PART OF SPEECH TAGGING FOR AMHARIC

Binyam Gebrekidan Gebre
Centre Tesnière
Université de Franche-Comté, France
Research Institute in Information and Language Processing
Wolverhampton University, UK
binyamkidan@yahoo.com

Abstract

This paper discusses theoretical and practical POS tagging issues with the view to improving POS tagging performance for Amharic, which was never above 90%. Knowledge of Amharic morphology, the given annotated data and the tagging algorithms have been examined and shown to play critical roles in the final performance result. With the experiments carried out using state-of-the-art machine learning algorithms, POS tagging accuracies for Amharic have crossed above the 90% limit for the first time. The reasons for such relatively higher performance have come from three factors: usage of partially cleaned version of a corpus, selection of the most informative features, resulting from morphological study of the language and application of parameter tuning, resulting from understanding the tagging algorithms and experimenting with them.

Key-words

Language; Semitic; Amharic; Part of speech; POS; Tagging; HMM; CRF; SVM; Brill; TnT; NLTK.

Résumé

Cet article discute les problèmes théoriques et pratiques d'étiquetage des parties du discours en vue d'améliorer son efficacité pour l'amharique, qui n'a jamais été supérieur à 90%. Connaissance de la morphologie de l'amharique, les corpus d'apprentissage fournis et les algorithmes d'étiquetage ont été examinées et montrées à jouer un rôle crucial dans le résultat final obtenu. Avec les expériences réalisées en utilisant des algorithmes d'apprentissage de l'état de l'art, les précisions d'étiquetage pour l'amharique ont franchi la limite de 90% pour la première fois. Les raisons pour la performance relativement plus élevée sont venues de trois facteurs: l'utilisation d'une version d'un corpus partiellement nettoyé, la

sélection des éléments les plus instructifs, résultant de l'étude morphologique de la langue et le réglage des paramètres, résultant de la compréhension des algorithmes d'étiquetage et d'expérimentation avec eux.

Mots-clés

Langue; Sémitiques; Amharique; La partie du discours; POS; Etiquetage; HMM; CRF; SVM; Brill; TNT; NLTK

1. Introduction

Much of the research in natural language processing has been dedicated to resource-rich languages like English and French. African languages have, however, received far too little attention. In fact, most of them are being spoken by less and less people. One exception that is seeing an increase in use and number of speakers is Amharic, a language that is mainly spoken in Ethiopia. Currently, it has an estimated 30 million speakers (Gamback et al., 2009), which puts it in second position as the most spoken Semitic language in the world (after Arabic).

The number of speakers of the language is on the rise for two reasons. First, it is the working language of the federal democratic republic of Ethiopia, a country with more than 85 million people (CIA, 2010). Second, unlike most other African languages, Amharic is a written language with its own alphabet and written materials, actively being used every day in newspapers and other media outlets.

However, even under these favorable conditions, Amharic has been one of the under-resourced languages both in terms of electronic resources and processing tools. Recently, however, there have been independent attempts to develop them. One outcome of such an attempt is the publicly available medium-sized part-of-speech-tagged news corpus (Demeke and Getachew, 2006) and a morphological analyzer (Gasser, 2009). The availability of these resources has encouraged researchers to process the language by adapting and applying different NLP models that have proven effective for analyzing English and other most-studied languages.

One basic task in natural language processing is part-of-speech tagging or POS tagging for short. It is the process of assigning a part-of-speech tag like noun, verb, pronoun, preposition, adverb, adjective or other lexical

class markers to each word in a text. POS tagging is not useful by itself but it is generally accepted to be the first step to understanding a natural language. Most other tasks and applications heavily depend on it.

In addition to that, POS tagging is seen as a prototype problem because any NLP problem can be reduced to a tagging problem. For example, machine translation can be seen as the tagging of words in a given language by words of another language; speech recognition can be seen as the tagging of signals by letters and so on. In general, the input-output relationship can be as complex as sequences, sets, trees and others that can be imagined. POS tagging represents the simplest of these problems.

At first sight, the solution to this POS tagging problem may seem trivial, but it is actually very hard. There is no known method that solves the problem with complete accuracy for any language. The reason for this is partly related to inconsistencies of our understanding of categories of words. Even trained human annotators do not agree as to the category of a word 3-4% of the times (Marcus et al., 1993). The other reason arises from language ambiguities and the ineffectiveness of the resolving methods.

Language expressions are ambiguous and computers do not have the commonsense and the world knowledge that humans have when they communicate. For example, **I made her duck** can have the following meanings (Jurafsky et al., 2000).

1. I cooked waterfowl for her.
2. I cooked waterfowl belonging to her.
3. I created the (plaster?) duck she owns.
4. I caused her to quickly lower her head or body.
5. I waved my magic wand and turned her into undifferentiated waterfowl.

These different meanings are caused by a number of ambiguities. The first one is part of speech ambiguity. **Duck** can be a verb or a noun and **her** can be a dative pronoun or a possessive adjective. The other ambiguities are related to semantics and syntax (i.e. make can mean cook or create and it can take one or two arguments).

To a human being, the intended meaning of the above sentence is clear depending on the circumstances but for a computer it is far from obvious. Therefore, the purpose of tagging is to give the computer as much information and knowledge as necessary to enable it to assign each word the correct tag as used in the given context.

There are three approaches to solving this tagging problem based on two fundamental concepts: rules and statistics. Rule-based taggers use handcrafted linguistically-motivated rules. Stochastic taggers, by contrast, use probabilistic mathematical models and a corpus. The third approach combines the best of both concepts. None of them is perfect for all languages and for all purposes. The relevance and effectiveness of each approach depends on the purpose and the given language.

This paper applies state-of-the-art tagging methods and tests their effectiveness on Amharic, a morphologically-rich language. Section 2 discusses previous work on Amharic POS tagging. Sections 3 and 4 present and discuss the tagging models and the features used. Sections 5 through 7 describe the research methodology. Sections 8 through 10 present, analyze and summarize the results obtained.

2. Previous Work

NLP research on Amharic has started fairly recently and has been constrained by lack of linguistic resources and an authoritative body to define and develop them. Unlike Arabic and Hebrew, with which it shares most of its characteristics, Amharic does not yet have a Treebank. Even so, NLP researchers from native speakers to non-speakers have shown interest in the language and developed prototypes by applying some of the state-of-the-art tagging models (Getachew, 2001; Adafre, 2005; Gamback et al., 2009; Tachbelie and Menzel, 2009).

Getachew (2001) is the pioneer for Amharic POS tagging experiments. He developed a tagging prototype using Hidden Markov models, which he trained and tested on a text of one page. His contribution also included the definition of a tagset of 25 that has served as a basis for the tagsets used by subsequent researchers.

Adafre (2005), who did the next POS tagging experiment for Amharic, revised Getachew's tagset and reduced it to ten. As there were no POS annotated data at the time, Adafre collected five news articles and manually annotated them, which he then used for both training and testing of a stochastic model based on conditional random fields (Lafferty, 2001).

He obtained an average accuracy of 74% on a 5-fold cross-validation where one file is used for testing and the other files for training. The main reason for the poor performance is the small size of the dataset. 80% of the words in the test files consist of unseen words. From this result and successful experiences in other experiments for large datasets, it became clear that Amharic POS-annotated data is necessary to achieve performances comparable to the state-of-the-art results.

In 2006, a medium-sized corpus of reportedly 210,000[1] tokens annotated with parts of speech was released (Demeke and Getachew, 2006). The corpus consists of 1065 news articles collected from Walta Information Center (WIC), a private news agency located in Addis Ababa. It is tagged with 31[2] parts of speech and is publicly available on the Internet. This corpus has been a useful resource for the recent experiments on Amharic POS tagging.

Using the WIC corpus, Gamback et al. (2009) and Tachbelie and Menzel (2009) applied different tagging methods and obtained worse performances than the state-of-the-art results for Arabic or English.

Gamback conducted detailed experiments using TnT (Brants, 2000), SVMTool (Giménez and Marquez, 2004) and Mallet (McCallum, 2002) on three different tagsets. The overall accuracies using the ELRC[3] tagset are 85.56% for TnT, 88.30% for SVM and 87.87% for MaxEnt. Similarly, Tachbelie and Menzel (2009) also conducted similar experiments using TnT and SVMTool models with overall accuracies of 82.99% for TnT and 84.44% for SVM. For both sets of experiments, the best performances are

[1] Actual counting reveals a number less than that
[2] 30 is reported, actually there are 31 tags
[3] Ethiopian Languages Research Center

achieved by SVM but Gamback's SVM performs better (88.30% against 84.44%).

Those poor performances (compared to those of English or Arabic) can be explained by four reasons. First, the corpus used is small; it is one-sixth of the size of the WSJ corpus. Second, the taggers use no more knowledge source than a pre-tagged training corpus. Third, the quality of the corpus is poor. Tagging errors and inconsistencies are considerable in the corpus. Fourth, little parameter tuning of the algorithms was done to suit the WIC corpus.

Except for Adafre (2005), who used dictionaries of affixes and some 15,000 entries (Aklilu, 1987) with their POS tags (Noun, Verb, Adjectives, Adverb, and Adposition), all other previous POS experiments for Amharic used language independent features.

For Amharic, one feature that is important and not included by previous experiments is the vowel patterns embedded in words. For example, *kebad* and *kelal* are adjectives and share the same **e, a** vowels. Verbs also show similar vowel patterns. *manbebu* (that he read), *madregu* (that he did), *mabedu* (that he became mad), etc all share **a, e, u** vowel patterns. Another feature that may prove useful is the radicals (the consonants in the words). For example, *sebere* (he broke), *sebro* (having broken (he)), *sebra* (having broken(she)) can be reduced to just the radical *sbr* and be treated as a verb. Both the vowel pattern and the radical features have the advantage of reducing data sparsity problem. Any language modelling technique would perform better by capturing them.

The right features are not sufficient for performance improvement if the quality of the corpus is poor. The WIC corpus has significant errors and tagging inconsistencies. This problem has been acknowledged by researchers who worked on it and they have made efforts to correct some of them. For example, Gamback's experiments were done on a partially corrected WIC corpus. The corrections included tagging non-tagged words, removing double tags, treating consistently "" and '/' as punctuation, retagging some wrongly tagged words and some spellings errors. However, they acknowledge that tagging inconsistencies related to time and number

expressions had been left as they were. Therefore this type of error and others left unnoticed have contributed to the relatively poor performance.

This paper will attempt to improve performance by doing three things. The first one is based on cleaning the corpus. This step is crucial and will determine the performance of any method. The second thing involves feature selection. The usual features used for POS tagging are used. In addition, however, the vowel patterns and the radicals, which are characteristics unique to Semitic languages, are also included. The third is by applying the state-of-the-art tagging machine learning algorithms and doing necessary parameter tuning as much as possible. Algorithms used are based on conditional random fields, support vector machines, Brill tagging and HMM.

All of these things combined have contributed to the most accurate part of speech tagger ever reported for Amharic.

3. Tagging Models

Part of speech tagging can be done either using handcrafted linguistically-motivated rules (Greene and Rubin, 1971; Voutilainen, 1995; Cardey and Greenfield, 2003) or by stochastic methods (Stolz et al., 1965; Bahl and Mercer, 1976; Marshall, 1983; Garside, 1987; Church, 1988; Derose, 1988; Brants, 2000; Giménez and Marquez, 2004; Tsuruoka et al., 2005). It is also done by combining the best of both rule-based and stochastic methods (Brill, 1992; 1995; Garside and Smith, 1997). While rules are specific for languages, stochastic or machine learning based tagging methods are independent of languages.

For our POS tagging experiments, we have applied CRF++ (Kudo, 2007), LIBSVM (Chang and Lin, 2001), Brill (Brill, 1992; 1995; Garside and Smith, 1997) and TnT (Brants, 2000). Python implementations of Brill and TnT (HMM-based) provided in NLTK (Loper and Bird, 2002) have been used. Conditional random fields and support vector machines are widely used classification or sequencing labeling machine algorithms for a variety of applications.

4. Feature Selection

Each of the aforementioned tools, albeit critical, is only the last step of the stochastic tagging process, which usually involves, either explicitly or implicitly, the following tasks: a) tokenization b) feature extraction c) tagging.

While tagging can be done by using standard language independent classification algorithms, tokenization and feature extractions need to be adapted to the nature of the given language. For tokenization, a consistent rule has been applied: split Amharic text into sentences on double colon, the sentences into words on space and treat punctuation marks as separate tokens. Unlike for English, the resulting words for Amharic usually represent the concatenation of morphemes each capable of having its own POS tag (e.g.: NP for Noun attached with a Preposition).

After tokenization follows feature extraction. The following features are used in POS tagging each token.

- the current word, the previous/following word, the word before/after the previous/following word {String}
- prefixes and suffixes of length of up to five {String}
- vowel patterns(current word) {String}
- radicals (current word){String}
- is punctuation(current word){True, False}
- has punctuation(current word){True, False}
- is alphabetic(current word) {True, False}
- is alphanumeric(current word){True, False}
- is digit(current word){True, False}
- has digit(current word){True, False}

- has e suffix(previous word){True, False}

The novel features are the vowel patterns and radicals. The vowel patterns have been shown to have linguistic importance in Gebre (2009).

5. The Corpus and the Tagset

The medium-sized POS tagged corpus for Amharic consists of 1,065 news articles (approximately 210,000 tokens) collected from Walta Information Center (WIC), in the period that spans from 1998 to 2002.

The ELRC tagset, used to tag the corpus, is based on 11 basic tags, most of which have further been refined to provide more linguistic information, thus increasing the tagset to 31. For example, the tags for **N**ouns are **VN** (***V**erbal **N**oun*), **NP** (***N**oun with **P**reposition*), **NC** (***N**oun with **C**onjunction*), **NPC** (***N**oun with **P**reposition and **C**onjunction)* and **N** (for any other **N**oun). There are similar patterns for **V**erbs, **ADJ**ectives, **PRON**ouns and **NUM**erals. Additional tags under the verb category are tags **AUX (**for **AUX**iliary) and **VREL** (for **REL**ative **V**erbs). Numerals are divided into cardinals and ordinals represented by the **NUMCR** and **NUMOR** tags**.** The rest of the tags are **PREP** for *prepositions*, **CONJ** for *conjunctions*, **ADV** for *adverbs*, **PUNC** for *punctuation*, **INT** for *interjection* and **UNC** for unclear (difficult to classify).

6. Cleaning the Corpus

Any POS tagging method cannot be expected to have less error rates than the fraction of errors or discrepancies introduced by the annotators. Since our objective is to improve performance, the best strategy is to start with a cleaned version of the WIC corpus before designing the tagging method. With this view, an effort has been made to correct as many errors and inconsistencies as possible.

Even though 210k tagged tokens were reported for the WIC corpus, the actual number without cleaning the corpus is 200545, a difference of 9455 (about 5%). Part of the reason for this discrepancy is caused by tagging errors. The errors are such that some tokens have multiple tags and other tokens do not have tags at all, which makes us think that they constitute

multi-unit tokens. Some punctuation marks (quotation marks and forward slash) are considered as part of some tokens (e.g. "bodigardna" <NC>). This kind of error accounts for almost half of the errors. Other errors, probably associated with typing, include some tags without angle brackets <TAG> and so can be mistaken for tokens. Eight headlines and one sentence are not tagged at all or they are just tagged as multi-word units. In reality, however, the tag is the correct tag of only the last word in the headline or sentence.

Besides the aforementioned errors, there are serious inconsistencies with respect to what constitutes a word and what tags should be assigned for a word under the same contexts. For example, words in collocations are sometimes treated as one unit and other times as separate words.

Correction of the simple errors mentioned earlier resulted in an increase of the total number of token-tag pairs from 200545 to 200766 (an increase of 0.11%). Correcting the inconsistencies proved more laborious and required more sophisticated techniques at times.

The first inconsistency problem is related to tokenization of time, number and name expressions. In some cases, the expressions are considered as independent tokens. In other cases, under similar conditions, they are tagged as multi-word tokens. To solve this problem in a principled manner, the expressions were tokenized on space and given the tag that together they had in the first place. This has the advantage of decreasing data sparsity (improving the language model). One trap that we should avoid falling into is that when prepositions and conjunctions are attached with the beginning or last words in the multi-word tokens, then the middle words should have the form of the tags in which the preposition and/or conjunction tags are stripped off. This is because, in Amharic, prepositions and conjunctions are attached with the beginning or last words of the multi-word tokens. Applying the suggested method increased token-tag pairs from 200766 to 206929 (an increase of 3.07%).

The second inconsistency problem is related to tokens receiving multiple tags under the same conditions. An effort has been made to identify and correct some of them. One technique that we have used is to list all the tokens and the frequency of their association with each tag they are

assigned. A closer examination of this list for a given word reveals that some tags are wrongly assigned.

For example, the punctuation mark (") has been tagged as PUNC correctly 97% of the times. In the rest 3%, it has been assigned the wrong tags. Similarly, the punctuation mark (,) is correctly tagged as PUNC in 99.8% times, but it is also tagged incorrectly in few other instances. Such errors are not limited to few cases, but in fact, most frequent multi-tag tokens have some extra tags assigned incorrectly infrequently.

Some of the aforementioned tagging inconsistencies have been corrected. About 552 tokens have been correctly retagged as prepositions and 893 tokens as nouns, verbs and their variants. Similarly, about 980 numbers and punctuation marks have also been correctly tagged. For multi-tag tokens, token-tag pair with frequency of appearance of one has been replaced by the tag with the highest frequency of at least 10 (double the average frequency of each word in the WIC corpus). With this method, 1209 tokens have been retagged with the tag of the highest frequency.

Table 1: 10-fold Cross-validation Data

Folds	Training	Test	Known	Unknown
1	186406	20523	17927	2596
2	185832	21097	18581	2516
3	186724	20205	18043	2162
4	186154	20775	18458	2317
5	186500	20429	18081	2348
6	186719	20210	18108	2102
7	185788	21141	18795	2346
8	185372	21557	19132	2425
9	186615	20314	18085	2229
10	186251	20678	18444	2234
Average	186236	20693	18365	2328

7. Training and Test Data

The partially cleaned WIC corpus consists of 8067 sentences. The corpus is divided into training and test data. The training data is the 90% portion of the data and the remaining 10% is the test data. To get a more reliable result, a 10-fold cross-validation is applied. Table 1 shows the number of

tokens in the training and test sets. The numbers of tokens in each fold is not the same because the partition is made at a sentence level. Except for the last fold, which has 804 sentences, each fold has 807 sentences. Each fold is also divided into known tokens and unknown tokens. About 11.25% of the test tokens are unseen in the training data.

8. Results

CRF++, LIBSVM, Brill and TnT have all been applied for our POS tagging experiments using the WIC corpus. For good comparison, CRF and SVM are treated together as they use exactly the same features. Similarly, Brill and TnT are also treated together as they are similar in terms of their dependence on neighboring words/tags and their mechanisms in handling unknown words.

8.1. Baselines

The simplest tagger that can serve as a baseline in Amharic part of speech tagging is to tag all new tokens as **N**, which is the most frequent tag in the WIC corpus. This achieves an accuracy of about 36%. This is too low to be used as a baseline as most algorithms have much higher accuracies. Another baseline is assigning the most frequent tag of every word seen in the 90% of the training corpus and assigning **N** to unseen words. This achieves about 81% accuracy on the remaining data (10%). All the algorithms applied in this paper achieve much higher accuracies than 81%.

8.2. CRF++ and LIBSVM

Both CRF and SVM have been trained and tested on the same dataset using exactly the same features. Parameters have also been selected for both. The critical parameter in both cases is the penalty parameter C. A too small value for C causes underfitting and a too large value causes overfitting. In other words, a small value for C will allow a larger number of training errors, while a large value will minimize training errors. For CRF, C = 0.05 and for SVM, C = 0.5 have been experimentally found to give higher accuracies. These smaller C values have been chosen for a good reason. The WIC corpus has a number of training errors and so using a larger C can only make the algorithm learn the errors too. A smaller C

value, however, ignores some of the errors. An additional critical parameter for SVM is the kernel type. Here, the LINEAR[4] kernel has been found to give higher accuracies.

On a 10-fold cross-validation, CRF achieves an average accuracy of 90.95%, while SVM achieves 90.43% under exactly the same conditions. The difference might seem too little, but a statistical significance test proves otherwise. This difference of 0.52 can happen by chance once in thousands, which is less than 0.05 (the conventional level of significance).

The average accuracies for both algorithms on known and unknown tokens are shown in table 2. As can be seen from the table, SVM achieves a slightly higher average accuracy of 80.59% than SVM (80.52%) on unknown tokens, which may lead to the conclusion that SVM generalizes better. However, a statistical significance test shows that the difference is too little to reach that conclusion ($0.461 > p = 0.05$). On the other hand, CRF achieves relatively higher on known tokens which explains its slight overall higher accuracy.

Table 2 : Average Accuracies on 10-fold Cross-validation (in %)

Algorithm	Known	Unknown	Overall
CRF	92.28	80.52	90.95
SVM	91.67	80.59	90.43
Brill	91.90	51.98	87.41
TnT	91.54	51.98	87.09

[4] Other kernels tried did not improve performances much and required more parameter space searching

8.3. Brill and TnT

The Brill and TnT taggers achieve average accuracies of about 87%, which is 3% less than CRF and SVM. On average, Brill achieves an accuracy of 87.41%, 0.32% higher than TnT (highly statistically significant) but both achieve the same average accuracy on unknown tokens. This is to be expected as they have both been designed in these experiments to use the same techniques for handling unknown words. The same simple regular expression tagger has been used in both cases.

This regular expression tagger assigns tags based on affixes in Brill as part of the initial state tagger and in TnT as part of the unknown words tagger. The reason Brill performs better on average is because it has significant higher performance on tagging known tokens (91.90% against 91.54%). This is also to be expected given that TnT depends on using the statistics of previous two tags and the association of words and tags, while Brill uses much more information from the left and the right neighboring tags and words.

The Brill tagger has two important parameters: the maximum number of rules and the minimum score. The values for these parameters have to be chosen carefully by experimenting. Increasing and decreasing both parameters too much decreases performance. For example, for minimum score of 3 and maximum number of 50 rules, the average accuracy is 87.39% and for the minimum score of 15 and maximum number of 200 rules, the same performance is obtained. Intermediate values usually have better performances. The highest accuracy (87.41%) is obtained with minimum score of 6 and a maximum number of 50 rules. Other combinations with the same performance have more rules or lower minimum score, which makes the training slower and the tagger more complex.

One of the interesting features of Brill tagging is that we can see which rules are contributing the most to improving the tagging accuracies. Brill tagging has transformation templates which examine the neighboring words and tags. One of the interesting rules it formed from the WIC corpus is related to tagging the Amharic word *adis* (= new), which is usually used as adjective and is tagged as such by the initial stage tagger. When it is,

however, followed by *abeba*[5] (=flower), it should be tagged as noun. Brill has been able to learn that rule automatically.

9. Error Analysis

Confusion matrices for both CRF and SVM show that confusions between nouns and other tags account for most of the errors in both tagging models. Table 3 shows confusion matrix for CRF. For lack of space, the equivalent table for SVM is not shown. More than 44% of the errors in CRF resulted from taking non-nouns and their variants (ie: NP,NC, and NPC) as noun families. The corresponding percentage for SVM is a little less (39.05%). From these confusions, the bigger portions are taken by confusions between nouns and adjective families, which account for more than 19% in CRF and 18% for SVM. In both tagging models, non-noun families are taken to be noun families more than the other way round. For example, in CRF, 7.7% of the error rates resulted from confusing **ADJ**s for **N**s, whereas 4.51% resulted from confusing **N**s for **ADJ**s. The corresponding values for SVM are 6.59% and 4.89%. This should not come as a surprise if we closely examine the morphology of the words. The same affixes are shared by noun families and most of the non-noun families.

Table 3: Confusion Matrix for CRF in Percentage

	ADJ	N	NP	NC	V	VP	VREL	Other
ADJ	0	7.7	1.09	0.08	0.14	0.36	0.49	2
ADJP	0.23	0.08	4.88	0	0	0.51	0.3	0.55
ADV	0.25	0.94	1.36	0.07	0.5	0.58	0.07	0.66
CONJ	0	0.26	0.38	0.02	0.1	0.05	0	0.41
N	4.51	0	3.06	1.66	0.69	0.32	0.09	2.61
NC	0	1.32	0.15	0	0.02	0.01	0.01	0.87

[5] 'adis abeba' is the capital city of Ethiopia

NP	0.19	1.71	0	0.2	0.08	3.42	1.22	4.44
NPC	0	0.01	1.23	0.41	0	0.08	0.01	0.46
PRONP	0.02	0.05	0.83	0	0	0.04	0	0.47
PREP	0.06	1	0.26	0.05	0.03	0.02	0.04	0.41
PRON	0.13	0.3	0.19	0	0	0.1	0.03	1.17
V	0.13	1.82	0.14	0.04	0	1.56	0.44	0.86
VN	0.01	2.19	0.21	0.23	0.08	0.1	0.1	0.02
VP	0.04	0.56	6.06	0.07	2.64	0	7.91	2.36
VPC	0	0.02	0.05	0.35	0	0.73	0.06	0.79
VREL	0.05	0.09	1.14	0.02	0.5	3.75	0	0.5
Other	0.12	0.7	0.5	0.81	0.48	0.32	0.22	2.45

A noun phrase that consists of only the head noun gets affixes such as prepositions, definite article, and the case marker. However, if a noun phrase contains prenominal constituents such as adjectives, numerals, and other nouns, then the stated affixes appear on the prenominal constituents. This phenomenon blurs the morphological distinctions that would otherwise have been useful for distinguishing nouns against their constituents including adjectives. That is why **ADJ**s are mistaken for **N**s more than the other way round in both CRF and SVM.

The largest error percentage resulted from confusion between **VP**s (verb with preposition) and **VREL** (verb relative). In CRF, mistaking **VP**s for **VREL** accounts for 7.91% of the total errors. The corresponding value for SVM is 7.84%. Closer examination of the results shows that the POS taggers did not actually predict the wrong tags in some cases. The problem is that the predictions were made against wrongly assigned tags in the test set. In fact, some of the confusions between some pairs can be shown to be

errors in the test set. For example, *yehonu* (= that are) has been tagged as both **VP** and **VREL** in the same test under similar conditions, making either prediction wrong for the other.

10. Conclusion

Knowledge of Amharic morphology, the given annotated data and the tagging algorithms have been examined and shown to play critical roles in the final performance result. With the experiments carried out on WIC corpus, POS tagging accuracies for Amharic have crossed above the 90% limit for the first time.

The improvement in performance is attributed to a combination of three factors. First, the POS tagged corpus (WIC) has been cleaned up to minimize the pre-existing tagging errors and inconsistencies. Second, the vowel patterns and the roots, which are characteristics of Semitic languages, have been used to serve as important elements of the feature set. Third, state-of-the-art of machine learning algorithms have been used and parameter tuning has been done whenever necessary and as much as possible.

Acknowledgment

The author would like to thank his supervisors Prof. Sylviane Cardey and Prof. Ruslan Mitkov for their support and guidance. Special thanks also go to Dr Peter Greenfield for his constructive comments.

References

ADAFRE, S. F. (2005), "Part of speech tagging for Amharic using conditional random fields", in '*Semitic '05: Proceedings of the ACL Work-shop on Computational Approaches to Semitic Languages'*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 47–54.

AKLILU, A. (1987), "Amharic-English Dictionary", Kuraz Publishing Agency.

BAHL, L. R. and MERCER, R. L. (1976), "Part of speech assignment by a statistical decision algorithm", in *'Proceedings IEEE International Symposium on Information Theory'*, pp. 88–89.

BRANTS, T. (2000), "TnT- a statistical part-of-speech tagger".

BRILL, E. (1992), "A simple rule-based part of speech tagger".

BRILL, E. (1995), "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging", *Computational Linguistics* 21, 543–565.

CARDEY, S. and GREENFIELD, P. (2003), "Disambiguating and tagging using systemic grammar", *in Proceedings of the 8th International Symposium on Social Communication*, pp. 559–564.

CHANG, C.-C. and LIN, C.-J. (2001), "LIBSVM: a library for support vector machines", Software available at *http://www.csie.ntu.edu.tw/cjlin/libsvm*.

CHURCH K. W. (1988), "A stochastic parts program and noun phrase parser for unrestricted text", *In Proceedings of the Second Conference on Applied Natural Language Processing*, pp. 136–143.

CIA (2010), 'The world fact book - ethiopia', [Accessed in May 2010]. URL: *https://www.cia.gov/library/publications/the-world-factbook/geos/et.html*

DEMEKE, G. and GETACHEW, M. (2006), "Manual annotation of Amharic news items with part-of-speech tags and its challenges", *Ethiopian Languages Research Center Working Papers* 2, 1–16.

DEROSE, S. J. (1988), "Grammatical category disambiguation by statistical optimization", *Computational Linguistics* 14, 31–39.

GAMBACK, B., OLSSON, F., ARGAW, A. A. and ASKER, L. (2009), "Methods for Amharic part-of-speech tagging", in *'AfLaT '09: Proceedings of the First Workshop on Language Technologies for African Languages', Association for Computational Linguistics*, Morristown, NJ, USA, pp. 104–111.

GARSIDE, R. (1987), "The claws word-tagging system", *The computational analysis of English: a corpus-based approach* pp. 30–41.

GARSIDE, R. and SMITH, N. (1997), "A hybrid grammatical tagger: Claws4", *Corpus annotation: Linguistic information from computer text corpora* pp. 102–121.

GASSER, M. (2009), "Semitic morphological analysis and generation using finite state transducers with feature structures, in *'EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics'*, *Association for Computational Linguistics*, Morristown, NJ, USA, pp. 309–317.

GEBRE, B. G. (2009), "Part of speech tagging for Amharic", in *'ISMTCL Proceedings, International Review Bulag'*, PUFC, ISSN 0758 6787, ISBN 978-2-84867-261-8, pp. 114–120.

GETACHEW, M. (2001), Automatic part of speech tagging for Amharic: An experiment using stochastic hidden markov (hmm) approach, Master's thesis, Addis Ababa University.

GIMENEZ, J. and Marquez, L. (2004), "SVMTool: A general pos tagger generator based on support vector machines", *in 'Proceedings of the 4th International Conference on Language Resources and Evaluation'*, Citeseer, pp. 43–46.

GREENE, B. and RUBIN, G. (1971), "Automatic grammatical tagging of English", Providence, RI: Department of Linguistics, Brown University.

JURAFSKY, D., MARTIN, J. and KEHLER, A. (2000), *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, MIT Press.

KUDO, T. (2007), "CRF++: Yet another crf toolkit", software available at *http://crfpp.sourceforge.net*.

LAFFERTY, J. (2001), "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", Morgan Kaufmann, pp. 282–289.

LOPER, E. and BIRD, S. (2002), "NLTK: The natural language toolkit", CoRR cs.CL/0205028.

MARCUS, M.P., MARCINKIEWICZ, M. A. and SANTORINI, B. (1993), "Building a large annotated corpus of English: the Penn Treebank", *Comput. Linguist.* 19(2), 313–330.

MARSHALL, I. (1983), "Choice of grammatical word-class without global syntactic analysis: tagging words in the lob corpus", *Computers and the Humanities* 17(3), 139–150.

McCALLUM, A. (2002), "Mallet: machine learning for language toolkit".

STOLZ, W. S., TANNENBAUM, P. H. and CARSTENSEN, T. V. (1965), "A stochastic approach to the grammatical coding of English", *Communications of the ACM* pp. 399–405.

TACHBELIE, M. and MENZEL, W. (2009), "Amharic part-of-speech tagger for factored language modeling".

TSURUOKA, Y., TATEISHI, Y., KIM, J., OHTA, T., McNAUGHT, J., ANANIADOU, S. and TSUJII, J. (2005), "Developing a robust part-of-speech tagger for biomedical text", *Advances in Informatics* pp. 382–392.

VOUTILAINEN, A. (1995), "A syntax-based part-of-speech analyser", in *'Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics'*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 157–164.