***N. Cortegoso Vissio, V. Zakharov***

**A rule-stochastic hybrid POS-tagger for Sranan Tongo with minimal lexicon and training dataset**

**Abstract.** This article presents the results of an experiment designed to evaluate the performance of a part-of-speech (POS) tagger for Sranan Tongo. The tagger combines a rule-based approach and a minimal lexicon with a trigram tag model to disambiguate the attributed tags. Since Sranan Tongo has no corpora, the trigram model was trained on sentences extracted from the APiCS dataset (327 sentences, 2851 tokens) [Winford et al, 2013] and a description of the language (219 sentences, 1858 tokens) [Nickel et al., 1984], that were manually annotated with POS tags for this purpose.

**Keywords.** Sranan Tongo, rule-based, POS-tagger, trigram tag model, low-resource.

**1. Introduction**

Sranan Tongo is a Creole language spoken in urban areas of Suriname and by the Surinamese diaspora in The Netherlands. Although since 1950 some Surinamese writers have been publishing their works in Sranan Tongo, it is still primarily a spoken language, used especially in informal communication. The official language in Suriname is Dutch and for this reason, there are very few texts written in Sranan Tongo. As far as natural language processing tools are concerned, there are no automatic morphoanalyzers for Sranan Tongo.

In Sranan Tongo the semantics of words do not determine part-of-speech affiliation and grammatical relationships are expressed through word order. This language shows also a high degree of multifunctionality, so that lexical elements can function as members of different grammatical categories without any change in form [Sebba, 1997:119]. For example:

* transitive verbs can be employed as nous;
* stative verbs can function as predicative adjectives;
* attributives adjectives are used as nouns;
* most attributives adjectives could perform as adverbs and vice-versa.

Words such as verbs of motion, time and aspect markers or conjunctions are less likely to take other functions in the syntax, although they may have homonyms in other classes. Anyways, in most of the cases, the class of a word is determined by examining the possibilities of a combination of words in the context. For example, given the following sentence:

*mi nen Juwan [Voorhoeve, 1956:191]*

The word form «nen» can function as the noun «name» or as the verb «to be called». In addition, «mi» can refer both to the personal pronoun «I» and the possessive «my». Although the interpretation of the pair «mi nen» as «my name» is correct, it is impossible to read the whole sentence as «my name *is* Juvan», because the linking copula «na» is missing. Consequently, in the context of Voorhoeve’s example, the word form «nen» can only be a verb.

Hidden Markov models (HMM) are probabilistic sequence classifiers that have been widely used in part-of-speech tagging and word class disambiguation. Being a purely stochastic method, HMM need to be trained on corpora but, unfortunately, there is no corpus available yet for Sranan Tongo. Since, manual tagging is an expensive and time-consuming task, the experiment presented here proposes a workaround to train a model on a very small amount of data while trying to overcome data sparsity. The resulting tagger is a hybrid that disambiguates pre assigned POS tags using trigrams.

**2. The hidden Markov model**

HMM is a stochastic model in which it is assumed that the system to be modeled is a Markov process with unknown parameters (hidden states). The task of the model is to determine the hidden parameters of a sequence, in this case the part-of-speech tags, from the observable parameters, the word forms.

To train the model, it is necessary to compute two parameters in a labeled corpus with POS tags, the emission and transition probabilities:

* the emission probability *p(wi|ti)* is the conditional probability that a word *wi* corresponds to the tag *ti*. This assumes that the probability of an output observation *wi* depends only on the state that produced the observation ti and not on any other state or observation;
* the transition probability (for trigrams) *p(ti|ti-2, ti-1)* is the probability that the tag *ti* occurs, provided that is preceded by the tags *ti-1* and *ti-2*. This relies on the assumption that the probability of a particular tag depends only on the previous two.

Once the model is trained, to predict the most probable sequence of tags for a given a word sequence the following formula is applied:

| $$argmax\left(x\_{1...}x\_{n},y\_{1...}y\_{n}\right)≈argmax\prod\_{i=1}^{n+1}q\left(y\_{i}|y\_{i-2},y\_{i-1}\right)\prod\_{i=1}^{n}e\left(x\_{i}|y\_{i}\right)$$ | (1) |
| --- | --- |

where the left side of the equation is the joint probability of a sequence of word forms and a sequence of tags while the right side is the product of two factors: *q* the probability that tag *yi* occurs given before it appeared tags *yi-1* and *yi-2* and *e,* the probability that tag *yi* corresponds to the word form *xi*. The most likely tag sequence is the one that maximizes the product of on both sides of the equation [Jurafsky, 2009:139-141].

Returning to the task of building a POS-tagger for Sranan Tongo, the starting hypothesis is that the size of the training data is not critical to calculate the transition probabilities, provided it covers most of the syntactical features of the language. Another assumed principle is that the examples from the APiCS database and those given by Nickel et al. are a good option to train the model. Since they are part of works that describe most of the morph-syntactic features of the language, it is very likely that they will also cover most of the common POS combination sequences in Sranan Tongo.

However, if the training data is not big enough, the lexical items and their respective part-of-speech could not be learned from the data and must therefore be provided from another source to the tagging algorithm. For this matter, a lexicon was extracted from the online version of the Sranan Tongo-English dictionary «Wortubuku fu Sranan Tongo» [Wilner, 2007].

In the proposed experiment, instead of calculating the emission probabilities, the model only counts tags during the training process. These counts are then used by the tagging algorithm to estimate “on the fly” an replacement for the emission probabilities. Transition probabilities are computed as usual.

**3. Tag set, lexicon and tagging algorithm**

Despite the fact that the «Wortubuku fu Sranan Tongo» includes part-of-speech tags for the entries, it does so regarding the grammatical categories of the target languages (English or Dutch). Consequently, some manual editing was required to adjust the entries to the set of tags used in this experiment.

The part-of-speech tags were defined mainly on the basis of the language description from Nickel et al. [1984] with some minor changes, for a total of 31 tags. Along with the parts-of-speech that are commonly listed (noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection, numeral, article, etc), other more specific POS tag have also been introduced (proper name, wh-question, copula, determiner, modal, tense and aspect marker, etc).

The compiled lexicon is trusted to cover most (if not all) words of the closed class like articles, pronouns, modals, etc. However, regardless of size, a lexicon cannot contain all open class words in a language like nouns, verbs, adjectives and adverbs, interjections. To make things worse, Sranan Tongo shows variations in spelling (despite standardization efforts) and borrows extensively words from Dutch. For this reason, the tagging algorithm will come across many words that are not included in the lexicon. To avoid handling unknown words, each word is pre-marked with a list that contains the tags «noun», «verb», «adjective» and «adverb». The preassignment of these tags is an attempt (albeit very simplistic) to simulate the previous described phenomenon of word multifunctionality.

This general assumption of multifunctionality is then restricted by the lexicon following some simple conditions. Once the pre-tagging is done, the algorithm looks for the word form in the lexicon, if it is found, then the preassigned tag list is replaced by the tag list given in the lexicon. Words of the closed class that have a structuring role in the grammar are expected to be identified by the lexicon. If the word is not in the lexicon, the algorithm checks if it starts with a capital letter. If this is not the case, the tags from the preassigned list are considered the best guess. But if the word does start with a capital letter and it is not the first word in the sentence, then the algorithm assumes that it is a proper noun and the preassigned list is overwritten with this tag. However, if the word does occupy the starting position of the sentence, then the algorithm cannot accept the uppercase letter as conclusive proof that the word is a proper noun, so it returns the preassigned list with the addition of the tag corresponding to the proper noun.

After these rules were applied, if the mapping word form/part-of-speech is unambiguous, its probability equals 1 (tags that are not listed for a given word form have probability 0 and, consequently, they cannot occur). If a word form has more than one tag assigned, then the probability for each tag is calculated using one of the three metrics described in the next section.

**4. The metrics**

All three metrics share the same principle: they use the tag counts from the training set to calculate the probability of the tags for a given word form.

The first metric (A) translates the proportion of the tags in the training data in the context of the assigned tags for a word form. Tags with higher counts in the training data get higher probabilities:

| $$p\left(t\_{i}\right)=\begin{array}{c}tf\_{i,j}\\\overline{\sum\_{i=1}^{n}tf\_{i}}\end{array}$$ | (A) |
| --- | --- |

where *tf* is the number of times the tag *i* appears in the training data *j*. The count is normalized by adding all the frequencies of the tags in the list. For example, if the list for a given word form *w* contains *t1*, *t2* and *t3* and *t1* appears 100 times in the training data, while *t2* and *t3* just 20 and 10 respectively, the probabilities of the tags will be 0,77, 0,15 and 0,08.

The metric (B) is basically the same as (A) with the addition of the natural logarithm to smooth out the difference between tag counts:

| $$p\left(t\_{i}\right)=\begin{array}{c}ln\left(tf\_{i,j}\right)\\\overline{\sum\_{i=1}^{n}ln\left(tf\_{i}\right)}\end{array}$$ | (B) |
| --- | --- |

as result, the probabilities of the tags *t1*, *t2* and *t3* from the previous example are now closer to each other: 0,47, 0,30, 0,23.

The third metric (C) penalizes frequency, making those tags with lower counts more likely:

| $$p\left(t\_{i}\right)=\begin{array}{c}\sum\_{i=1}^{n}tf\_{i,j}-tf\_{i,j}\\\overline{\sum\_{i=1}^{n}tf\_{i}n-1}\end{array}$$ | (C) |
| --- | --- |

where the numerator is the difference between the total frequencies of the tags in the list and the count for a given tag, so that tags with lower counts get higher values. The denominator contains the normalizing constant. The tags *t1*, *t2* and *t3* from the two previous examples will get now the probabilities of 0,12, 0,42 and 0,46 respectively.

Table 1 shows the POS tags given to the word form «moro» (more) according to the lexicon, the frequencies of the POS tag in a training set with 3890 tags and the probabilities calculated on the basis of the three described metrics:

*Table 1.* **Tag probability for «moro» (more)**

| **tag** | **f** | **part-of-speech** |  **(A)** | **(B)** | **(C)** |
| --- | --- | --- | --- | --- | --- |
| RB | 129 | adverb | 0,66 | 0,42 | 0,16 |
| AB | 48 | quantifier | 0,24 | 0,33 | 0,37 |
| COMP | 16 | comparative | 0,08 | 0,24 | 0,45 |

**5. Experiment and testing results**

For the experiment the HMM was trained in four stages to assess the impact of the amount of training data on the results. Consequently, the training data was divided into four samples. The first two contain the examples from the APiCS database [Winford et al., 2013] and the last two, those from [Nicket et al., 1984]:

* T1: APiCS database: 164 sentences, 1472 tokens;
* T2: APiCS database: 165 sentences, 1381 tokens;
* T3: Nickel et al.: 111 sentences, 827 tokens;
* T4: Nickel et al.: 110 sentences, 833 tokens.

During the first stage of training, only sample T1 was used. The other samples (T2, T3, T4) were added one by one in the following instances, which means that in the fourth stage the model was trained with all of them. The performance of the model was evaluated on the test data after each phase. The values presented for precision, recall and f-score are the average of the individual measurements for the 31 tags.

The test data is intended to represent different syntactic constructs. It consists of 70 manually POS-tagged sentences extracted from the «Wortubuku fu Sranan Tongo» [Wilner, 2007] (612 tokens).

The employed lexicon was minimal, including only 346 word forms, although it comprised all the closed class words in the online dictionary «Wortubuku fu Sranan Tongo» [Wilner, 2007].

*Table 2.* **Testing results**

|  | **x=T1** | **x+T2** | **x+T3** | **x+T4** |
| --- | --- | --- | --- | --- |
| (s) cumulative sentences | 164 | 329 | 440 | 550 |
| (t) cumulative tokens | 1472 | 2853 | 3680 | 4513 |
| (K) | precision | 0,75 | 0,79 | 0,80 | 0,78 |
| recall | 0,74 | 0,78 | 0,80 | 0,76 |
| f-score | 0,72 | 0,77 | 0,78 | 0,76 |
| (A) | precision | 0,79 | 0,79 | 0,81 | 0,80 |
| recall | 0,72 | 0,73 | 0,74 | 0,72 |
| f-score | 0,73 | 0,73 | 0,75 | 0,73 |
| (B) | precision | 0,78 | 0,80 | 0,79 | 0,78 |
| recall | 0,73 | 0,76 | 0,77 | 0,75 |
| f-score | 0,73 | 0,77 | 0,76 | 0,75 |
| (C) | precision | 0,71 | 0,79 | 0,81 | 0,81 |
| recall | 0,74 | 0,80 | 0,81 | 0,81 |
| f-score | 0,70 | 0,77 | 0,78 | 0,79 |

* x is the training data, initialized in the column (x=T1) with the first training sample and expanded in the successive stages;
* the columns (x=T1), (x+T2), (x+T3) and (x+T4) contain the values and results for each of the stages of the experiment;
* (s) shows the cumulative number of sentences and (t) the cumulative tokens the model is trained on after adding a new sample to the training data;
* (K) is a constant = 1 replacing the emission probabilities, so the model predictions depend exclusively on the transition probabilities. This represents the baseline that is expected to be improved upon;
* (A) (B) and (C) show the performance of the model when applying the respective metrics to replace the emission probabilities;

The experiment did not take into account the results of tagging the punctuation signs that would have artificially improved the previous numbers.

**5. Discussion**

The remarks presented here are drawn primarily from observing the f-score values. As expected, the size of the training data has a significant impact on the model's predictions. However, the baseline (K), which only shows the disambiguation power of the trigrams applied directly on the lexicon constraints, does not improve from (x+T3) to (x+T4) after adding 110 sentences and 883 tokens. Future experiments should explore the threshold, where more training data ceases to significantly alter the transition probabilities.

Metric (A) has generally good precision but low recall. This happens because this metric favors tags with higher counts in the training data. Nouns and verbs are likely to be correctly identified (increasing overall precision), while less frequent parts-of-speech are misclassified (reducing the recall). The natural logarithm from metric (B) reduces the negative effect of the extreme counts from metric (A) while maintaining the relationship between more and less frequent tags in the training data. Nevertheless (B), just like metric (A), does not improve over the baseline.

Metric (C), that penalizes tags with higher counts in the training data, is the only method to perform above the baseline. It seems to indicate that a metric that promotes less frequent tags has a positive impact on the classifications when combined with the transition probabilities. Further experiments should be conducted to analyze the behavior of this metric with more training data.

**7. Conclusions**

This paper shows how far a POS-tagger for Sranan Tongo can go with very limited resources. The presented tagger is a hybrid that combines a rule-based and a stochastic approach. It relies on a broad assumption of the language (word multifunctionality), a minimal lexicon (containing almost only closed class words) and a couple of simple rules for assigning possible POS tags to a word form. The probabilities of the attributed tags are estimated by simple tag counts and then disambiguated using a trigram tag model trained with just 550 sentences.

Despite the fact that the performance achieved by the tagger is low, it serves as a baseline for future developments. The stochastic part of the tagger could improve with the addition of more training data. A broader lexicon will certainly limit the incidence of the preassigned tags, easing the disambiguation task and yielding better results.

Finally, although the article revolves around the case of Sranan Tongo, the method drawn here can be extrapolated to languages with a similar structure.

**References**

1. *Bocharovl .V.V. et Mitrenina O.V.* (2017), Komp'juternaja morfologija [Computer morphology]. In: *Nikolaev I.S., Mitrenina O.V. et Lando T.M.* (Ed.), 2nd ed, Saint Petersburg, pp. 14-33.

2. Jurafky*.D. et Martin J.* (2019), Speech and language processing: An introduction to natural language processing, computional linguistics, and speech recognition, 2nd. ed. Pearson, Upper Saddle River, New Jersey, pp 139-151.

2. *Nickel M., Wilner J.* (1984), Papers on Sranan Tongo. n.d. Summer Institute of Linguistics, available at: https://archive.org/
details/rosettaproject\_srn\_morsyn-1 (last access 21.05.2021)

3. *Sebba M.* (1981), Derivational regularities in a Creole lexicon: the case of Sranan, Linguistics an Interdisciplinary Journal of the Language Sciences. De Gruyter Mouton. 22 (4), pp. 719–763.

4. *Sebba M.* (1997), Contact languages pidgins and creoles, St. Martins’s Press, New York, 22 (4).

5. *Voorhoeve J.* (1956), Structureel onderzoek van het Sranan. De West-Indische Gids, 37ste Jaarg, pp. 189-211.

6. *Wilner J.* (2007), Wortubuku fu Sranan Tongo. Sranan Tongo English Dictionary. SIL International Fifth edition.

7. *Winford D., Plag I.* (2013), Sranan structure dataset. In: Michaelis, Susanne Maria & Maurer, Philippe & Haspelmath, Martin & Huber, Magnus (eds.) Atlas of Pidgin and Creole Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology, available online at http://apics-online.info/contributions/2 (last access 21.05.2021)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Victor Zakharov**

Saint Petersburg State University (Russia).

***E-mail:*** vz1311@yandex.ru

**Nicolás Cortegoso-Vissio**

Saint Petersburg State University (Russia).

***E-mail:*** st082534@student.spbu.ru