A Comparative Study of Hidden Markov Model and Conditional Random Fields on a Yorùbá Part-of-Speech Tagging Task

Ikechukwu I. Ayogu*, Adebayo O. Adetunmbi†, Bolanle A. Ojokoh‡ and Samuel A. Oluwadare§
Department of Computer Science, Federal University of Technology
Akure, Ondo State, Nigeria.
*ayoguii@futa.edu.ng, †aoadetunmbi@futa.edu.ng, ‡baojokoh@futa.edu.ng, §saoluwadare@futa.edu.ng

***Abstract*—Parts-of-speech tagging, the predictive sequential labeling of words in a sentence, given a context, is a challenging problem both because of ambiguity and the infinite nature of natural language vocabulary. Unlike English and most European languages, Yorùbá language has no publicly available part-of-speech tagging tool. In this paper, we present the achievements of variants of a bigram hidden Markov model (HMM) as compared to the achievement of a linear-chain conditional random fields (CRF) on a Yorùbá part-of-speech tagging task. We have investigated the likely improvements due to using smoothing techniques and morphological affixes on the HMM-based models. For the CRF model, we defined feature functions to capture similar contexts available to the HMM-based models. Both kinds of models were trained and evaluated on the same data set. Experimental results show that the performance of the two kinds of models are encouraging with the CRF model being able to recognize more out-of-vocabulary (OOV) words than the best HMM model by a margin of 3.05 %. While the overall accuracy of the best HMM-based model is 83.62 %, that of CRF is 84.66 %. Although CRF model gives marginal superior performance, both HMM and CRF modeling approaches are clearly promising, given their OOV words recognition rates.**

***Index Terms*—Yoruba language, Part-of-speech tagging, Features, Bigram HMM, linear-chain CRF**

I. Introduction

Part-of-Speech (PoS) tagging, the predictive sequential labeling of words in a sentence is a challenging problem, both as a result of ambiguity and the infinite nature of natural language vocabulary which makes it impossible to explicitly list all possible occurrences of all words and context in a training data and or a lexical dictionary. PoS taggers with state-of-the-art performance have been developed for English and other privileged Romance languages. Asian languages have also received a fair amount of research attention. African languages which make up 30.2 % of the world languages [1], with the notable exception of Arabic, have yet to see much computational linguistic research. With the exception of some recent attempts such as [2] and [3] for Igbo and Yoruba languages respectively, there has not been any recorded effort at developing PoS tagging systems for any Nigerian language.

This paper assess the extent to which two smoothing techniques, as well as the use of affix-derived information can improve the performance of first-order hidden Markov model (HMM) based PoS tagger for Yorùbá, a Nigerian language as a first extension to the work presented in [3]. We also compare the performance of the HMM based models to a Conditional Random Fields (CRF) [4] based model of PoS tagger for Yoruba language. The other objective of this paper is to investigate the suitability of either an HMM based or CRF based model of PoS tagger in a data scarce scenario for Yorùbá language and by extension, other Nigerian languages where data resources is a challenge.

An important motivation for considering these algorithms for comparison is that they are language in-dependent. Further, both can be modeled to include carefully engineered language-specific features for performance gains. The two models also belong to the same family but are sharply different. While HMM is generative, CRF, an analogue of logistic regression is discriminative. As [5] points out, the conditional part of the HMM model is in fact a CRF.

In the course of this research, a number of challenges were encountered. First, there is no standard, publicly available PoS tag set for Yorùbá language. To make progress, we designed our research tagset by adapting the universal PoS tagset of [6]. We devised 17 tags that we consider coarse-grain but adequate for the primary purpose for which we are creating the PoS tagger. Further refinement is necessary and would subsequently be carried out in future work. Second there were no publicly available PoS-annotated Yoruba corpora/corpus. We created a research corpus composed with texts from religious genre only. We restricted the genre to religion because we preferred orthographically well-formed[1] text data which were only available to us in the religious texts.

The most successful taggers have been developed using a number of umbrella approaches including: rule-based [7] and stochastic approaches: n-gram-based, HMM-based [8,9,10,11,12], clustering [13,14,15], bayesian [16,17], cyclic dependency networks [18], maximum entropy based [19,20], neural network [21,22] and genetic [23] approaches. There are also a number of taggers that combine the strengths of two or more of the main stream approaches in a hybrid framework e.g. [24]. HMM has been in use for PoS tagging for over

978-1-5090-4642-3/17/$31.00 ©2017 IEEE

[1]Text with full compliments of the diacritic/tone marks

three decades [25] and has proven useful for solving the PoS tagging challenge. Though it has interesting success stories, e.g. the TnT tagging system [9], CRF, a more recent entrant has been shown to outdo it in the task of PoS tagging [26].

PoS tagging finds extensive use in many higher-level Natural Language Processing (NLP) problems. It has been shown to complement and be complemented by morphological analysis; it is a necessary pre-processing tool for machine translation and it serves as a veritable tool for developing large scale tagged corpus from which treebanks can be constructed. A good reference case is the use of the TAGGIT system [27] in tagging the Brown Corpus (with an accuracy of 77 %) from which the Penn treebank [28] was eventually built [29]. Further more, PoS taggers are deployed in information retrieval systems for detection of best suited index terms and in speech applications where it helps to disambiguate pronunciations [30].

The rest of this paper is organized as follows: Section II presents a background on the modeling approaches while Section III describes the development of PoS tagset and data used in the experiments. In Section IV, we describe our experiments and present the results. The paper concludes in Section V.

II. Background on Our Approach

In this section, we discuss the HMM and CRF models that we implement in this paper. HMM and CRF are both members of the Markov Random Fields family but while HMM is generative, CRF is discriminative. CRF models conditional probability of the output, given the input directly, avoiding distributions that are due to the input alone, i.e. CRF does not model prior distribution $p(w)$ of the observed word sequences. This feature makes it possible for CRF to explicitly model overlapping features in a manner that HMM is unable to do.

A. HMM model for PoS Tagging

Given a sentence consisting of a sequence of words, w_1^n, for which there exists a corresponding tag sequence, t_1^n, an HMM models the joint probability of the tag and word sequences, $p(t, w)$. For tractability, HMM makes two simplifying assumptions: first, it assumes that every given tag t_i is independent of all other tags except tag t_{i-1}; and secondly, it assumes that every observed word depends on its tag t_i. With these assumptions, a bigram HMM model for PoS tagging can be stated, following the Bayesian inference approach as follows:

$$\begin{aligned} t_1^n &= \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \\ &\approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} p(w_i|t_i) \prod_{i=1}^{n} p(t_i|t_{i-1}) \end{aligned} \tag{1}$$

The parameters (emission and transition probabilities) of the HMM-based models were estimated using the MLE technique from the training data. To find the best tag sequence, $t_1, t_2, ..., t_n$ for a given input sequence, $w_1, w_2, ..., w_n$, we used the Viterbi algorithm.

Given an input sentence $x_1, x_2, ..., x_n$, inference, the problem of finding the best tag sequence $t_1, t_2, ..., t_n$ for this sentence is defined as:

$$\begin{aligned} p(w_1, w_2, ..., w_n,\, t_1, t_2, ..., t_{n+1}) = \\ \prod_{i=1}^{n+1} \overbrace{\tau(t_i|t_{i-1})}^{bigram\ model} \prod_{i=1}^{n} \epsilon(w_i|t_i) \end{aligned} \tag{2}$$

where τ and ϵ are transition and emission parameters, respectively.

The Viterbi algorithm computes a truncated version of (2) following a dynamic programming approach. It uses *back pointers* values to keep track of the best tag sequence at each step up to the first k terms. The truncation is defined as a function γ:

$$\gamma(t_1, t_2, ..., t_k) = \prod_{i=1}^{k} \tau(t_i|t_{i-1}) \prod_{i=1}^{k} \epsilon(w_i|t_i) \tag{3}$$

B. Conditional Random Fields

Lafferty et al [4] were the first to apply CRF to the modeling of sequential pattern recognition problem. The CRF model we experiment in this paper is a linear-chain type. Since CRF models the conditional probability directly, given an observation sequence (here input sentence) for which there are some corresponding states (here output tags), the linear chain CRF conditional probability is defined as:

$$p(t|w) = \frac{1}{Z} \exp(\sum_{t=1}^{T} \sum_{i=1}^{F} \mu_i f_i(s_{t-1}, s_t, w, t)) \tag{4}$$

Where T is the set of t sequence of PoS tags and w is the sequence of corresponding words. F is the set of f_i is a feature functions and μ_i is the weight for each feature. f_i and μ_i are arbitrarily real-valued functions. The arguments to f_i are the observed word sequences w, previous state s_{t-1}, current state s_t and the current position in the chain t.

A feature function can be defined as follows:

$$\begin{aligned} f_i(s_{t-1}, s_t, w, t) = \\ \begin{cases} 1 & \text{if } s_t = x \text{ and } w_{t+1} = y \text{ and } s_{t-1} = z, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

The Z term is itself defined to keep the probability valid and is defined over the entire state sequence as:

$$Z = \sum_{s} \exp(\sum_{t=1}^{T} \sum_{i=1}^{F} \mu_i f_i(s_{t-1}, s_t, w, t)) \tag{6}$$

HMM and CRF have some useful similarities in terms of their model specifications and as stated in [5], the forward-backward and Viterbi algorithms used for HMM can be applied to CRF without modification by re-writing the CRF model into:

$$p(t|w) = \frac{1}{Z(x)} \prod_{t=1}^{T} \psi(s_{t-1}, s_t, w, t)) \tag{7}$$

Such that if $\psi(s_{t-1}, s_t, w, t) \equiv \sum_k \mu_k f_k(s_{t-1}, s_t, w, t)$, (7) then becomes:

$$p(t|w) = \frac{1}{Z(x)} \prod_{t=1}^{T} \sum_k \mu_k f_k(s_{t-1}, s_t, w, t)) \quad (8)$$

Hence, CRF computes $Z(x)$ instead of $p(x)$ computed by HMM. For the CRF model, we relied on the implementation of the CRF++ tool for the parameter estimation and inference.

C. Smoothing in the HMM Models

Smoothing techniques were devised to enhance the performance of algorithms that employ the Maximum Likelihood Estimation (MLE) technique for the estimation of probabilities based on context or history [31] which will usually be unable to recognize or classify a previously unseen event whose MLE will be zero. Smoothing techniques adjust the MLE probabilities in such a way that events with zero probabilities are given some little probability from the higher probability events, thus making the probability distributions more uniform [31]. We experimented with Laplace (L) and Kneser-Ney (KN) smoothing techniques in this paper.

1) Laplace Smoothing: The intuition of Laplace smoothing is to pretend that events actually occurs a little more than they have been observed. This is formalized by adding an additional count of one, in the basic setup, to the count of every event. Some notable shortcomings of additive smoothing have been shown to be reasonably overcome by adding a factor δ that is dependent on the events in the data set rather than just increasing the counts by one. We implemented this technique using (9)

$$p_L(t_i|t_{i-n+1}^{i-1}) = \frac{\delta + c(t_{i-n+1}^{i})}{\delta|V| + \sum_{t_i} c(t_{i-n+1}^{i})} \quad (9)$$

$|V|$ is the vocabulary size.

2) Kneser-Ney Smoothing: This method is built on the absolute discounting technique [32]. Contrary to absolute discounting, KN smoothing sets the lower-order n-gram probability to be proportional to the number of different context in which it has appeared in the training data [31]. that is, the backoff distribution of KN smoothing is based on the lower-order n-gram with highest context variability. We use an interpolated bigram form of KN defined in [33] thus:

$$p_{KN}(t_i|t_{i-1}) = \frac{max(c(t_{i-1}, t_i) - d, 0)}{c(t_{i-1})} + \gamma(t_{i-1}) p_c \quad (10)$$

where:

$$d = \frac{n_1}{n_1 + 2n_2}$$

$$\gamma(t_{i-1}) = \frac{d}{c(t_i - 1)} |t_{i-1} : c(t_{i-1}, t_i) > 0|$$

$$p_c = \frac{|t_{i-1} : c(t_{i-1}, t_i) > 0|}{\sum_{t_i} |t_{i-1} : c(t_{i-1}, t_i)\, 0|}$$

n_1, n_2 are the total of bigrams that occurred exactly once and twice respectively.

D. Tagging Unknown

A PoS tagging system must address two forms of potential failures: assignment of inappropriate PoS label due to the lexical ambiguity of the word and inability to correctly label words that were not seen at the training time. While the first form of problem could arise from inconsistent labeling in the training data or low frequency counts of each possibilities in the ambiguous case, the second problem is exacerbated when the training data is extremely small; such that the coverage of vocabulary space of the language is very poor. In this situation, the tagger would often encounter a new word. Even when the data is adjudged to be *large enough*, languages are productively dynamic; always producing new words. Hence, for any tagger to be practically useful, it must be robust at handling previously unseen words.

We use affix information to estimate the most probable tag of a previously unseen word. We use a method described in [34] but instead, our affixes are morphological affixes, not a sequence of n characters as in the original model. The model is defined as follows:

$$p(t|c_n) = \frac{p(t|c_n) + \theta p(t|c_{n-1})}{1 + \theta} \quad (11)$$

c_n is the affix context and $\theta = \sigma(c_n)$, the standard deviation in c_n Where an unknown word has both prefix and suffix, we estimated the probability for each using (11) and interpolated them linearly to obtain the final probability estimate.

III. Tagset and Data

A. Tagset

The set of PoS tags used for this work were created by us. This became necessary since there were no publicly available PoS tagset for Yorùbá. We created a minimally coarse tag set by adapting existing standard PoS tagsets and annotation convention for English. We adapted the universal PoS tagset of [6] which we adjudged simple and adequate for our immediate purpose. Though Petrov's tagset follows the EAGLEs [35] guideline, we however for the reason of widespread acceptability, re-code the tags into the form used by the very widely referred Penn Treebank tagset convention [28].

Furthermore, there was the evident need to include a few of the recommended category of the EAGLE guidelines into our tagset. Petrov's scheme designate a generic word class X that lumps a few other categories together; we however elect to unbundle this due to some compelling evidences in the our data. Specifically, VBG, INT, DEM, MD, EX and NNP sub-categories were added, bringing the tagset to 17 tags as described in Table I. These additions allow us to capture some very important scenarios in Yorùbá language.

TABLE I
POS TAGSET USED

PetrovTag	OurTag	Description
ADJ	JJ	Adjectives
ADP	IN	Adpositions
ADV	RB	Adverbs
CONJ	CC	Conjunctions
DET	DT	Determiner
NOUN	NN	Noun
NUM	CD	Number
PRON	PRP	Pronoun
PRT	RP	Particle
VERB	VB	Verb
X	FW	Foreign words
.	PUN	Punctuation
n/a	NNP	Proper Nouns
n/a	VBG	Verbal Nouns
n/a	INT	Interjection
n/a	DEM	Demonstrative
n/a	MD	Modal Aux
n/a	EX	Existential There

B. Data

The data was obtained from www.jw.org[2], cleaned and then carefully hand-tagged to obtain the training corpus. The set of 17 tags described in Table I were used and in the tagging process, we adopted the word/tag convention (e.g. ṣe/VB, gbé/VB, iṣẹ́/NN). The data we used in the experiments consists of 8,075 words. The tagging exercise was painstakingly done with guides from the Penn Treebank tagging guideline of [36] whenever a difficult tagging decision has to be made. It is important to reveal that upon carefully studying the data, tagset and the Santorini's guide, it became convincing that the guide is a good instrument with which we can safely work. However, there are a number of cases where the guide could not suffice. A few of these can be seen when the issue of adposition is examined vis-a-vis English and Yorùbá languages. It is therefore pertinent to state it that effort required to produce an accurately PoS tagged corpus of appreciable size and coverage is not a job a few linguists in a short time. The Penn treebank of a million word corpus took several years to produce. Thus, this a time for synergy between the computer scientists and the linguists in order to ameliorate the challenges faced by computational linguistic research on Nigerian languages.

IV. EXPERIMENT AND RESULTS

In this section, we describe our experimental settings and discuss the results obtained therefrom.

1) Experimental Setup: The HMM and CRF-based models were trained and evaluated on data set of identical composition and configuration to allow for comparison. The data set was shuffled and split randomly into training and test sets consisting 6,620 and 1,445 words respectively. The test set consists of 214 unknown words. We use accuracy, determined from the ratio of correctly tagged words to the total words in the test set, as an evaluation measure. First, we evaluated the bigram HMM model with each of the two smoothing techniques: HMM with Laplace (hmm+L) and HMM with Kneser-Ney (hmm+KN), then HMM with no smoothing but augmented with affix-derived information (hmm+AF), and finally, combined each of Laplace and Kneser-Ney smoothing method with affix-derived information (hmm+L+AF; hmm+KN+AF). In the CRF-based model, we experimented with context features that are similar to the history available to the HMM-based models, especially the affix-augmented model. The feature templates were specified according to the format supported by the CRF++ tool. We used features that captured immediate bigram tag history on both sides of the current word.

2) Results and Discussion: The results showing the performance of the various PoS tagging models we investigated are summarized in Table II and shown graphically in Fig. 1.

Judging from the perspective of both overall and unknown word accuracy, the CRF-based model simply outperformed ($p < 0.05$) the best HMM-based model, hmm+KN+AF, by a margin of 1.04 % and 2.05 % respectively. Among all the variants of the HMM-based model, hmm+KN+AF achieves the best result by correctly tagging 78.64 % of the unknown words, doing better than, hmm+AF with a margin of 3.02 %. The remaining two variants, hmm+L and hmm+KN with unknown word accuracy of 60.33 % and 69.67 % respectively are the poorest.

The average tag-wise precision and recall for the various models in Table II indicates a general performance improvements in the HMM-based model with hmm+KN+AF model achieving the overall best precision and recall of 95.12 % and 79.41 % respectively. These are however lower than the corresponding 97.19 % and 83.08 % respectively attained by the CRF model.

A careful examination of the performance of the various tagging models evidently indicates that combining smoothing with information derived from morphological affixes helps improve the performance of the bigram HMM model for Yorùbá PoS tagging, even in significantly limiting data scenarios even though the simple CRF-based model still maintains a marginal superior performance.

TABLE II
EXPERIMENTAL RESULTS

Model	Overall Acc	Unk Acc	Precision	Recall
hmm+L	80.36	60.33	0.8601	0.6120
hmm+KN	82.86	69.67	0.9012	0.6288
hmm+AF	83.01	75.62	0.9233	0.7011
hmm+L+AF	81.72	72.89	0.9348	0.7513
hmm+KN+AF	83.62	78.64	0.9512	0.7941
CRF	84.66	80.69	0.9719	0.9308

Although these results are promising and indicates a significantly improved performance over the 79.44 % accuracy we reported in [3], it is yet to achieve the state-of-the-art [37] but performs better than [38]. Further investigations are necessary

[2]we acknowledge the kind permission granted us to harvest data for our research by Jehova's Witnesses Mission in Nigeria

to improve on it. The results agree with the literature in terms of the superiority of CRF over HMM in sequence labeling task. That notwithstanding, it is clear that there exist a possibility to building a competitive Yoruba PoS tagger with relatively small data set using either technique.

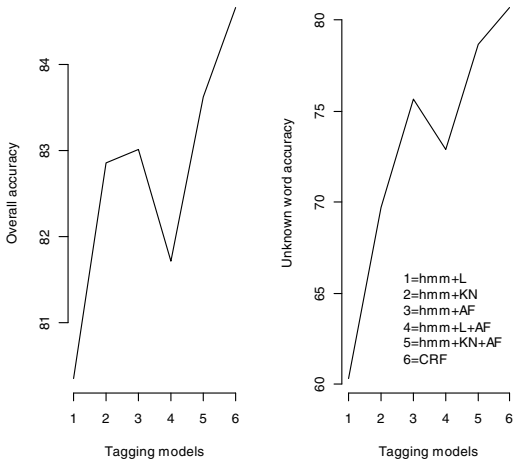

Fig. 1. Accuracy of the models: left - Overall accuracy, right - Unknown word accuracy.

V. Conclusion

This paper has described our preliminary experiments in the design of PoS tagger for Yoruba language. We have built and compared the accuracy of HMM-based taggers to that of a CRF-based PoS tagger using a small data set of 8,075 words. We investigated variations of HMM-based models where we explored the use of smoothing techniques: Laplace and Kneser-Ney smoothing, and also experimented with affix-based augmentation separately and combined each of the smoothing techniques from which preliminary results show that Kneser-Ney smoothed HMM that incorporates additional information from morphological affixes (prefixes and suffixes) performs best overall. We also built a simple linear-chain CRF model using features that capture simple contexts and orthographic features using the same data set configuration and composition as in the HMM-based models.

The overall results are encouraging but still below the state-of-the-art PoS tagging performance. The ongoing work focuses on step-wise improvement of the tagger accuracy on unknown words to such an extent where its output will require only minimal manual hand-correction to allow for speedy data set creation.

Acknowledgment

We acknowledge the kind permission of The Jehovah's Witnesses Mission in Nigeria to use data from www.jw.org for our NLP research.

References

[1] G. F. Simons and C. D. Fenning, "Ethnologue: languages of the world," twentieth edition, 2nd ed., Dallas: Texas, Online at: http://www.ethnologue.com, 2017.

[2] I. E. Onyenwe, and M. Hepple,"Predicting Morphologically-Complex Unknown Words in Igbo," in international conference on text, speech, and dialogue, pp. 206-214, 2016.

[3] I. I.Ayogu, A. O. Adetunmbi, B. A. Ojokoh and S. A. Oluwadare, "Developing a practical part-of-speech tagger for the yoruba language using a first-order hidden markov model," in proceedings of the 5th annual conference of the school of sciences, federal university of technology akure, 2017.

[4] J. Lafferty, A. McCallum and F. Pereira, "Conditional random fields: probabilistic models for segmenting and labelling sequence data," in proceedings of the international conference on machine learning, Williams, MA, 2001.

[5] C. Sutton and A. McCallum, "Introduction to conditional random fields," foudation and trends in machine learning, vol. 4, no. 4, pp. 267–373, 2012.

[6] S. Petrov, D. Das and R. McDonald, "A universal part-of-speech tagset," arXiv preprint arXiv:1104.2086, 2011.

[7] E. Brill, "Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging," Computational Linguistics, vol.21, no. 4,pp. 543-565, 1995.

[8] D. Cutting, J. Kupiec, J. Pederson and P. Sibun, "A practical part-of-speech tagger," in proceedings of the 3rd Conference on applied NLP, 1992.

[9] T. Brants, "TNT - a statistical part-of-speech tagger," in Proceedings of the sixth applied natural language processing conference, Seattle, 2000.

[10] B. Merialdo, "Tagging English text with a probabilistic model," Computational Linguistics, vol. 20, No. 2, pp. 155-171, 1994.

[11] M. Albared, N. Omar, M. Aziz, and M. Nazri, "Automatic part of speech tagging for Arabic: an experiment using bigram hidden markov model," rough set and knowledge technology, pp. 361-370, 2010.

[12] S. Dandapat, and S. Sarkar, "Part of speech tagging for Bengali with hidden markov model," in proceeding of the NLPAI Machine Learning Competition, 2006.

[13] A. Clark, "Combining distributional and morphological information for part of speech induction," in proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, vol. 1, pp. 59-66, 2003.

[14] A. Clark, "Inducing syntactic categories by context distribution clustering," in proceedings of the 2nd workshop on learning language in logic and the 4th conference on computational natural language learning, Association for Computational Linguistics, vol. 7, pp. 91-94, 2000.

[15] B. Can and S. Manandhar, "Unsupervised learning of morphology by using syntactic categories," in CLEF (Working Notes), 2009.

[16] C. Christodoulopoulos, S. Goldwater and M. Steedman, "A bayesian mixture model for part-of-speech induction using multiple features," in proceedings of the conference on empirical methods in natural language processing, ACL, pp. 638-647, 2011.

[17] S. Goldwater, and T. Griffiths, "A fully bayesian approach to unsupervised part-of-speech tagging," in proceedings of the 45th annual meeting-association for computational linguistics, vol. 45, pp.744–852, 2007.

[18] K. Toutanova, D. Klein, C. D. Manning and Y. Singer, "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, vol. 1 pp. 173-180, 2003.

[19] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in EMNLP vol. 1, pp. 133-142, 1996.

[20] G. De Pauw, P. W. Wagacha and D. A. Abade, "Unsupervised induction of Dholuo word classes using maximum entropy learning," in Getao, K. and E. Omwenga Eds. proceedings of the 1st international conference in computer Science and ICT. University of Nairobi, 2007.

[21] H. Schmid, "Part-of-speech tagging with neural networks," in proceedings of the 15th conference on computational linguistics, ACL, vol. 1 pp. 172-176, 1994.

[22] M. Frodl, "Part-of-Speech Tagging Using Neural Networks," PhD thesis, Doctoral dissertation, Masarykova univerzita, Fakulta informatiky, 2014.

[23] B. B. Ali and F. Jarray, "Genetic approach for Arabic part of speech tagging," international journal on natural language computing, vol. 2, pp. 1-12, 2013.
[24] S. Dandapat, S. Sarkar, and A. Basu, "A hybrid model for part-of-speech tagging and its application to Bengali," in international conference on computational intelligence, pp. 169-172, 2004.
[25] M. Fruzangohar, T. A. Kroeger and D. L. Adelson, "Improved part-of-speech prediction in suffix analysis," plos ONE vol. 8, no. 10, pp. 1–6, 2013.
[26] B. R. Shambhavi and K. P. Ramakanth, "Kanada part-of-speech tagging with probabilistic classifiers," international journal of computer applications, vol. 48, no. 17, pp. 26–30, 2012.
[27] B. B. Greene and G. M. Rubin, "Automatic grammatical tagging of English. technical report," Department of Linguistics, Brown University, Providence Rhode Island, 1971.
[28] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: the Penn Treebank," computational linguistics, vol. 19, pp. 313-330, 1993.
[29] W. Francis, and H. Krucera, Frequency analysis of English usage: lexicon and grammar, Houghton Mifflin, Boston, 1982.
[30] A. Votilainen, Part-of-Tagging. In: Ruslan Mitkov (Eds) Oxford Handbook of Computational Linguistics. Oxford University Press, Great Clarendon Street, NY, pp. 109–111, 2003.
[31] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," computer speech and language, vol. 13, pp. 359–394, 1999.
[32] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in proceedings of the IEEE international conference on acoustics, speech and signal processing, Detroit, MI, vol. 1, pp.181–184, 1995.
[33] D. Jurafsky and H. Martin, Speech and Language Processing: An introduction to language processing, computational linguistics and speech recognition. 2nd Edition. Prentice Hall, New Jersey, pp. 97–113, 2009.
[34] C. Samuelson, "Handling sparse data by successive abstraction," in proceedings of COLING, Copenhagen, Denmark, pp. 895–900, 1996
[35] G. Leech and A. Wilson, "Recommendations for the morphosyntactic annotation of corpora," EAGLES Report, EAG-TCWG-MAC/R, 1996.
[36] B. Santorini, "Part-of-speech tagging guidelines for the Penn treebank project. Technical report MS-CIS-90-47, Third Revision, Second printing (February, 2005), Department of Computer and Information Science, University of Pennsylvania, 1990.
[37] C. D. Manning, "Part-of-speech tagging from 97% to 100%: is it time for some linguistics?" in proceedings of the 12th international conference on computational linguistics and intelligent text processing, pp. 171-189, 2011.
[38] F. M. Hasan, N. UzZaman and M. Khan, "Comparison of different part-of-speech tagging techniques (n-gram, HMM, and Brill's tagger) for Bangla" in Advances and Innovation Systems, Computing Sciences and Software Engineering, pp. 121-126, 2007.