A Robust Transformation-Based Learning Approach Using Ripple Down Rules for Part-Of-Speech Tagging

Dat Quoc Nguyen, Dai Quoc Nguyen, Dang Duc Pham, and Son Bao Pham

a Department of Computing, Macquarie University, Australia
E-mail: dat.nguyen@students.mq.edu.au
b Department of Computational Linguistics, Saarland University, Germany
E-mail: daiquocn@coli.uni-saarland.de
c L3S Research Center, University of Hanover, Germany
E-mail: pham@l3s.de
d University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
E-mail: sonpb@vnu.edu.vn

Abstract. This paper presents our new approach to construct a system of transformation rules for the Part-Of-Speech tagging task. Our tagging approach is based on an incremental knowledge acquisition methodology where rules are stored in an exception-structure and new rules are only added to correct errors of existing rules; thus allowing systematic control of the interaction between rules. Experiments on 13 languages exhibit that our method is fast in terms of training time and tagging speed. Furthermore, our method is able to attain state-of-the-art accuracies for relatively isolating or analytic languages while obtaining competitive accuracy results on morphologically rich Indo-European languages.

Keywords: Natural language processing, Part-Of-Speech tagging, Ripple Down Rules, RDRPOSTagger

1. Introduction

As one of the most important tasks in Natural Language Processing, Part-of-speech (POS) tagging is to assign a tag representing its lexical category to each word in a text [27]. After the text is tagged or annotated, it can be used in many applications such as machine translation, information retrieval, information extraction and the like.

Recently, statistical and machine learning-based POS tagging approaches have been mainstream methods obtaining state-of-the-art performance [4]. However, most of them are time-consuming in the learning process and require a powerful computer for training the tagging models. For instance, as reported in Mueller et al. [39], the POS taggers SVMTool [24] and CRF-Suite [46] correspondingly took 2,454 minutes (about 41 hours) and 9,274 minutes (about 155 hours) to be trained on a corpus of 38,727 Czech sentences (652,544 words), using a machine with two Hexa-Core Intel Xeon X5680 CPUs with 3.33 GHz and 6 cores each and 144 GB of memory. Therefore, such methods are not suitable for individuals having computing system with limited resources.

Turning to rule-based approaches, Brill [10] proposed the most well-known approach which is to automatically learn transformation rules for the POS tagging problem. In the Brill’s method, the learning process selects a new rule based on the context that is generated by all preceding rules, where the new rule can change the outputs of all the preceding rules. Hence, this method introduces a potential difficulty in controlling the interaction among a large number of rules.

In this paper, we present a new failure-driven approach to automatically restructure transformation rules in the form of a Single Classification Ripple Down Rules (SCRDR) tree [13, 51]. Our approach accepts the interaction between rules, but a rule only changes the outputs of some previous rules in a controlled context. All rules are structured in a SCRDR tree where
Figure 1.: A small part of our SCRDR tree for English POS tagging.

2. SCRDR methodology

In this section, we present the basic idea of Ripple Down Rules (RDR) \cite{13, 51} which inspired our approach. RDR allows one to add rules to a knowledge base incrementally without the need of a knowledge engineer. A new rule is only created when the knowledge base performs unsatisfactorily on a given case.

A SCRDR tree \cite{13, 51} is a binary tree with two distinct types of edges. These edges are typically called except and if-not edges. Associated with each node in a tree is a rule. A rule has the form: if \( \alpha \) then \( \beta \) where \( \alpha \) is called the condition and \( \beta \) is referred to as the conclusion.

To ensure that a conclusion is always given, the root node typically contains a trivial condition which is always satisfied. This node is called the default node. Default rule - the rule at the default node - is the unique rule which is not an exception rule of any other rule. Every rule in layer \( n \) is an exception rule of a rule in layer \( n - 1 \).

A new node containing a new exception rule is added to an SCRDR tree when the evaluation process returns a wrong conclusion. The new node is attached to the last node in the evaluation path of the given case with...
if word == “object.word” then tag = “correctTag”
if next1²Tag == “object.next1²Tag” then tag = “correctTag”
if prev1¹Tag == “object.prev1¹Tag” then tag = “correctTag”
if word == “object.word” && next1²Tag == “object.next1²Tag” then tag = “correctTag”
if prev1¹Tag == “object.prev1¹Tag” && next1²Tag == “object.next1²Tag” then tag = “correctTag”

Figure 2.: Several rule template examples.

the except link if the last node is the fired one. Otherwise, it is attached with the if-not link.

For illustration, with a SCRDR tree in Figure 1 given a case of 5-word window context “as/IN investors/NNS anticipate/VB a/DT recovery/NN” where “anticipate/VB” is the current word and POS tag pair, the case satisfies the conditions of the rules at nodes (0), (1) and (4), then it is passed to node (5) (following except links). As the case does not satisfy the condition of the rule at node (5), it will be transferred to node (8) using if-not link. Also the case does not satisfy the conditions of the rules at nodes (8) and (9). Therefore, we have the evaluation path (0)-(1)-(4)-(5)-(8)-(9) with the fired node (4). Thus, the POS tag for “anticipate” is concluded as “VBP”.

Rule (1) - the rule at node (1) - is the exception rule of the default rule (0). As node (2) is the if-not child node of node (1), rule (2) is also an exception rule of rule (0). Likewise, rule (3) is an exception rule of rule (0). Similarly, both rules (4) and (10) are exception rules of rule (1) whereas all rules (5), (8) and (9) are exception rules of rule (4), and so on. Therefore, the exception structure of the SCRDR tree extends to 4 levels: rules (1), (2) and (3) at layer 1; rules (4), (10), (11), (12) and (14) at layer 2; rules (5), (8), (9), (13) and (15) at layer 3; and rules (6) and (7) at layer-4 exception structure.

3. Our approach

In this section, we present our new failure-driven approach to automatically construct a SCRDR tree of transformation rules for POS tagging task. Figure 3 describes the learning process in our approach.

The initialized corpus is created by using the initial tagger to perform POS tagging on the raw corpus which comprises entire raw text extracted from the golden training corpus (golden corpus) excluding tags.

Our initial tagger is based on a lexicon to assign a tag for each word, in which the lexicon contains all words appearing in the golden corpus. Each entry in the lexicon consists of a unique word and its most frequent associated tag in the golden corpus.

To tag unknown-words (i.e. out-of-dictionary words) in English and Vietnamese, the initial tagger utilizes several regular expressions whereas the most frequent tag in the lexicon (i.e. the tag with highest number of different words associated) is exploited to label unknown-words when adapting to other languages.

By comparing the initialized corpus with the golden corpus, an object-driven dictionary containing pairs Object and correctTag is produced. Each Object matches a 5-word window context surrounding a word and its current tag in the format of (previous 2nd word / previous 2nd tag, previous 1st word / previous 1st tag, word / currentTag, next 1st word / next 1st tag, next 2nd word / next 2nd tag) extracted from the initialized corpus, while correctTag is the corresponding tag of the word in the golden corpus.

There are 27 rule templates used for the rule selector which selects the most suitable rules to build the SCRDR tree. Examples of our rule templates are shown in Figure 2 where the elements in bold will be replaced by concrete values from the Object and correctTag pairs in the object-driven dictionary for generating concrete rules.

The SCRDR tree of rules is initialized by building the default rule if True then tag = “TAG” as displayed in Figure 1. In order to build the rules at the layer-1
exception structure, for each concrete POS tag \textit{Label} in the list of all POS tags from the initialized corpus, our method creates a new rule in the form of \textit{if currentTag \text{==} \text{“Label”} then tag \text{=} \text{“Label”}}. Then the new rule will be added to the SCRDR tree as an exception rule of the default one, for example rules (1), (2) and (3) and the like in Figure 1.

3.1. Learning process

The learning process to construct new rules from the layer-2 exception structure to the tree is as follows:

- At a node \( \eta \) in the SCRDR tree, let \( \Theta \) be the set of Object and correctTag pairs from the object-driven dictionary for which every Object in \( \Theta \) is \textit{fired} at the node \( \eta \), however, the node \( \eta \) returns an incorrect POS tag (i.e. the POS tag concluded by the node \( \eta \) for the Object is not the corresponding correctTag).

- In order to select a new exception rule of the rule at the node \( \eta \) from all concrete rules which are generated for all Objects in \( \Theta \), the selected rule have to satisfy the following constraints: (i) If the node \( \eta \) is not at the layer-1 exception structure, the rule must not be satisfied by the Objects which the node \( \eta \) has already returned correct POS tags. (ii) The rule must associate to the highest score value of subtracting B from A in comparison to other ones, where A and B are the numbers of Objects in \( \Theta \) in which the rule produces correct and incorrect POS tags, respectively. (iii) The highest score value is not smaller than a given threshold.

Our method applies two threshold parameters: the first threshold is to find exception rules at the layer-1 exception structure for example, rules (3), (4) and (5) in Figure 1 while the second threshold is to find rules for higher exception layers.

- If the learning process could not select a new exception rule of the rule at the node \( \eta \), the process is repeated at another node \( \eta_p \) for which the rule at the node \( \eta \) is an exception rule of the rule at the node \( \eta_p \). Otherwise, the learning process is repeated at the new selected exception rule.

**Illustration:** To illustrate how new exception rules are appended to construct a SCRDR tree as shown in Figure 1, we start with node (1) to be the current node \( \eta \) associated to rule (1) \textit{if currentTag \text{==} \text{“VB” then tag \text{=} \text{“VB”}} at the layer-1 exception structure. The learning process produces the most suitable rule \textit{if prev1\textsuperscript{st}Tag \text{==} \text{“NNS” then tag \text{=} \text{“VBP”}} to be an exception rule of rule (1). Thus, node (4) associated to rule (4) \textit{if prev1\textsuperscript{st}Tag \text{==} \text{“NNS” then tag \text{=} \text{“VBP”}} is added as an except child node of node (1). The process then refers to node (4) as the node \( \eta \). Similarly, nodes (5) and (6) are appended to the tree as presented in Figure 1.

The node \( \eta \) now is node (6). At node (6), there is no concrete rule fulfilling the three constraints to be an exception rule of rule (6). Hence, the learning process is repeated at node (5) because rule (6) is an exception rule of rule (5).

At node (5), the learning process selects a new rule (7) \textit{if next\textsuperscript{1st}Word \text{==} \text{“into” then tag \text{=} \text{“VBP”}} to be another exception rule of rule (5). Consequently, a new node (7) containing rule (7) is added to the SCRDR tree as an if-not child node of node (6).

At node (7), the learning process could not find a new rule to be an exception rule of rule (7). Therefore, node (5) is again considered as the current node \( \eta \). The procedure to add new exception rules is repeated until there is no rule satisfying the three constraints.

3.2. Tagging process

The tagging process firstly tags unlabeled text by using the initial tagger. Next, it slides in a left-to-right direction on a 5-word window context to create a corresponding Object for each initially tagged word. The Object is then classified by the learned SCRDR tree model to produce the final tag of the word as illustrated in the example in section 2.

Noted that if the Object is fired at the root node of the tree model, the final tag is concluded as the initialized tag returned by the initial tagger.

4. Empirical study

This section presents the experiments validating our proposed approach in 13 languages. Experimental datasets are detailed in section 4.1 while training/tagging speed and accuracy results of our approach are described in sections 4.2 and 4.3 respectively.

4.1. Experimental setup

Evaluation for English used the Penn WSJ Treebank \textsuperscript{[35]} to exploit the sections 0-18 (38,219 sentences - 912,344 words) for training, the sections 19-21 (5,527 sentences - 131,768 words) for validation and the sections 22-24 (5,462 sentences - 129,654 words) for test, where the percentage of unknown-words in the test set to the training set is about 2.81% (3,649 unknown-
Aside from English, most previous published works have different experimental setups on various data splits. In addition, it is difficult to create the same evaluation settings used in the previous works. Therefore, excluding a 5-fold cross-validation evaluation scheme for Vietnamese, we examine performances based on the 10-fold cross-validation evaluation scheme on the datasets in 11 remaining languages.

**Training strategy:** We have two training strategies namely Open strategy and Leave-1 strategy:

- Open strategy: We use our initial tagger to label raw corpus. The initial tagger is based on a lexicon generated from the training set. During the open strategy-based learning process, all words are known as the lexicon contains all words in the training set. Therefore, the tagging model is not trained from errors over unknown-words. To prevent this, we follow another learning strategy of leave-1.

- Leave-1 strategy: The difference between this leave-1 strategy and the open strategy is that, in training phase, all words appearing only 1 time in the training set are initially tagged as tagging unknown-words as described in section 3.

**Threshold parameter:** By varying our learning approach’s threshold parameters on the English validation set, we have found the most suitable threshold pair of 3 and 2 for evaluating our approach on the English test set. Thus, we set the threshold pair of 3 and 2 as the default parameters utilized for all other experiments.

### 4.2 Training time and tagging speed

We run all experiments on the laptop computer of Window7 OS 64-bit, core i5 2.50GHz CPU and 6GB of memory. While most published works did not report the training time and the tagging speed, we present our single-threaded implementation results in Table 2. It is interesting to note that, in our approach, the training speed for learning combined POS and morphological (POS+MORPH) tagging model is faster than the one for POS tagging as presented in experimental results for French (11 minutes vs 19 minutes) and German (23 minutes vs 37 minutes). This is different from most machine learning-based approaches where the fewer number of tags, the higher training speed. For example, Mueller et al. [39] showed that on a dataset of 40,474 German sentences, the SVM-Tool took 899 minutes (about 15 hours) to be trained with 54 POS tags against 1,649 minutes (about 27 hours) accounted for 681 POS+MORPH tags.

In order to compare with existing POS taggers in terms of the training time, we show in Table 3 the time taken for training the SVMTool, CRFSuite, Morfette and RFTagger using an extremely powerful computer. For instance, using the same German TIGER corpus [8], our approach took an average time of 23 minutes (about 0.4 hours) to train with 54 POS tags against 1,649 minutes (about 27 hours) accounted for 681 POS+MORPH tags.

| Language     | Source                           | #sen | #words   | #P | #PM | OOV |
|--------------|----------------------------------|------|----------|----|-----|-----|
| Bulgarian    | BuTreeBank-Morph [60]            | 20,558 | 321,538 | —  | 564 | 8.22|
| Czech        | PDT Treebank 2.5 [5]             | 115,844 | 1,957,246 | —  | 1,570 | 4.71|
| Dutch        | Lassy Small Corpus [45]          | 65,200 | 1,096,177 | —  | 933  | 4.12|
| French       | French Treebank [1]              | 21,562 | 587,687  | 17 | 306  | 3.86|
| German       | TIGER Corpus [8]                 | 50,474 | 888,236  | 54 | 795  | 6.15|
| Hindi        | Hindi Treebank [49]              | 26,547 | 588,995  | 39 | —    | 0.00|
| Italian      | ISDT Treebank [7]                | 10,206 | 190,310  | 70 | —    | 7.66|
| Portuguese   | Tycho Brahe Corpus [20]          | 68,859 | 1,482,872 | —  | 344  | 2.83|
| Spanish      | IULA LSP Treebank [36]           | 42,099 | 589,542  | —  | 241  | 3.96|
| Swedish      | Stockholm—Umeå Corpus 3.0 [64]   | 74,245 | 1,166,593 | —  | 153  | 5.70|
| Thai         | ORCHID Corpus [62]               | 23,225 | 344,038  | 47 | —    | 2.83|
| Vietnamese   | (VTB) Vietnamese Treebank [44]   | 10,293 | 220,574  | 22 | —    | 3.41|
|              | (VLSP) VLSP Evaluation Campaign 2013 | 28,232 | 631,783  | 31 | —    | 2.06|

Table 1: The experimental datasets. #P: the number of POS tags. #PM: the number of combined POS and morphological (POS+MORPH) tags. OOV (Out-of-Vocabulary): the average percentage of unknown-words in each testing fold.

For each dataset, we split the dataset into 10 folds where the i-th sentence is selected for the (i%10)th fold. The evaluation procedure is repeated 10 times. Each fold is employed as the test set and 9 remaining folds are merged as the training set. All accuracy results are reported as the average results over the testing folds.
Table 2: The training time and tagging speed results. * indicates the results for POS+MORPH tagging. #rules: the average number of transformation rules in a learned tree model. EL: the average number of exception levels (rounded value) in the tree model. Tr.T: training time (minutes). Tg.S: tagging speed (number of words/tokens per second).

| Language      | Open strategy Tr.T | #rules | EL | Tg.S |
|---------------|---------------------|--------|----|------|
| Bulgarian*    | <1                  | 1,105  | 4  | 140K |
| Czech*        | 37                  | 14,494 | 5  | 45K  |
| Dutch*        | 31                  | 5,659  | 5  | 92K  |
| English       | 24                  | 2,319  | 4  | 215K |
| French        | 12                  | 987    | 4  | 140K |
| French*       | 7                   | 2,156  | 4  | 237K |
| German        | 18                  | 1,755  | 4  | 267K |
| German*       | 14                  | 10,576 | 5  | 92K  |
| Hindi         | 25                  | 3,041  | 4  | 282K |
| Italian       | 1                   | 631    | 4  | 275K |
| Portuguese*   | 32                  | 3,590  | 5  | 215K |
| Spanish*      | 3                   | 823    | 4  | 221K |
| Spanish       | 27                  | 3,841  | 4  | 158K |
| Thai          | 6                   | 1,355  | 4  | 280K |
| Vn (VTB)      | 8                   | 701    | 3  | 257K |
| Vn (VLSP)     | 36                  | 2,283  | 4  | 143K |

Table 3: The training time in minutes reported in Mueller et al. [39] for POS+MORPH tagging on a machine of two Hexa-Core Intel Xeon X5680 CPUs with 3.33 GHz and 6 cores each and 144 GB of memory. #sent: the number of sentences in training set. SVMT: SVMTool, Morf: Morfette, CRFS: CRFSuite, RFT: RFTagger (about 22 hours) and 1,649 minutes (about 27 hours), respectively.

Not taking the computer configuration into account, with almost the same experimental setting for German, our method is 56 times faster than the CRFSuite and up to 72 times faster than the SVMTool in training a POS+MORPH tagging model. Furthermore, we used much larger-in-size training datasets for Czech and Spanish and obtained fast training process.

Regarding to tagging speed, as reported in Moore [38] with the same evaluation scheme for English, on a workstation equipped with Intel Xeon X5550 2.67 GHz: the SVMTool, the UPenn bidirectional tagger [59], the COMPOST tagger [63] and the Stanford tagger [67] gained tagging speed at 7700, 270, 2600 and 80,000 tokens per second, respectively. In our experiment, we obtain a much faster tagging speed at 208,000 tokens per second on a weaker computer. To the best of our knowledge, we conclude that our method achieves fast training and tagging processes.

4.3. Accuracy Results

The accuracy results are detailed in Table 4. To sum up, our approach is able to reach state-of-the-art accuracies for English, Hindi, Thai and Vietnamese. It also achieves competitive accuracy results for Czech, French, German and Italian. The experimental results for remaining languages of Bulgarian, Dutch, Portuguese, Spanish and Swedish are promising.

Excluding Hindi where the unknown-word percentage (i.e. the OOV rate) is 0.0%, the accuracies obtained by the leave-1 strategy are higher than the accuracy results achieved by the open strategy on all other experimental languages. It is because in the open strategy-based training process, since all words are known, the tree models of rules are not learned from errors over unknown-words. Therefore, this leads to small improvements against the initial tagger in unknown-word tagging accuracies, especially for POS+MORPH tagging experiments.

On the contrary, the leave-1 strategy produces tagging models containing transformation rules learned on error contexts of unknown-words. Consequently, the leave-1 strategy obtains significant increases on unknown-word tagging accuracy results. Moreover, in most cases, there are no considerable differences between known-word tagging accuracies achieved by the two open and leave-1 strategies. This is because the selected pair of thresholds 3 and 2 tends to skip choosing concrete rules covering words appearing only 1 time in the training set.

4.3.1. English

With the same experimental setup, our approach achieves a leave-1 strategy-based accuracy at 96.51% against 96.46% accounted for the state-of-the-art POS tagger TnT [9]. The leave-1 strategy returns a SCRDR tree model of 2,418 rules in a 4-level exception structure. Table 4 presents the accuracy results obtained at exception levels. We only describe those results for English in Table 4 because of its non-cross-validation evaluation setup.

When using the outputs of the initial tagger developed in the Brill’s tagger [10], our approach gains an

---

4The initial tagging result on the test set is 93.58%.
Table 4: The accuracy results (%). * indicates the accuracies on combined POS+MORPH tags. **Known**: the known-word tagging accuracy. **Unk.**: the unknown-word tagging accuracy. **All.**: the overall accuracy result.

| Language   | Initial accuracy | Open strategy | Leave-1 strategy |
|------------|------------------|---------------|------------------|
|            | Known | Unk. | All. | Known | Unk. | All. | Known | Unk. | All. |
| Bulgarian  | 95.61 | 4.14 | 88.09 | 96.98 | 4.15 | 89.35 | 96.83 | 23.62 | **90.81** |
| Czech      | 84.43 | 6.08 | 80.74 | 92.84 | 6.42 | 88.76 | 92.89 | 25.44 | **89.71** |
| Dutch      | 89.40 | 19.67 | 86.53 | 94.54 | 20.41 | 91.48 | 94.47 | 34.87 | **92.01** |
| English    | 93.94 | 78.84 | 93.51 | 96.90 | 82.35 | 96.49 | 96.90 | 82.90 | **96.51** |
| French     | 96.09 | 49.06 | 94.27 | 98.15 | 55.94 | 96.52 | 98.09 | 69.81 | **97.00** |
| French*    | 90.19 | 10.22 | 87.10 | 95.14 | 19.22 | 92.21 | 95.13 | 42.51 | **93.09** |
| German     | 94.88 | 54.40 | 92.39 | 97.83 | 54.55 | 95.16 | 97.60 | 67.81 | **95.77** |
| German*    | 71.68 | 4.86 | 67.56 | 86.55 | 13.06 | 82.03 | 87.12 | 34.14 | **83.86** |
| Hindi      | 90.08 | — | 90.08 | 96.35 | — | **96.35** | 96.32 | — | 96.32 |
| Italian    | 93.43 | 27.72 | 88.40 | 96.33 | 32.36 | 91.43 | 96.09 | 59.21 | **93.26** |
| Portuguese | 93.17 | 10.15 | 90.82 | 96.50 | 14.74 | 94.19 | 96.50 | 31.12 | **94.65** |
| Spanish*   | 98.04 | 13.31 | 94.69 | 98.97 | 14.07 | 95.61 | 98.87 | 39.41 | **96.51** |
| Swedish*   | 91.23 | 15.88 | 86.94 | 96.43 | 17.50 | 91.94 | 96.41 | 35.37 | **92.93** |
| Thai       | 92.60 | 71.77 | 92.01 | 95.66 | 73.11 | 95.02 | 95.66 | 76.32 | **95.11** |
| Vn (VTB)   | 92.14 | 46.55 | 90.59 | 94.08 | 47.33 | 92.48 | 94.07 | 50.71 | **92.59** |
| (VLSP)     | 91.89 | 57.36 | 91.18 | 94.13 | 58.11 | 93.38 | 94.12 | 59.77 | **93.42** |

Table 5: Results (%) due to levels of exception structures.

| Level | Number of rules | Accuracy |
|-------|-----------------|----------|
| <= 1  | 47              | 93.51    |
| <= 2  | 1,450           | 96.33    |
| <= 3  | 2,384           | 96.50    |
| <= 4  | 2,418           | 96.51    |

4.3.2. Bulgarian

On the 10-fold cross validation evaluation scheme using the BulTreeBank [60], we achieve an encouraging accuracy of 90.81%.

Given in the BulTreeBank’s webpage[6] about Bulgarian POS+MORPH tagging experiments on the BulTreeBank, the TnT tagger, the SVMTool and the memory-based tagger in the Acopost package[7] obtained accuracies of 92.53%, 92.22% and 89.91%, respectively.

In addition, Georgiev et al. [22] obtained an accuracy result at 95.72% for POS+MORPH tagging without utilizing external resources.

4.3.3. Czech

Mueller et al. [39] showed accuracies of the four POS taggers SVMTool, CRFSuite, RFTagger and Morfette on POS+MORPH tagging for Czech. The four taggers were trained using a training set of 38,727 sentences (652,544 tokens) and evaluated on a test set of 4,213 sentences (70,348 tokens) extracted from the Prague Dependency Treebank 2.0. The accuracy results correspondingly are 89.62%, 90.97%, 90.43% and 90.01% accounted for the SVMTool, CRFSuite, RFTagger and Morfette.

Since we could not access the Czech datasets used in the experiment above, we employ the Prague Dependency Treebank 2.5 of about 116K sentences. We obtain a competitive accuracy of 89.71% which is comparable to the result of the SVMTool.

4.3.4. Dutch

van den Bosch et al. [70] used a manually-annotated POS tagged corpus of 11 million Dutch words with 316 tags to develop the TADPOLE tagger, where the corpus was split at the sentence level into a 90% training set and a 10% test set. The TADPOLE tagger reached an accuracy result of 96.5%.

---

[5] We retrained the Brill’s tagger for English with the default threshold of 2 on the same evaluation scheme.

[6] http://www.bultreebank.org/taggers/taggers.html

[7] http://acopost.sourceforge.net

[8] http://opennlp.sourceforge.net

[9] Georgiev et al. [22] split the BulTreeBank corpus into training set of 16,532 sentences, development set of 2,007 sentences and test set of 2,017 sentences.
Due to limited access the 11 million-word corpus, we use the Lassy Small Corpus [45] of about 1.1 million words with 933 tags. We achieve a favorable accuracy at 92.01%.

4.3.5. French

Current state-of-the-art methods for French POS tagging attained accuracies around 97.75% [58, 16] when using the French Treebank [1] with 9,881 sentences for training and 1,235 sentences for test. However, these methods employed external large-scale morphological lexicons extracted from the Lefff [53]. Without using the lexicons, Denis and Sagot [16] reported an accuracy performance at 97.00%.

On the French Treebank which we are granted containing 21,562 POS+MORPH tagged sentences, we get a very promising accuracy at 97.00% for French POS tagging. Regarding to POS+MORPH tagging, as far as we know that this is the first experimental evaluation for French where we obtain an accuracy of 93.09%.

4.3.6. German

Using 10-fold cross validation evaluation scheme on the TIGER corpus [8] of 50,474 German sentences, Giesbrecht and Evert [23] showed that the five taggers of TreeTagger [55], TnT tagger, STanford tagger [67], and Apache UIMA Tagger obtained the POS tagging accuracies of 96.89%, 96.92%, 97.12%, 97.63% and 96.04%, respectively. In the same evaluation setting, we obtain a very considerable accuracy result of 95.77%. Furthermore, we achieve a known-word tagging accuracy of 97.60% which is as high as the known-word tagging accuracies computed for the taggers TreeTagger (97.62%), TnT (97.59%), SVMTool (97.71%) and Apache UIMA Tagger (97.50%).

Turning to POS+MORPH tagging, Mueller et al. [39] also performed experiments on the TIGER corpus using 40,474 sentences for training and 5,000 sentences for test. They presented accuracy performances of 83.42%, 85.68%, 84.28% and 83.48% accounted for the four taggers SVMTool, CRFSuite, RFTagger and Morfette, respectively. In our evaluation scheme, we gain a very promising accuracy at 83.86%.

4.3.7. Hindi

We reach an up-to-date highest accuracy result at 96.35% for Hindi POS tagging evaluated on the Hindi Treebank [49]. As one of the most popular language in the world, there are many research works on POS tagging for Hindi. However, there is no publicly standard dataset. Previous works used locally manually-labeled datasets which are smaller in size than the Hindi Treebank, and those datasets are currently not available.

Joshi et al. [29] achieved an accuracy of 92.13% accounted for a Hidden Markov Model-based approach, using a dataset of 358K words for training and a test set of 12K words for test. Conducting experiments in using another training set of 150K words and test set of 40K words, Agarwal et al. [2] compared machine learning-based approaches and presented the highest POS tagging accuracy at 93.70% reached by the Conditional Random Fields-based method.

In the 2007 Shallow Parsing Contest for South Asian Languages [6], the POS tagging track provided a small training set of 21,470 words and a test set of 4,924 words. The highest accuracy in the contest was 78.66% obtained by Avinash and Karthik [4]. Employing a dataset of 15,562 words with 27 POS tags, Singh et al. [61] reported an average accuracy of 93.45% whilst Dalal et al. [15] achieved a result at 94.38% in the same 4-fold cross-validation evaluation scheme.

4.3.8. Italian

In the EVALITA 2009 workshop on Evaluation of NLP and Speech Tools for Italian [11] the POS tagging track [3] provided a training set of 3,719 sentences (108,874 word forms) with 37 POS tags. The participating teams in the closed task of not using external resources achieved various tagging accuracies on a test set of 147 sentences (5,066 word forms) ranging from 93.21% to 96.91%.

Our experiment on Italian POS tagging employs the ISDT Treebank [7] of 10,206 sentences (190,310 word forms) with 70 POS tags. 12 In 70 tags, there are 33 tags containing morphological information. The total number of words associated to those 33 tags is 108.

Our experiment on Italian POS tagging employs the ISDT Treebank [7] of 10,206 sentences (190,310 word forms) with 70 POS tags. We obtain a competitive accuracy performance at 93.26%.

4.3.9. Portuguese

The previous works [17, 30] on Portuguese POS tagging used an early version of the Tycho Brahe Corpus [20] containing of about 1,036K words. The corpus was split into a training set of 776K words and a test set of 260K words. While Kepler and Finger [30] achieved an accuracy of 95.51%, dos Santos et al. [17] reached a state-of-the-art accuracy result at 96.64%.

The Tycho Brahe Corpus collected in our work consists of about 1,639K words. We reach a favorable accuracy performance at 94.65% using 10-fold cross validation evaluation scheme.

[10]https://uima.apache.org/sandbox.html#tagger.annotator

11http://www.evalita.it/2009
12In 70 tags, there are 33 tags containing morphological information. The total number of words associated to those 33 tags is 108.
4.3.10. Spanish

In addition to Czech and German, Mueller et al. [39] evaluated the taggers of SVMTool, CRFSuite, RFTagger and Morfette for Spanish POS+MORPH tagging, employing a training set of 14,329 sentences (427,442 tokens) and a test set of 1,725 sentences (50,630 tokens). The obtained accuracies of the four taggers ranged from 97.35% to 97.60%.

As we could not have the training and test sets above, we use the IULA Spanish LSP Treebank [36] consisting of 42K sentences with 241 tags. We achieve an encouraging accuracy result of 96.51%.

4.3.11. Swedish

On the same SUC corpus 3.0 [64] consisting of 500 text files with about 74K sentences, Östling [47] evaluated the Swedish POS tagger namely Stagger according to 10-fold cross validation scheme. It is different that Östling [47] split the SUC corpus 3.0 into 10 folds at file level instead of sentence level. The Stagger attained an accuracy of 96.06% which is 3.13% higher than our accuracy of 92.93%. However, we obtain a promising known-word tagging accuracy result at 96.41% in comparison to the known-word tagging accuracy of 97.12% accounted for Stagger.

4.3.12. Thai

On the Thai POS Tagged corpus ORCHID [62] of 23,225 sentences, we achieve an accuracy of 95.11%.

It is difficult to directly compare our research work with previous work for Thai POS tagging, for example the previous works [34, 40] performed their experiments on an unavailable corpus of 10,452 sentences. The ORCHID corpus was also used in a POS tagging experiment presented by Krueangkrai et al. [31], however, the achieved accuracy result of 79.342% was evaluated in depending on the performance of automatic word segmentation. Pailai et al. [48] reached an accuracy of 93.64% using 10-fold cross validation on another corpus of 100K words.

4.3.13. Vietnamese

We participated the first VLSP evaluation campaign [5] on Vietnamese language processing. The campaign’s POS tagging track provided a training set of 28,232 sentences and a raw test set of 2,130 sentences. Our learning process returned a SCRDR tree model of 2,896 rules. With the obtained accuracy at 95.31% for tagging the test set, we achieved the 1st place in the POS tagging track.

We also carry out POS tagging experiments on a 5-fold cross-validation evaluation scheme using the VLSP set of 28,232 sentences and the standard benchmark Vietnamese Treebank [44] of about 10K sentences. The average accuracy results are presented in Table 4 Achieving an accuracy of 92.59% on the Vietnamese Treebank, our approach outperforms previous Maximum Entropy Model, Conditional Random Fields and Support Vector Machine-based POS tagging approaches [68] in the same evaluation scheme.

4.4. Discussion

As can be seen from Table 4, unknown-word tagging accuracy performances are especially low on POS+MORPH tagging for morphologically rich languages such as Bulgarian, Czech, Dutch, French, German, Portuguese, Spanish and Swedish. The reason is that as presented in section 3, we use a simple heuristic of using the most frequent tag in the lexicon for initially tagging unknown-words. However, similar to most of previous research works, our approach could utilize external resources, regular-expressions and existing morphological analyzers to enhance initial results. Thus, our approach could achieve higher accuracies.

The POS tagging accuracy results on Vietnamese, Thai, Hindi and English lead to a believe that our approach enables to reach state-of-the-art performances on other relatively isolating or analytic languages. In addition to Czech, French, German and Italian, we could obtain very competitive results on other morphologically rich languages.

An important point is that our approach is suitable to use experts to add new exception rules given a concrete case that is misclassified by the tree model. This is especially important for under-resourced languages where obtaining a large annotated corpus is difficult.

5. Related work

Early POS tagging approaches are rule-based ones in which the transformation-based learning approach...
proposed by Brill \cite{10} is the most well-known rule-based method.

The key idea of the Brill’s approach is to compare a golden corpus which is correctly annotated manually with an initialized corpus which is generated by executing an initial tagger on the corresponding raw corpus. Based on predefined rule templates, the approach then automatically produces list of concrete rules to correct wrong POS tags. For example, corresponding with a template “transfer tag of current word from A to B if the next word is W” is some concrete rules such as “transfer tag of current word from JJ to NN if the next word is of” or “transfer tag of current word from VBD to VBN if the next word is by”.

At each iteration during the Brill’s learning process, all concrete rules will be generated and the rule selection is computed based on all incorrect POS tags in the entire training corpus. Therefore, the Brill’s training process takes a significant amount of time. To prevent that, Hepple \cite{28} presented an approach with two assumptions for disabling interactions between rules to reduce the training time while sacrificing a small fall of accuracy. Ngai and Florian \cite{41} proposed a method to reduce the training time by recalculating the score of transformation rules while keeping the accuracy result.

The difference between our approach and the Brill’s is that we construct transformation rules in form of SCRD method where a new transformation rule is produced based on only a subset of POS tagging errors. Hence, our approach obtains faster performances in term of the training speed. In the first version of our approach’s conference publication \cite{42}, we reported an improvement up to 33 times in training speed in comparison with the Brill’s method. Besides, in the Brill method, a following rule enables to change the outputs of all preceding rules, so a word can be tagged multiple times in the tagging process. This is different from our approach where a word is tagged only one time.

In addition to our research, there is only another research applying SCRD method for POS tagging described by Xu and Hoffmann \cite{71}. Though Xu and Hoffmann’s approach obtained a competitive accuracy, it is a hand-crafted approach taking about 60 hours to manually build a SCRD tree model for English POS tagging.

Turning to statistical and machine learning approaches for POS tagging, these methods can be listed as various Markov Models-based methods \cite{37, 65, 9, 23, 19}, maximum entropy-based methods \cite{50, 66, 67, 59, 11}, perceptron algorithm-based approaches \cite{12, 59, 63}, hybrid methods \cite{26, 18}, and other machine learning-based approaches including neural networks \cite{54}, decision trees \cite{55}, finite-state transducers \cite{52}, memory-based learning \cite{14}, Support Vector Machines \cite{24} and Conditional Random Fields \cite{32, 33, 39}. An overview about POS tagging could be found in \cite{27}.

### References

1. A. Abeillé, L. Clément, and F. Toussenel. Building a Treebank for French. In *Treebanks*, volume 20 of *Text, Speech and Language Technology*, pages 165–187. http://www.lif.cnrs.fr/Gens/Abeille/Frenc. 2003.
2. M. Agarwal, R. Goutam, A. Jain, S. R. Kesidū, P. Kosaraju, S. Muktyar, B. Ambati, and R. Sangal. Comparative Analysis of the Performance of CRF, HMM and MaxEnt for Part-of-Speech Tagging, Chunking and Named Entity Recognition for a Morphologically Specific Association for Computational Linguistics, pages 3–6, 2011.
[3] G. Attardi and M. Simi. Overview of the EVALITA 2009 Part-of-Speech Tagging Task. In Poster and Workshop Proc. of the 11th Conference of the Italian Association for Artificial Intelligence, 2009.

[4] P. Avinesh and G. Karthik. Part-Of-Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning. In Proceedings of IJCAI 2007 Workshop on Shallow Parsing for South Asian Languages, 2007.

[5] E. Bejcek, J. Panevová, J. Popelka, P. Stanáč, M. Sevekiová, J. Stepánek, and Z. Zabokrtský. Prague Dependency Treebank 2.5 - a Revisited Version of PDT 2.0. In Proceedings of 24th International Conference on Computational Linguistics, pages 231–246, 2012.

[6] A. Bharathi and P. R. Mannem. Introduction to the Shallow Parsing Contest for South Asian Languages. In Proceedings of IJCAI 2007 Workshop on Shallow Parsing for South Asian Languages, 2007.

[7] C. Bosco, S. Montemagni, and M. Simi. Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank. In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pages 61–69, 2013.

[8] S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. Research on Language and Computation, 2(4):597–620, http://www.ims.uni-stuttgart.de/forschung, 2004.

[9] T. Brants. TnT: a statistical part-of-speech tagger. In Proceedings of the Sixth Applied Natural Language Processing Conference, pages 224–231, 2000.

[10] E. Brill. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. Computational Linguistics, 21(4):543–565, 1995.

[11] G. Chrupala, G. Dinu, J. van Genabith, G. Chrupala, and J. van Genabith. Learning morphology with morfette. In Proceedings of the 6th International Conference on Language Resources and Evaluation, pages 2362–2367, 2008.

[12] M. Collins. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1–8, 2002.

[13] P. Compton and R. Jansen. A philosophical basis for knowledge acquisition. Knowledge Acquisition, 2(3):241–257, 1990.

[14] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. MBT: A memory-based part of speech tagger generator. In Proceedings of the Fourth Workshop on Very Large Corpora, pages 14–27, 1996.

[15] A. Dalal, K. Nagaraj, U. Sawant, S. Shelke, and P. Bhatucharyya. Building Feature Rich POS Tagger for Morphologically Rich Languages: Experiences in Hindi. In Proceedings of the 5th International Conference on Natural Language Processing, 2007.

[16] P. Denis and B. Sagot. Coupling an Annotated Corpus and a Lexicon for State-of-the-art POS Tagging. Lang. Resour. Eval., 46:721–736, 2012.

[17] C. N. dos Santos, R. L. Milidii, and R. P. Rentería. Portuguese Part-of-Speech Tagging Using Entropy Guided Transformation Learning. In Proceedings of the 8th International Conference on the Computational Processing of Portuguese, pages 143–152, 2008.

[18] R. Forsati and M. Shamsfard. Hybrid PoS-tagging: A cooperation of evolutionary and statistical approaches. Applied Mathematical Modelling, 38(13):3193–3211, July 2014.

[19] M. Fruzangohar, T. a. Kroeger, and D. L. Adelson. Improved part-of-speech prediction in suffix analysis. PloS one, 8(10):e76042, Jan. 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0076042.

[20] C. Galves and P. Faria. Tycho Brahe Parsed Corpus of Historical Portuguese. http://www.tycho.iel.unicamp.br/~tycho, 2010.

[21] G. Georgiev, P. Nakov, P. Osenova, and K. Simov. Cross-lingual Adaptation as a Baseline : Adapting Maximum Entropy Models to Bulgarian. In Proceedings of the Workshop Adaptation of Language Resources and Technology to New Domains 2009, pages 35–38, 2009.

[22] G. Georgiev, V. Zhikov, P. Osenova, K. Simov, and P. Nakov. Feature-Rich Part-of-speech Tagging for Morphologically Complex Languages: Application to Bulgarian. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 492–502, 2012.

[23] E. Giesbrecht and S. Evert. Is Part-of-Speech Tagging a Solved Task ? An Evaluation of POS Taggers for the German Web as Corpus. In Proceedings of the Fifth Web as Corpus Workshop, pages 27–36, 2007.

[24] J. Giménez, L. Márquez, and L. Marquez. Svmtool: A general pos tagger generator based on support vector machines. In Proceedings of the 4th International Conference on Language Resources and Evaluation, pages 43–46, 2004.

[25] S. Goldwater and T. L. Griffith. A fully Bayesian approach to unsupervised part-of-speech tagging. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 744–751, 2007.

[26] J. a. V. Grac¸a, K. Ganchev, L. Coheur, F. Pereira, and B. Taskar. Controlling Complexity in Part-of-Speech Induction. Journal of Artificial Intelligence Research, 41:527–551, 2011.

[27] T. Güngör. Part-of-Speech Tagging. In Handbook of Natural Language Processing, second edition, pages 205–235, Chapman & Hall/CRC, 2010.

[28] M. Hepple. Independence and Commitment: Assumptions for Rapid Training and Execution of Rule-based POS Taggers. In Proceedings of 38th Annual Meeting of the Association for Computational Linguistics, pages 277–278, 2000.
[29] N. Joshi, H. Darbari, and I. Mathur. HMM based POS tagger for Hindi. In Proceedings of 2nd International Conference on Artificial Intelligence and Soft Computing, pages 341–349, 2013.

[30] F. N. Kepler and M. Finger. Comparing Two Markov Methods for Part-of-speech Tagging of Portuguese. In Proceedings of the 2nd International Joint Conference of IBERAMIA 2006 and SBIA 2006, pages 482–491, 2006.

[31] C. Kruengkrai, V. Sornlertlamvanich, and H. Isahara. A Conditional Random Field Framework for Thai Morphological Analysis. In Proceedings of the Fifth Int. Conf. on Language Resources and Evaluation, pages 2419–2424, 2006.

[32] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the 18th International Conference on Machine Learning, pages 282–289, 2001.

[33] T. Lavergne, O. Cappé, and F. Yvon. Practical Very Large Scale CRFs. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 504–513, 2010.

[34] Q. Ma, M. Murata, K. Uchimoto, and H. Isahara. Hybrid Neuro and Rule-Based Part of Speech Taggers. In Proceedings of the 18th conference on Computational linguistics - Volume 1, pages 509–515, 2000.

[35] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: the penn treebank. Comput. Linguist., 19(2):313–330, 1993.

[36] M. Marimon, B. Fisas, N. Bel, M. Villegas, J. Vivaldi, S. Torner, M. Lorente, and S. Vázquez. The IULA Treebank. In Proceedings of the eighth international conference on Language Resources and Evaluation, pages 1920–1926, http://hdl.handle.net/10230/20048, 2012.

[37] B. Merialdi. Tagging English text with a probabilistic model. Computational linguistics, 20(2):155–171, 1994.

[38] R. Moore. Fast High-Accuracy Part-of-Speech Tagging by Independent Classifiers. In Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, pages 1165–1176, 2014.

[39] T. Mueller, H. Schmid, and H. Schütze. Efficient Higher-Order CRFs for Morphological Tagging. In Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing, pages 322–332, 2013.

[40] M. Murata, Q. Ma, and H. Isahara. Comparison of three machine-learning methods for Thai part-of-speech tagging. ACM Transactions on Asian Language Information Processing, 1(2):145–158, 2002.

[41] G. Ngai and R. Florian. Transformation-based learning in the fast lane. In Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics, pages 1–8, 2001.

[42] D. Q. Nguyen, D. Q. Nguyen, S. B. Pham, and D. D. Pham. Ripple Down Rules for Part-of-Speech Tagging. In Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics - Volume Part I, pages 190–201, 2011.

[43] D. Q. Nguyen, D. Q. Nguyen, D. D. Pham, and S. B. Pham. RDRPOSTagger: A Ripple Down Rules-based Part-Of-Speech Tagger. In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 17–20, 2014.

[44] P. T. Nguyen, X. L. Vu, T. M. H. Nguyen, V. H. Nguyen, and H. P. Le. Building a Large Syntactically-Annotated Corpus of Vietnamese. In Proceedings of the Third Linguistic Annotation Workshop, pages 182–185, Aug. 2009.

[45] G. Noord, G. Bouma, F. Eynde, D. Kok, J. Linde, I. Schuurman, E. Sang, and V. Vandeghinste. Large Scale Syntactic Annotation of Written Dutch: Lassy. In Essential Speech and Language Technology for Dutch, Theory and Applications of Natural Language Processing, pages 147–164, 2013.

[46] N. Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs), 2007.

[47] R. Östling. Stagger: an Open-Source Part of Speech Tagger for Swedish. Northern European Journal of Language Technology, 3:1–18, 2013.

[48] J. Pailai, R. Kongkachandra, T. Supnithi, and P. Boonkwan. A comparative study on different techniques for Thai part-of-speech tagging. In Proc. of 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pages 1–5, 2013.

[49] M. Palmer, R. Bhatt, B. Narasimhan, O. Rambow, D. M. Sharma, and F. Xia. Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure. In Proc. of 7th International Conference on Natural Language Processing, pages 261–268, 2009.

[50] A. Ratnaparkhi. A Maximum Entropy Model for Part-Of-Speech Tagging. In Proceedings of the fourth Workshop on Very Large Corpora, pages 133–142, 1996.

[51] D. Richards. Two decades of ripple down rules research. Knowledge Engineering Review, 24(2):159–184, 2009.

[52] E. Roche and Y. Schabes. Deterministic part-of-speech tagging with finite-state transducers. Computational Linguistics, 21(2):227–253, 1995.

[53] B. Sagot, L. Clément, E. V. d. L. Clergerie, and P. Boullier. The Lefff 2 syntactic lexicon for French: architecture, acquisition, use. In Proceedings of the 5th Language Resource and Evaluation Conference, pages 1348–1351, 2006.

[54] H. Schmid. Part-Of-Speech Tagging with Neural Networks. In Proceedings of the 15th International Conference on Computational Linguistics, Volume 1, pages 172–176, 1994.
[55] H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In Proceedings of International Conference on New Methods in Language Processing, pages 44–49, 1994.

[56] H. Schmid and F. Laws. Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-grained POS Tagging. In Proceedings of 22nd International Conference on Computational Linguistics, pages 777–784, 2008.

[57] I. Schroder. A case study in part-of-speech tagging using the ICOPOST toolkit. Technical report, Department of Computer Science, University of Hamburg, 2002.

[58] D. Seddah, G. Chrupała, O. Cetinoglu, J. van Genabith, and M. Candito. Lemmatization and Lexicalized Statistical Parsing of Morphologically Rich Languages: the Case of French. In Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, number June, pages 85–93, 2010.

[59] L. Shen, G. Satta, and A. Joshi. Guided Learning for Bidirectional Sequence Classification. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 760–767, 2007.

[60] K. Simov, P. Osenova, A. Simov, and M. Kouylekov. Design and Implementation of the Bulgarian HPSG-based Treebank. Research on Language and Computation, 2:495–522. http://www.bultreebank.org/bbtmorf/, 2004.

[61] S. Singh, K. Gupta, M. Shrivastava, and P. Bhattacharyya. Morphological richness offsets resource demand-experiences in constructing a POS tagger for Hindi. In Proceedings of the COLING/ACL on Main conference poster sessions, pages 779–786, 2006.

[62] V. Sornlertlamvanich, T. Charoenporn, and H. Ishihara. ORCHID: Thai Part-Of-Speech Tagged Corpus. http://culturelab.in.th/files/orchid.html. http://culturelab.in.th/files/orchid97-tr.pdf, 1997.

[63] D. j. Spoustová, J. Hajič, J. Raab, and M. Spousta. Semi-supervised Training for the Averaged Perceptron POS Tagger. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, number April, pages 763–771, 2009.

[64] SUC-3.0. The Stockholm—UmeaCorpus (SUC) 3.0. http://sprakbanken.gu.se/eng/resource/suc3, 2012.

[65] S. M. Thede and M. P. Harper. A Second-Order Hidden Markov Model for Part-of-Speech Tagging. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics, pages 175–182, 1999.

[66] K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora, pages 63–70, 2000.

[67] 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- Volume 1, pages 173–180, 2003.

[68] O. T. Tran, C. A. Le, T. Q. Ha, and Q. H. Le. An Experimental Study on Vietnamese POS Tagging. In Proceedings of 2009 International Conference on Asian Language Processing, pages 23–27, 2009.

[69] Y. Tsuruoka and J. Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language, pages 467–474, 2005.

[70] A. van den Bosch, B. Busser, S. Canisius, and W. Daelemans. An efficient memory-based morphosyntactic tagger and parser for Dutch. In Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands, pages 191–206, 2007.

[71] H. Xu and A. Hoffmann. RDRCE: combining machine learning and knowledge acquisition. In Proceedings of the 11th international conference on Knowledge management and acquisition for smart systems and services, pages 165–179, 2010.