talenta PUBLISHER

DATA SCIENCE

Journal of Computing and Applied Informatics

Samawa Part of Speech Tagging using Brill Tagger

***Trienani Hariyanti*[1]*, Saori Aida*[2]*, and Hiroyuki Kameda*[3]**

[1]*Universitas Teknologi Sumbawa, Sumbawa, Indonesia*
[2,3]*School of Computer Science, Tokyo University of Technology, Tokyo, Japan*

Abstract. There exist 7,097 living languages in the world cited by Ethnologue. Most of them, however, do not exist on the Internet as the objects of research. It indicates the gap in language resources. One of them is Samawa language which has over 500,000 native speakers and is identified as an endangered language by UNESCO. What we have known about Samawa so far is a lack of information, tools, and resources to maintain its sustainability. This paper aims to contribute to NLP, a growing field of research, by exploring Samawa part of speech tagging problem using rule-based approach, i.e. Brill tagger. It has been trained on very limited data of Samawa corpus, which is 24,627 tokens including punctuation marks with 24 tags of our original tagset. K-fold cross-validation (k = 5 and k = 10) was applied to compare Brill's performance with Unigram, HMM, and TnT. Brill tagger with the combination of default tagger, Unigram, Bigram and Trigram as baseline tagger achieve higher accuracy over 95% than others. It suggests that the Brill tagger can be used to extend Samawa corpus automatically.

Keyword: Samawa language, Brill tagger, part of speech tagging, accuracy

Abstrak. *Terdapat 7,097 bahasa yang hidup di dunia yang dirilis oleh Ethnologue. Banyak dari bahasa-bahasa tersebut tidak terdapat di Internet sebagai objek riset. Hal ini menunjukkan adanya kesenjangan dalam sumber daya keberadaan sumber daya Bahasa tersebut. Salah satunya adalah Bahasa Samawa yang memiliki 500,000 penutur aktif dan dikategorikan sebagai bahasa yang punah oleh UNESCO. Apa yang kita ketahui tentang Samawa adalah kurangnya informasi, alat-alat, dan sumber daya yang menunjang keberlanjutannya. Paper ini bertujuan untuk berkontribusi kepada NLP, sebuah bidang riset yang sedang berkembang, dengan mengeksplorasi permasalahan penandaan kelas kata dengan menggunakan pendekatan berbasis aturan, yaitu Brill tagger. Brill tagger dilatih pada korpus Samawa yang terdiri dari 24,627 token termasuk tanda baca dengan 24 kelas kata. Prosedur k-fold cross-validation (k = 5 dan k = 10) diterapkan dan membandingkan kinerja dari Brill dengan Unigram, HMM, and TnT. Brill tagger dengan kombinasi tagger default, Bigram, dan Trigram sebagai tagger dasar mencapai akurasi yang tertinggi yakni 95% dibanding lainnya. Ini menunjukkan bahawa Brill tagger dapat digunakan untuk memperluas korpus Samawa secara otomatis*

Kata Kunci: *Bahasa Samawa, Brill tagger, penandaan kelas kata, akurasi*

Received 18 March 2019 | Revised 12 July 2019 | Accepted 12 July 2019

*Corresponding author at: Universitas Teknologi Sumbawa, Sumbawa, Indonesia

E-mail address: trienanihariyanti@gmail.com, saori@stf.teu.ac.jp, kameda@stf.teu.ac.jp

ISSN: 2580-6769 | e-ISSN: 2580-829X | DOI: 10.32734/jocai.v3.i2-860
Journal Homepage: https://jocai.usu.ac.id

1 Introduction

Natural language processing abbreviated by NLP is a branch of artificial intelligence that helps computers to understand, interpret, and manipulate human languages. NLP belongs surely to many disciplines, including computer science and computational linguistics, in an attempt to fill the communication gap between a human and a computer. As a human, we speak and write in English, Spanish, Japanese and others. However, a computer's native language known as machine code or machine language is generally incomprehensible to people. The communication occurs not with words, but through ones and zeros that produce consistent actions.

To converse with humans, a program must understand syntax, semantics, morphology, phonology and so on. Recently, there are a number of different NLP tasks incorporated into software programs, such as part of speech tagging (PoS), information retrieval, automatic summarization, machine translation and so on. Part of speech tagging is a technique that reads a text in some languages and assigns its part of speech to each word (and another token), such as noun, verb, adjective, and others. It is, however, useful in itself as an essential step in many NLP pipelines, informing deeper layers of annotation and helping to understand the syntactic aspect of the language.

Automatic part of speech tagging methodologies fell into two distinct groups, i.e., rule-based and stochastic (probabilistic) taggers. Eric Brill's tagger is one of the first and the most widely used English Post-tagger, employs rule-based algorithms. Typically rule-based approaches use contextual information to assign tags to unknown or ambiguous words. Disambiguation was done by analyzing the linguistic features of the word itself, its pre-context words, its post-context words, and rules of some sort. Defining a set of rules by hand is a quite extremely cumbersome process and is not scalable. For this reason, it is strongly required some automated fashion of doing this process. Brill's tagger is a rule-based tagger that has the general idea in a simple form such as guessing the tag of each word and going back to fix mistakes. In detail, it goes through the training data and discovers the set of tagging rules that best specify the data and minimize part of speech tagging error. The most notable point to note here about Brill tagger is that the rules are not hand-crafted, but are instead found out using the corpus provided. The only feature required in engineering is a set of rule templates that the model can use to come up with new features. As for stochastic taggers, they have a machine-learning component: the rules automatically induced from previously tagged training corpora. Brill tagger, for example, has transformation templates which examine the nearby words and tags.

There are now 7,097 living languages in the world cited by Ethnologue [1]. However, most of them do not exist on the Internet as the objects of research, and it indicates the gap in language resources. One of them is Samawa language which has over 500,000 native speakers. The atlas of endangered languages by united nations educational, scientific, and cultural organization (UNESCO) is identified and in that Samawa is listed as an endangered language [2]. What we know about Samawa is a lack of information, tools and resources to maintain sustainability. NLP can be one way to overcome these resource barriers. This paper aims to contribute to NLP, a growing field of research, by exploring Samawa.

In this paper, we describe our investigation regarding Brill tagger and implementation of Samawa tagger that we had started from scratch. Section 2 talks about several studies related to Brill tagger in the past. We describe the core of Brill tagger, i.e. transformation-based error-driven learning algorithm and how it works in section 3. Also, we present the Samawa corpus in general in section 4. Furthermore, we present the Samawa part of speech tagger system and compare with other taggers to see their performance and summarize with a conclusion in section 5, 6 and 7, respectively.

2 Related Work

There are several works reported in the literature regarding the implementation of Brill tagger. Hasan et al. [3] using Brill tagger for Bangla with a small size of corpus around 4,484 tokens and achieved 55% accuracy. On the other hand, Brill tagger has trained for German with some manual constraints and lexical look-up could gain around 96% accuracy [4]. Furthermore, Megyesi [5] using Brill's PoS tagger with extended lexical templates increased the accuracy into 97% for Hungarian. Moreover, examined Indonesian using Brill tagger obtained 99.75% accuracy [6]. Indonesian also have 89.70% accuracy when applying on Brill tagger with some modification and rewrite in C# [7]. Other research which implemented Brill's method on Swedish make accuracy in 95.18% [8]. Wilson et al. [9] achieved an accuracy of 97% when using a genetic algorithm in Brill's transformation-based part of speech tagger.

3 Transformation-Based Error-Driven Learning

Eric Brill described a rule-based algorithm for automatic part of speech tagging named Transformation-Based Error-Driven Learning (TEL) in 1995. It works based on transformation learns by detecting errors. In other words, TEL guesses the tag of each word in a sentence, and goes back to fix the mistakes.

This method can operate on two data. The first one is initially unannotated data that simulates the transformation process and records the error. The second one is goal corpus/gold standard data/annotated data. The initially unannotated data can be tagged by any simple part of speech tagger in initial state annotator stage to create the temporary corpus. Once after temporary corpus created, then it will compare with goal corpus which has been tagged manually.

Firstly, the Brill's algorithm works at the system by assign its most likely tag to each word in the training corpus. Then the learning algorithm constructs a ranked list of transformations which will change the initial tagging into the closer one to the correct one. Towards every rewriting rule, the algorithm keeps mark of how many good and bad transformations it is responsible. The goodness of the rewriting rule is the number of good transformations it performed, minus the number of bad transformations. The good rules are appended to an ongoing list, resulting in a list of rewriting rules ranked in descending order of goodness. Since every rule good enough to be attached to the list also gets applied to the training corpus before it stores. The ultimate result of executing this algorithm is a ranked list of rewriting rules that can be used to a new corpus. Figure 1 present how TEL works in general [10].

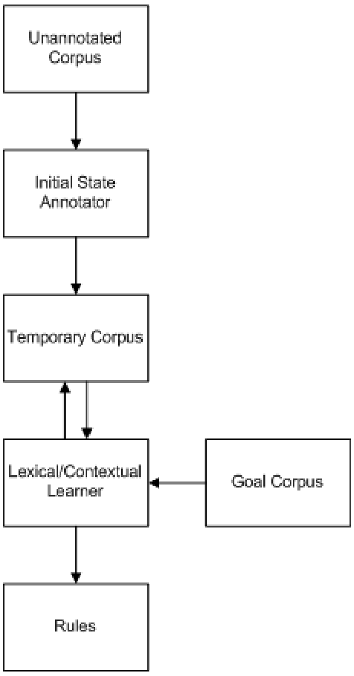


Figure 1. Error-Driven Learning Module (Brill, 1995)

For each iteration, the temporary corpus is updated based on learner and will return the new rule that improves the result of the annotation. Through this process, the learner will analyze and produce an ordered list of rules which can be applied to new unannotated text. Transformation-based error-driven learning uses lexical rules for deriving rules for tagging unknown words, and contextual rules for deriving rules that improve accuracy. A rule distinguishes into two parts: a condition, i.e. the trigger and possibly a current tag, and a resulting tag which has a form:

$$(A, B): X \rightarrow Y \tag{1}$$

A and B as triggering an environment that we observe, then the output will change from tag X to tag Y. The set of rules are created from all possible instantiations of all determined templates before. One of simple rule templates for part of speech tagging is to change the current word tag from tag X to tag Y if the previous word is tagged as Z. Variables X, Y and Z need to be instantiated during the learning process.

Brill tagger distinguishes into two important parameters, i.e., the maximum number of rules and the minimum score. The values for these parameters have to be chosen by the experimenter. Both increasing and decreasing parameters into decent values will affect the performance of taggers. When the combinations using the same performance have more rules or lower minimum score, which makes the training slower and the tagger much more complex.

The following table illustrates the step in TEL. Firstly, we tagged using a unigram tagger, then fixing the errors. All rules are written in the template of the form: “replace T1 with T2 in the context C”. The context usually indicates the word or tag of the previous or following word, or the appearance of a tag with 2 - 3 words of the current word. During the training phase, the TEL will guess for T1, T2 and C and make considerable candidates of rules. Each rule then is assigned the score on its net benefit: the number of incorrect tags that is correct, less than the number of correct tags it incorrectly modifies [11].

Table 1 Transformation-Based Error-Driven Learning in Samawa Corpus

Sentence	Gold	Unigram	Replace DT with PRP when next tag is Z
pang	IN	IN	
ta	PR	PR	
ahir	NN	NN	
mo	RP	RP	
palangan	NN	NN	
sadua	CD	CD	
nya	PRP	DT	PRP
.	Z	Z	

4 Samawa Corpus

In a computational linguistic area, corpora can be defined as a documentation of language in use and provide linguistic diversity. Corpora are classified into unannotated and annotated. Unannotated one usually contain only raw texts, but the text is tokenized and cleaned already. Otherwise, an annotated one is a corpus which tags information. A tagset for natural language processing gives information about a word and its neighbours. In Samawa corpus, we define our original tagset consisting of 24 tags (see detail in our previous work in [12]). Tagging process can be done manually and automatically. For initial corpus, we needed to assign a tag manually and recognized the rules which can be used for others text with the same pattern. Building a corpus feasibly involves a more significant investment in time, resources and energy than any other types of linguistic activity

Building a large size of Samawa corpus is a quite challenging task since the amount of document and textbook in Samawa is limited in size. Also, almost of them are particularly difficult to handle. Raw data used in the Samawa corpus was collected from the manuscripts, textbooks, magazines, and text from websites. Then in the preprocessing phase, it was cleaned and normalized by Unicode in plaintext format. The Samawa corpus was used for training Samawa PoS tagger, which consist of 24,627 hand-tagged tokens. They were collected and manually hand-annotated based on grammatical category of Samawa.

5 The Samawa Part of Speech Tagger System

Throughout this research, we have worked on building Samawa corpus and Samawa part of speech tagger by using Natural Language Toolkit (NLTK), an open source Python library and programs for working with human language data. It contains the text processing library for tokenization, parsing, classification, stemming, tagging and semantic reasoning functionalities. NLTK was created in 2001 by Steven Bird and Edward Loper as part of a computational linguistics course in the Department of Computer and

Information Science at the University of Pennsylvania [11]. It provides many NLP data types, processing tasks, corpus samples and readers, together with animated algorithms, tutorials and problem sets [13].

The application program based on Graphical User Interface (GUI) is designed to tag a word automatically to know the correct tag of a token. Nevertheless, a new document as the data input must be changed to format .txt and cleaned before applying in the tagging process. The Samawa corpus which has been cleaned and tagged based on the grammatical category of Samawa plays an essential role in the training process. Figure 2 shows the design of the training process and GUI application.

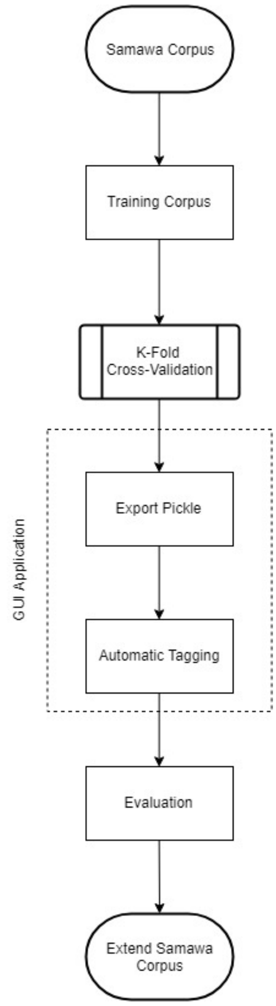
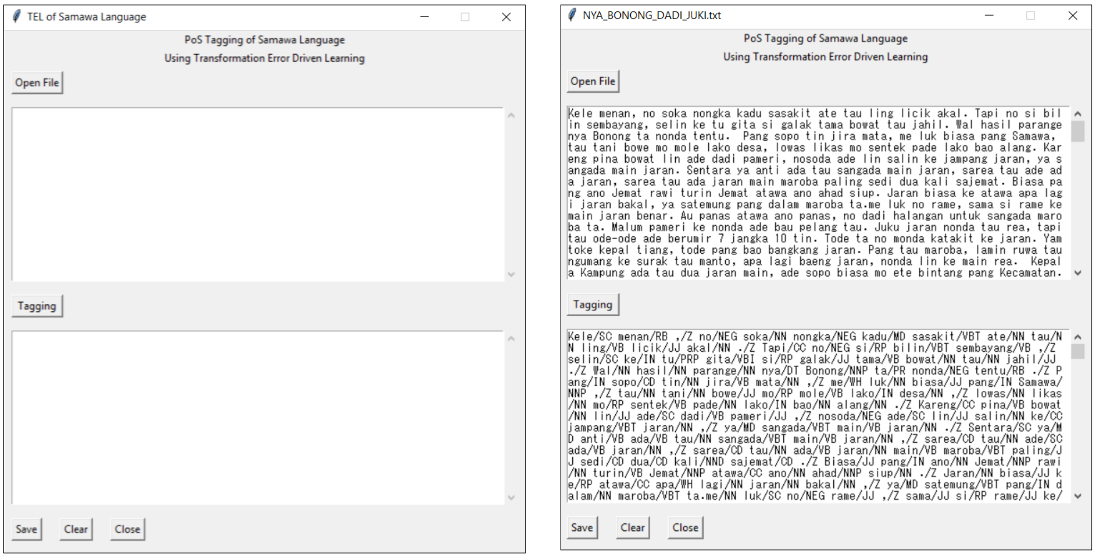


Figure 2 The design of Samawa PoS tagger system

After training the training phase, the performance between Unigram, HMM, TnT and Brill will be evaluated by k-fold cross-validation method to estimate the skill of those models. It is used to determine how the model is expected to perform in general while applied to produce prediction in the real data. Based on these results, the tagger will retrain all of tokens and it will be exported as a pickle file. This file is used to save the tagger as an object serialization and can be loaded in a simple way when an automatic tagging process happen. Afterwards, we enforce the evaluation in the form of identifying the tagging errors and the inconsistencies by hand-correcting to see the result of automatic tagging in the new document. The last phase is to merge the result of evaluation into Samawa corpus as a part of extending the corpus process. However, it is crucial to extend the corpus and achieve better performance of a tagger.

The application program made with Tkinter which is the standard Python for GUI application. It is the most commonly used method to develop the fastest and easiest GUI application and provide many classes as widgets. Figure 3 displays the GUI application that Brill presents the application while a new document is imported and tagged.


TEL of Samawa Language
PoS Tagging of Samawa Language
Using Transformation Error Driven Learning
Open File
Tagging
Save
Clear
Close
NYA_BONONG_DADI_JUKI.txt
PoS Tagging of Samawa Language
Using Transformation Error Driven Learning
Open File
Tagging
Save
Clear
Close

```
Kele menan, no soka nongka kadu sasakit ate tau ling licik akal. Tapi no si bil
in sembayang, selin ke tu gita si galak tama bowat tau jahil. Wal hasil parange
nya Bonong ta nonda tentu.  Pang sopo tin jira mata, me luk biasa pang Samawa,
tau tani bowe mo mole lako desa, lowas likas mo sentek pade lako bao alang. Kar
eng pina bowat lin ade dadi pameri, nosoda ade lin salin ke jampang jaran, ya s
angada main jaran. Sentara ya anti ada tau sangada main jaran, sarea tau ade ad
a jaran, sarea tau ada jaran main maroba paling sedi dua kali sajemat. Biasa pa
ng ano Jemat rawi turin Jemat atawa ano ahad siup. Jaran biasa ke atawa apa lag
i jaran bakal, ya satemung pang dalam maroba ta.me luk no rame, sama si rame ke
main jaran benar. Au panas atawa ano panas, no dadi halangan untuk sangada maro
ba ta. Malum pameri ke nonda ade bau pelang tau. Juku jaran nonda tau rea, tapi
tau ode-ode ade berumir 7 jangka 10 tin. Tode ta no monda katakit ke jaran. Yam
toke kepal tiang, tode pang bao bangkang jaran. Pang tau maroba, lamin ruwa tau
ngumang ke surak tau manto, apa lagi baeng jaran, nonda lin ke main rea.  Kepal
a Kampung ada tau dua jaran main, ade sopo biasa mo ete bintang pang Kecamatan.
```

```
Kele/SC menan/RB ,/Z no/NEG soka/NN nongka/NEG kadu/MD sasakit/VBT ate/NN tau/N
N ling/VB licik/JJ akal/NN ./Z Tapi/CC no/NEG si/RP bilin/VBT sembayang/VB ,/Z
selin/SC ke/IN tu/PRP gita/VBI si/RP galak/JJ tama/VB bowat/NN tau/NN jahil/JJ
./Z Wal/NN hasil/NN parange/NN nya/DT Bonong/NNP ta/PR nonda/NEG tentu/RB ./Z P
ang/IN sopo/CD tin/NN jira/VB mata/NN ,/Z me/WH luk/NN biasa/JJ pang/IN Samawa/
NNP ,/Z tau/NN tani/NN bowe/JJ mo/RP mole/VB lako/IN desa/NN ,/Z lowas/NN likas
/NN mo/RP sentek/VB pade/NN lako/IN bao/NN alang/NN ./Z Kareng/CC pina/VB bowat
/NN lin/JJ ade/SC dadi/VB pameri/JJ ,/Z nosoda/NEG ade/SC lin/JJ salin/NN ke/CC
jampang/VBT jaran/NN ,/Z ya/MD sangada/VBT main/VB jaran/NN ./Z Sentara/SC ya/M
D anti/VB ada/VB tau/NN sangada/VBT main/VB jaran/NN ,/Z sarea/CD tau/NN ade/SC
ada/VB jaran/NN ,/Z sarea/CD tau/NN ada/VB jaran/NN main/VB maroba/VBT paling/J
J sedi/CD dua/CD kali/NND sajemat/CD ./Z Biasa/JJ pang/IN ano/NN Jemat/NNP rawi
/NN turin/VB Jemat/NNP atawa/CC ano/NN ahad/NNP siup/NN ./Z Jaran/NN biasa/JJ k
e/RP atawa/CC apa/WH lagi/NN jaran/NN bakal/NN ,/Z ya/MD satemung/VBT pang/IN d
alam/NN maroba/VBT ta.me/NN luk/SC no/NEG rame/JJ ,/Z sama/JJ si/RP rame/JJ ke/
```

Figure 3 GUI application for automatic tagging and the display of tagging process

The open file button uses to open the new document in a .txt format file. After selecting the data, the text will appear in the text area. Then, the tagging button gives a tag to each token based on Brill tagger. The tagging text area will display the result of the tagging. Furthermore, the save button, and the clear button will be keeping the result of automatic tagging and clean both text area and tagging area, respectively. Later on, the close button is for closing the application.

6 Experimental Result

The whole training data were running on NLTK module and used relatively small tagged corpus that contains 24,627 tokens with 2,904 unique words. We had compared the performance of Unigram, HMM, TnT and Brill use k-fold cross-validation method with k = 5 and k = 10, respectively. As seen in table 2, HMM tagger has a poorer accuracy than others. It could happen due to the limited size of the corpus as training data which is around 20K tokens. Unigram and TnT achieve average accuracies of about 91% and 92%, respectively. It is only 4% lesser than Brill version 2. The size of corpus affected the performance in part of speech tagging problem. This finding confirmed by Hasan et al. [3] which has explored from the small size of the token to large scale of tokens. Performance increase followed the increasing of the corpus size.

Especially for HMM which need 1M tokens to achieved around 96.7% accuracy [9]. The detail result regarding accuracy of each tagger given in table 2 below.

Table 2 Accuracy result with k = 5 and k = 10

Tagger	Accuracy 5-fold	Accuracy 10-fold
Unigram	91.91%	91.76%
HMM	34.12%	46.47%
TnT	91.54%	92.15%
Brill v1	67.54%	69.41%
Brill v2	95.78%	95.67%

Generally, Brill tagger in this experiment has two versions of initial state annotator as baseline tagger. First, Brill in version 1 contains baseline tagger, i.e. default tagger. This default tagger assigns all of the tags of each token in Samawa corpus as a Noun (NN). Then, Brill will take its capacity to fix the error. Unfortunately, the performance of this version has quite a low accuracy. It is caused by choosing default tagger which is tag each even unknown word as a noun. Moreover, it takes much time (both k = 5 and k = 10) to get the accuracy.

In Brill version 2, the combination of default tagger as Noun, Unigram, Bigram and Trigram has chosen as baseline tagger, and Brill algorithm as the main part made the highest accuracy 95.78% and 95.67% for k = 5 and k = 10, respectively. Unigram, Bigram and Trigram use statistics of previous one, two and three tags while Brill uses information of surrounding tags and words.

In Brill, the rules which are contributing the most to improve the tagging accuracy can be seen after tagging process. One of the exciting rules that formed from Samawa corpus is related to tagging word *'nya'*. In default, this word has tagged as a determiner (DT), but while the part of speech of the following word is Z (even a comma or full stop), will change DT to PRP. There are 81 rules which are generated by Brill. Table 3 below presents the top 10 rules that Brill tagger learn from Samawa corpus.

Table 3 Top 10 Brill contextual rules in Samawa corpus

Part of Speech (PoS)	Contextual rules
DT → PRP	If the word is 'nya' and the PoS of following word is Z
NNP → DT	If the word is 'lalu' and and the Word of the following word is "Lepang", and the Word of word i+2 is "Kuning"
DT → PRP	If the word is 'nya' and the PoS of following word is PR
DT → PRP	If the word of words i+1...i+2 is "diri"
NN → NND	If the word is 'tau' and the word of the preceding word is 'sopo'
IN → CC	If the word is 'ke' and the PoS of following word is VBT
NND → NN	If the word is 'tau' and the word of preceding word is 'sarea'
PRP → DT	If the word is 'nya' and PoS following word is NNP
DT → PRP	If the word is 'nya' and the PoS of following word is PR
IN → RP	If the word is 'ke' and the PoS of following word is Z

7 Conclusion

In this work, we have presented Eric Brill's rule-based PoS tagger which automatically acquires rules from a training corpus, based on transformation-based error-driven learning algorithm. Tagger has been trained on very limited data of Samawa corpus, which consisting of 24,627 tokens including punctuation marks. The tagset of the training corpus consists of 24 part of speech tags**.** The result shows that the accuracy was 34.12% and 46.4% for HMM which delivers poorer accuracy than the others. Followed by TnT and Unigram in 91.54%, 92,15%, 91.91% and 91.76%, respectively. Besides, Brill version 1 make 67.54% and 69.41% each. Overwhelmingly, 95.78% and 95.67% were obtained by Brill version 2. These accuracies acquire from k-fold cross-validation procedure with k = 5 and k = 10, respectively. Based on these results, we take Brill version 2 to extend our corpus size as the required in NLP task. For the higher accuracy could probably gain using a large corpus size.

8 Acknowledgement

Mrs Trienani would thank Indonesia Endowment Fund for Education (LPDP) for acknowledge gratefully funding for her study.

References

[1] Simons, Gary F., and Charles D. Fennig (eds.), "Ethnologue: Languages of the World, Twenty-first edition," *Ethnologue*, 2018. [Online]. Available: https://www.ethnologue.com. [Accessed: 05-June-2018].

[2] "UNESCO Atlas of the World's Languages in danger." [Online]. Available: http://www.unesco.org/languages-atlas/. [Accessed: 10-Dec-2018].

[3] F. M. Hasan, N. UzZaman, and M. Khan, "Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, Springer, pp. 121–126, 2007.

[4] G. Schneider and M. Volk, "Adding manual constraints and lexical look-up to a Brill-tagger for German," in *In Proceedings of the ESSLLI-98 Workshop on Recent Advances in Corpus Annotation*, 1998.

[5] B. Megyesi, "Brill's PoS tagger with extended lexical templates for Hungarian," in *Proceedings of the Workshop (W01) on Machine Learning in Human Language Technology*, pp. 22–28, 1999.

[6] V. Christanti, J. Pragantha, and E. Purnamasari, "Implementasi Brill Tagger untuk POS-Tangging Dokumen Bahasa Indonesia", *Jurnal Universitas Kristen Krida Wacana*, vol. 1, no. 3, pp. 315–301, 2012.

[7] E. R. Setyaningsih, "Part of Speech Tagger untuk Bahasa Indonesia dengan menggunakan Modifikasi Brill," *Dinamika Teknoogi.*, vol. 9, no. 1, pp. 37–42, Apr. 2017.

[8] M. Larsson and M. Norelius, "Part-of-Speech Tagging Using the Brill Method," *Proj. 2004*, p. 69, 2004.

[9] G. Wilson and M. Heywood, "Use of a genetic algorithm in brill's transformation-based part-of-speech tagger," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 2067–2073, 2005,.

[10] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging," *Computer Linguistic.*, vol. 21, no. 4, pp. 543–565, 1995.

[11] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[12] T. Hariyanti, S. Aida, and H. Kameda, “Samawa Language: Part of Speech Tagset and Tagged Corpus for NLP Resources,” *Journal Physics Conference. Series.*, vol. 1061, p. 012007, 2018.

[13] S. Bird, “NLTK: the natural language toolkit,” in *Proceedings of the COLING/ACL on Interactive presentation sessions*, pp. 69–72, 2006.