the accuracy scores on the morphologically-complex words when TBL is applied, showing the benefits of using a linguistically-informed approach.

7.6 Out-of-Domain Evaluation

Igbo generates many word variants through various morphological processes, and so a POS tagger that can identify and handle these new words in a range of texts is important. For this reason, we compared the performance of the SLT, TnT, HunPOS and TBL taggers, developed from the above processes using IgbTNMT corpus, on the four out-of-domain texts of the corpus.

The evaluation shows the effect of using the true suffixes of the language instead of the initial/final character strings used as a proxy for the true suffixes by most taggers.

Table 2 gives the descriptions of texts in the Igbo corpus, and Table 9 shows the size of the out-of-domain texts in terms of words and morphologically-complex unknown words. For each

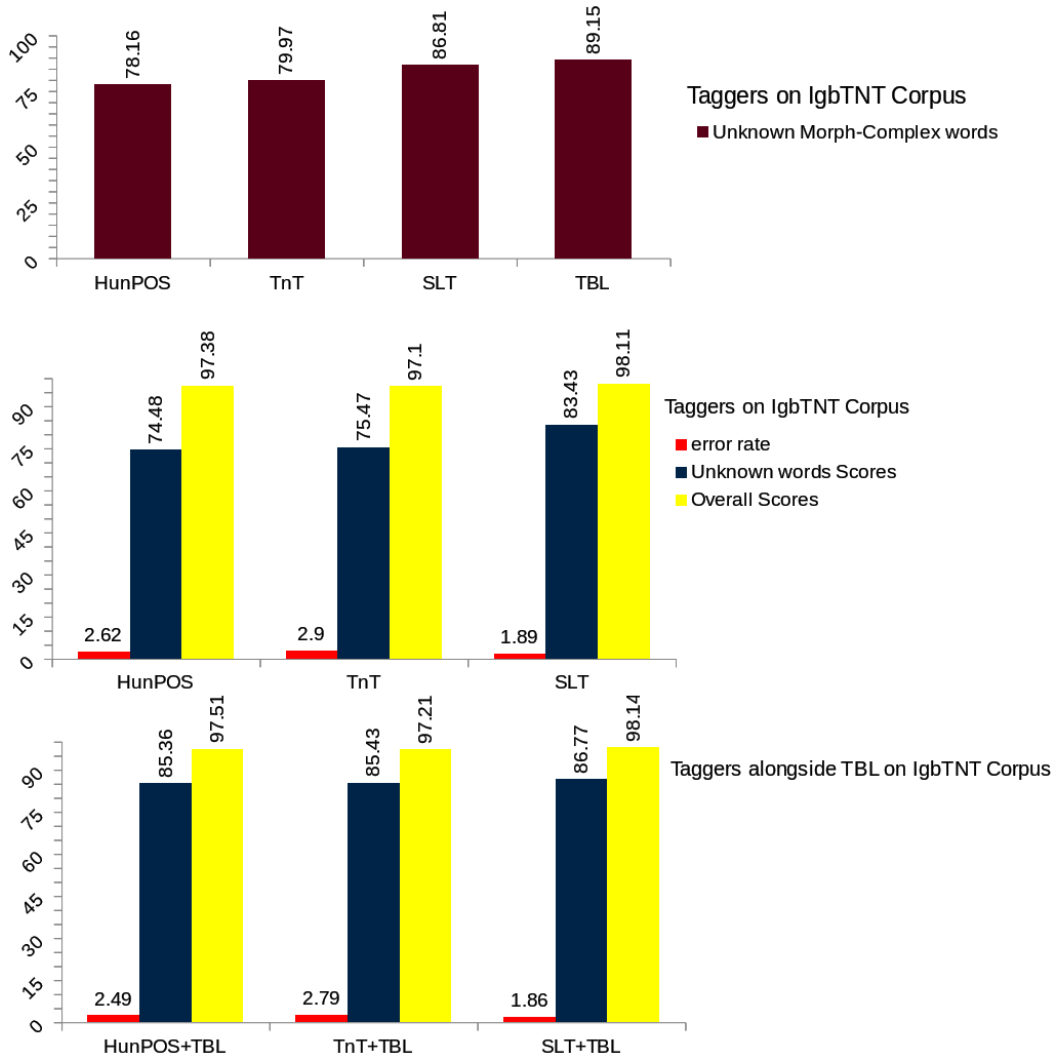


Fig. 5. Taggers performances on IgbTNT corpus.

Table 9. Statistics of dissimilar texts used

Corpus	Number of tokens	Unknown words	Unknown words that are morph-inflected
ESSAY	2,921	177	93
NEWS	407	80	16
POEM	584	83	30
STORY	248	11	11

Table 10. Accuracy scores of taggers on morphologically-complex unknown words. Taggers developed on IgbTNMT corpus and tested on different styles of Igbo text

Test data	Hun	TnT	SLT	TBL
ESSAY	67.74	67.74	89.25	91.40
NEWS	56.25	56.25	68.75	81.25
POEM	36.6	33.33	70.00	86.67
STORY	63.64	90.91	72.73	100.00

of the out-of-domain text, we used algorithm 2 to detect, reconstruct and classify words that are unknown and morphologically complex into PREFIX, STEM, and SUFFIXES (as in table 8). For example, ESSAY is 0.96% of IgbTNMT, and there are 93 morphologically complex words detected in ESSAY not found in IgbTNMT. We judge the taggers' performances based on these detected words that are labelled unknown and morphologically complex. Table 10 shows the performance scores of the taggers used. SLT, HunPOS and TnT were trained on the IgbTNMT training set (see section 6.1), while TBL was trained only on the morphologically-inflected words of IgbTNMT. The performance scores reveal that TBL outperformed the other taggers by several points, despite the fact that it works only on individual words, and cannot exploit contextual information, such as the surrounding words of the sentence, as the other taggers can.

8 CONCLUSION

There has been no previous work assessing the use of the part-of-speech (POS) tagging techniques, established on richly-resourced and well-studied languages, to support Natural Language Processing (NLP) tasks on Igbo. This article compares the effectiveness of six state-of-the-art POS taggers on the Igbo POS Tagged Corpora (IgbTC) comprising six different texts styles. We empirically observe that the results achieved by the taggers on the known words of IgbTC are quite satisfactory, but unsatisfactory on the unknown words, especially the morphologically-complex ones. It is commendable that these independent taggers that have been developed using some of the European languages could achieve good accuracy scores on the known words of Igbo despite the morphological complexity of the Igbo language. Since one of our major aim is to develop a tagger that will capture a good number of morphologically-inflected words for further research on computational morphology, we have investigated methods to improve upon the poor performance of taggers on the morphologically-complex unknown words.

Our experiments reveal that a major cause of the poor performance of taggers on unknown words is ineffective handling of the words that are morphologically-complex. We observe from our data that morphologically-complex words constitute the majority of the rare/unknown words class in Igbo, unlike English where it is mostly proper nouns. In Igbo, a single root can produce many word forms by combination with affixes in various orders. The more affixes attached to a word, the more likely it is to be rare. We have developed a Transformation-Based Learning (TBL) tagger that uses the knowledge of stems and associated affixes to process morphologically-complex unknown words. This system achieve significantly higher scores on these words than the other taggers employed in our experiments.

A comparative analysis that involves the use of taggers trained on IgbTNMT, a part of the main corpus IgbTC, to tag dissimilar Igbo texts indicates that our linguistically-informed approach to handling morphologically-complex unknown words is robust in achieving good performance across text genres. This method, using TBL and Igbo suffix dictionary, achieves an impressive accuracy

scores ranging 82%-100% on four dissimilar types of Igbo text. Using our system alongside the other probabilistic taggers, and replacing the tags they assign to morphologically-complex unknown words with the tags produced by our system, yields a a considerable increase in the accuracy achieved for these words.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the Tertiary Educational Trust Fund (TET-Fund) Nigeria and partly supported by Nnamdi Azikiwe University. Many thanks to my colleagues at the IgboNLP Project, University of Sheffield, UK.

A TABLE OF THE IGBO TAGSET (IGBTS) AND DESCRIPTION

This section describes the part of speech tags in the corpus data used for the experiments.

Table 11. Tags description and usage

Tag Name	Description
NNP	Proper noun
NNC	Common noun
NNM	Number Marking Noun for plurality
NNQ	Qualificative noun
NND	Adverbial noun
NNH	Inherent complement noun, used to complete a verb sense
NNCV	Multiword noun formed via verb nominalization
NNCC	Inherent complement noun of NNCV
VIF	Infinitive verb
VSI	Simple verb
VCO	Compound verb
VMO	Modal verb supplemented by modal suffixes
VMOV	Modal verb that require inherent complement noun
VMOC	Inherent complement noun of VMOV
VAX	Auxiliary verb
VPP	Participle
VCJ	Conjuctional verb
BCN	Bound Cognate Noun
VGD	Gerund
ADJ	Adjective
PRN	Pronoun
PRNREF	Reflexive pronoun
PRNEMP	Emphatic pronoun
PRNYNQ	Pronoun Yes/No Question
BPRN	Bound pronoun
ADV	Adverb
CJN	Conjuncion
CJN1	First correlative conjunction
CJN2	Second correlative conjunction
PREP	Preposition
QTF	Quantifier
DEM	Demonstrative
INTJ	Interjection
FW	Foreign/Borrowed word
SYM	Punctuations
CD	Numbers
WH	Interrogative
IDEO	Ideophone
LTT	Alphabets/Letters
TTL	Title
ENC	Collective, adverbial additive, negative interrogative, adverbial confirmation, adverbial immediate, present and past

Table 12. Tags description and usage for morphologically-inflected tags

Tag Name	Description
VrV	Active/Stative verb
VPERF	Perfect tense
α_XS	any POS tag with affix. $\alpha \in$ {VIF,VSI, VCO, VPP, VGD, VAX, CJN, WH, VPERF, VrV, PREP, DEM, QTF, ADJ, ADV}.
α_BPRN / *α_BPRN_XS*	Any verb whose vowel prefix *a/e* is bound to a pronoun that precedes or follows it. $\alpha \in$ {VrV, VAX, VCO, VPERF,VSI}
VAXPRN	Auxilliary with dependent pronoun for subject.

REFERENCES

[1] Mohammed A Attia. 2008. *Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation.* Ph.D. Dissertation. University of Manchester.

[2] ES Atwell. 2008. Development of tag sets for part-of-speech tagging. Walter de Gruyter.

[3] Cheikh M. Bamba Dione, Jonas Kuhn, and Sina Zarrieß. 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)*. European Language Resources Association (ELRA).

[4] Laurent Besacier, V-B Le, Christian Boitet, and Vincent Berment. 2006. ASR and translation for under-resourced language. In *Proceedings of Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, Vol. 5. IEEE International Conference on-Volume 5. IEEE.

[5] Thorsten Brants. 2000. TnT: A Statistical Part-of-speech Tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Association for Computational Linguistics. Stroudsburg, PA, USA, 224–231.

[6] Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics* 21, 4 (1995), 543–565.

[7] Eric Brill. 1995. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the third workshop on very large corpora, vol. 30, pp. 1–13. Somerset, New Jersey*, Vol. 30. Association for Computational Linguistics, 1–13.

[8] Sandipan Brill, EricDandapat, Sudeshna Sarkar, and Anupam Basu. 2007. Automatic Part-of-speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Prague, Czech Republic*, Vol. 30. Association for Computational Linguistics. Stroudsburg, PA, USA, 221–224.

[9] Nicoletta Calzolari, Riccardo Del Gratta, Gil Francopoulo, Joseph Mariani, Francesco Rubino, Irene Russo, and Claudia Soria. 2012. The LRE Map. Harmonising Community Descriptions of Resources.. In *LREC*. 1084–1089.

[10] Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In *arXiv preprint cmp-lg/9607012*.

[11] G. De Pauw, Gilles-Maurice de Schryverz, and J.ṽan de Looy. 2012. Resource-Light Bantu Part-of-Speech Tagging. In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages, SaLTMiL 8–AfLaT2012*. European Language Resources Association (ELRA), 85–92.

[12] Nọlue E Emenanjo. 1978. *Elements of Modern Igbo Grammar: A Descriptive Approach.* Ibadan Oxford University Press.

[13] Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. HunPos: an open source trigram tagger. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 209–212.

[14] U. Heid, E. Taljard, , and D.J̃. Prinsloo. 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*.

[15] Mark Hepple. 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*. Association for Computational Linguistics, 278–277.

[16] Daniel Jurafsky and James H Martin. 2016. Part of Speech Tagging. Speech and language processing, Draft of November 7, 2016, Academic Press Limited. https://web.stanford.edu/~jurafsky/slp3/9.pdf.

[17] F Karlsson. 1995. Designing a parser for unrestricted text. In *F. Karlsson, A. Voutilainen, J. HeikkilÃd, and A. Anttila, eds., Constraint Grammar âĂŤ A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin, New York, 1–40.

[18] Steven Krauwer. 2003. The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. *Proceedings of SPECOM 2003* (2003), 8–15.
[19] Grace Ngai and Radu Florian. 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*. Association for Computational Linguistics, 1–8.
[20] Ikechukwu Onyenwe, Mark Hepple, and Uchechukwu Chinedu. 2016. Améliorer la précision dâĂŹannotation dâĂŹun corpus Igbo par reconstruction morphologique et lâĂŹapprentissage basé sur la transformation. In *Atelier Traitement Automatique des Langues Africaines (TALAF). JEP-TALN 2016*, Vol. 11.
[21] Ikechukwu Ekene Onyenwe. 2017. *Developing Methods and Resources for Automated Processing of the African Language Igbo*. Ph.D. Dissertation. University of Sheffield.
[22] Ikechukwu E Onyenwe and Mark Hepple. 2016. Predicting Morphologically-Complex Unknown Words in Igbo. In *International Conference on Text, Speech, and Dialogue*, Vol. 9924. Springer, 206–214.
[23] Ikechukwu E Onyenwe, Mark Hepple, Uchechukwu Chinedu, and Ignatius Ezeani. 2018. A Basic Language Resource Kit Implementation for the Igbo NLP Project. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17, 2 (2018), 10.
[24] Braja Gopal Patra, Khumbar Debbarma, Dipankar Das, and Sivaji Bandyopadhyay. 2012. Part of speech (pos) tagger for kokborok. *Proceedings of COLING 2012: Posters* (2012), 923–932.
[25] Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing, Philadelphia, USA*, Vol. 1. 133–142.
[26] Navanath Saharia, Dhrubajyoti Das, Utpal Sharma, and Jugal Kalita. 2009. Part of speech tagger for Assamese text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, 33–36.
[27] Christer Samuelsson. 1994. Morphological tagging based entirely on Bayesian inference. In *Proceedings of the 9th Nordic Conference of Computational Linguistics (NODALIDA 1993)*. 225–238.
[28] Smriti Singh, Kuhoo Gupta, Manish Shrivastava, and Pushpak Bhattacharyya. 2006. Morphological richness offsets resource demand-experiences in constructing a POS tagger for Hindi. In *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, PA, USA, 779–786.
[29] Scott M Thede and Mary P Harper. 1999. A second-order hidden Markov model for part-of-speech tagging. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 175–182.
[30] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 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*. Association for Computational Linguistics, 173–180.

Received -; revised -; accepted -