

# Increasing the reliability of a part-of-speech tagging tool for use with learner language

Sylvie Thouësny\*  
Dublin City University  
sylvie.thouesny@icall-research.net

***Abstract:** The use of a part-of-speech (pos) tagger for L2 learner language is likely to result in tagging errors due to incorrect forms produced by learners. Since errors in pos-tagging will result in larger errors in the analysis of incorrect grammatical or lexical forms, it is essential to encode all components in a given text with robust and consistent tags. This article proposes a method to increase the reliability of part-of-speech tagging tools for tagging a corpus of language learners. Following a description of the creation and annotation of a learner language corpus, it explicates how the tagging accuracy of TreeTagger was increased by means of (a) identifying the lemmas (dictionary entry forms) that were unknown for the tagger, (b) checking the part-of-speech tags automatically obtained against an extended set of common-sense rules based on recurrent tagging errors, and (c) cross-referencing the part-of-speech tags with the error-encoded tags. TreeTagger's accuracy, before and after improvements, was evaluated through the means of not only recall and precision measures, but also two coefficients using distinct ways of computing the expected agreement (PE), namely the Krippendorff's alpha and Cohen's kappa coefficients. Bearing in mind that the input text was real-life input from language learners of French, the results clearly showed an increase in terms of tagging accuracy.*

## 1. Introduction

The part-of-speech tagging process is defined as the action of assigning the most reliable part-of-speech tag (e.g., noun, verb, adjective, adverb, ...) to not only each word in a learner's text, but also each punctuation mark, symbol or abbreviation, in other words, each token in the input (Jurafsky and Martin 2000). One issue with the tagging procedure however is that one token may be eligible for more than one part-of-speech tag. Several techniques exist to disambiguate a word class in order to apply the part-of-speech tag adapted to the situation. Since Greene and Rubin's (1971) TAGGIT, initially used to tag the Brown Corpus, taggers in terms of accuracy have sufficiently improved to be worthy of attention. The Brown Corpus is a collection of American English texts compiled and printed by Kučera and Francis at the Brown University in 1961 (Pradhan, Ward, and Martin 2008).

Taggers can be classified into four main groups depending on the technique they use to disambiguate linguistic units: Linguistic, machine-learning, statistical and hybrid models (Márquez, Padró, and Rodríguez 2000). For example, within the linguistic

---

\* **How to cite this article:** Thouësny, S. (2011). *Increasing the reliability of a part-of-speech tagging tool for use with learner language*. Proceedings from Pre-conference (AALL'09) workshop on automatic analysis of learner language: from a better understanding of annotation needs to the development and standardization of annotation schemes, 10-MAR-09 - 11-MAR-09, Arizona State University, Tempe, AZ.

approach, linguists manually formalise the grammar knowledge as a set of descriptive rules or constraints that characterises the syntactic properties of a given corpus. This set of logical rules, capable of generating an infinite number of possible sentences along with their structural specification, is then used to assign an appropriate description to each sentence in the corpus (Greene and Rubin 1971; Karlsson 1995). Within a machine-learning approach, the aim is “to automatically induce a model for some domain, given some data from the domain” (Jurafsky and Martin 2000, p. 118). There are a few taggers for the English language employing machine-learning techniques, such as the markov model (Cutting et al. 1992), the transformation-based error-driven learning (Brill 1995), or the decision trees (Black et al. 1992; Màrquez, Padró, and Rodríguez 2000; Schmid 1994). Statistical methods have focused on probabilities using stochastic algorithms to “pick the most-likely tag” for each word in a text (Jurafsky and Martin 2000, p. 303). Examples of stochastic taggers are illustrated in the work of Marshall (1983) or more recently Faaß et al. (2009). Finally, other taggers called hybrids use a combination of two or more techniques, such as rule-based and machine learning methods in the case of Tlili-Guiassa’s (2006) tagger used for the Arabic language.

Generally, taggers achieve excellent scores in terms of tagging accuracy, where a typical success rate evolves above 95% (Granger 2002, p. 18). However, as noted by Díaz-Negrillo et al. (2010, p. 4), “[w]hen applied to a new genre of text, taggers perform worse than they applied to the genre they were developed for”. Since taggers are generally trained with texts written by native speakers, tagging a text performed by a second language learner might be problematic. Indeed, the tagger will encounter potentially ambiguous words due to misspellings, foreign words, proper nouns, invented words or other unknown words, not to mention incorrect syntactic sentences. As stated by Granger (2003, p. 466), the particularity of learner language is that it reflects “a high rate of misuse, i.e., orthographic, lexical, and grammatical errors”. As a result, “post-correction steps are usually added to modify tags that are systematically wrongly assigned” (Díaz-Negrillo et al. 2010, p. 4). To improve the tagging process, this article proposes a three-step approach. In particular, it shows how the tagging accuracy of TreeTagger was increased when processing language learners’ ill-formed written productions by means of (a) identifying unknown lemmas, (b) checking automatically assigned part-of-speech tags against an extended set of rules based on recurrent tagging errors, and (c) cross-referencing the part-of-speech tags with the error-encoded tags. Evaluation results are then presented and future directions for research are discussed.

## 2. Methodology

### 2.1 Context

The intention of improving the part-of-speech tag accuracy was motivated by this author’s doctoral research whose primary interest was to analyse variability in a part-of-speech tagged corpus (Thouësnny Forthcoming). The corpus contains a collection of texts written by intermediate language learners of French. Participants were studying at university level and submitted their texts on a voluntary basis during the first semester of the 2008/2009 academic calendar. As the tagging accuracy reflected a fundamental requirement, it was essential to ensure that all tokens were marked with robust part-of-speech tags. TreeTagger<sup>1</sup>, the software module used to part-of-speech tag this corpus,

---

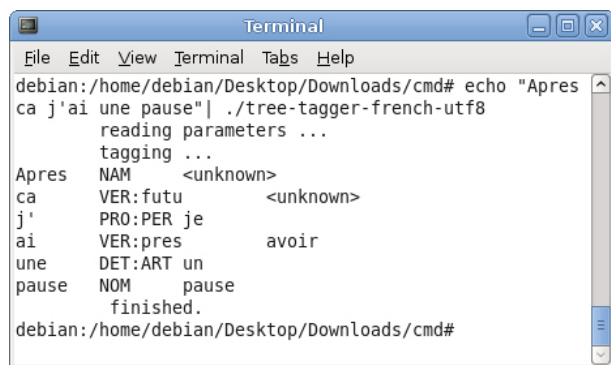
1 TC project: <http://www.ims.uni-stuttgart.de/projekte/tc/>

was developed and implemented at the university of Stuttgart in Germany. It was trained on native English input and tested on data from the Penn Treebank corpus (Marcus et al. 1993). There was, to this researcher's knowledge, no other tests with regard to the French language module. According to Schmid (1994), the tagger achieved a tagging accuracy of 96.34% on English texts written by native speakers. This result does not reflect the percentage obtained with texts written by second language learners.

## 2.2 Improving the tagging accuracy: A three step process

A major challenge in part-of-speech tagging is to identify the unknown and ambiguous words, since unambiguous words can be automatically and accurately processed (Mihalcea 2003). Provided that the word to be labelled had occurred in the corpus that was used to train the tagger, the lemma is identified, otherwise it is tagged as unknown. In the case of an unknown lemma, the tagging tool computes the most likely part-of-speech tag by extrapolating the best option (Daelemans and Van der Bosch 2005). While an exhaustive training corpus is inconceivable, finding ambiguous unknown words in a text not seen before by the tagger is unavoidable. As a result, since tagging error rates correlate with the amount of unknown words (Shrivastava et al. 2005), reducing the amount of not readily identifiable words is a relevant place to start.

**2.2.1 Identifying unknown lemmas.** The first improvement aims to reduce the amount of unknown lemmas that are, by definition, ambiguous for and not recognised by the tagger. It may be assumed that unknown lemmas can be of any open class tags, such as noun, adjective or verb, and that those unknown words may be the results of misspellings, word invention, or simply correct words that did not occur in the corpus used to train the tagger originally. Figure 1 shows a sample output obtained from TreeTagger after running the command shell: "*Apres ca j'ai une pause*" | ./tree-tagger-french-utf8, which contains two incorrect words: *\*Apres* 'After' and *\*ca* 'this'.



```

Terminal
File Edit View Terminal Tabs Help
debian:/home/debian/Desktop/Downloads/cmd# echo "Apres
ca j'ai une pause" | ./tree-tagger-french-utf8
  reading parameters ...
  tagging ...
Apres  NAM    <unknown>
ca     VER:futu   <unknown>
j'    PRO:PER  je
ai    VER:pres   avoir
une   DET:ART   un
pause NOM      pause
      finished.
debian:/home/debian/Desktop/Downloads/cmd#

```

**Figure 1**  
TreeTagger output

The sentence is tokenised and each token is annotated with its part-of-speech and lemma information. Some of the lemmas are unknown to the tagger. For example, the incorrect word *\*Apres* 'After' is tagged as a proper name (NAM) and the unrecognised lemma is marked as <unknown>. One cause identified for not recognising certain lemmas is due to capital letters: TreeTagger does not apply the proper part-of-speech tag for a few tokens, either placed at the beginning or in the middle of a sentence. Table 1,

for instance, shows the part-of-speech and lemma information provided by the tagger after running the input *Cette semaine en Allemagne* ‘This week in Germany’ (#1), and the output obtained after transposing all uppercases into lowercases (#2).

**Table 1**  
Lowercase transposition

Input:	Cette	semaine	en	Allemagne
#1	*NAM <unknown>	NOM <semaine>	PRP <en>	NAM <Allemagne>
#2	DET:DEM <ce>	NOM <semaine>	PRP <en>	*NOM <unknown>

The first token of the sentence *Cette* ‘this’, starting with an appropriate uppercase letter, is labelled as a proper noun (NAM). The lemma information, <unknown>, indicates that the tagger guessed the part-of-speech tag, which is an incorrect guess in this case. On the other hand, the tag applied to the last token *Allemagne* ‘Germany’ –starting with an appropriate uppercase letter– is recognised by the tagger and properly tagged. The last row shows the results obtained with the same tokens transposed into lowercases. While the lemma information and part-of-speech tag provided for the token *cette* are now correct (DET:DEM, demonstrative determiner), the token *Allemagne*, converted into lowercases, is no longer recognised as a proper noun. The tagger estimated its part-of-speech as being most likely a noun (NOM). Converting the original input into lowercases provides accurate part-of-speech and lemma information for most tokens. However, transposing the entire text into lowercases may be at the cost of valuable information, thus lowering the tagger performance. The algorithm used to reduce the amount of unidentified words checks each unknown lemma in the original input and looks for lemma information in the lowercase version. The following record, displayed in Figure 2, lists a sample of tokens whose unknown lemmas were identified with the lowercase conversion method.

token	lemma_1 pos-tagging from treetagger	lemma_2 after uc to lc transposition	pos_1 pos-tagging from treetagger	pos_2 after uc to lc transposition	man_pos_tagged_1 human #1
J'	unknown	je	NAM	PRO:PER	PRO:PER
J'	unknown	je	ADJ	PRO:PER	PRO:PER
Il	unknown	il	NOM	PRO:PER	PRO:PER
Fermer	unknown	fermer	NAM	VER:infi	VER:infi
Mettez	unknown	mettre	NAM	VER:pres	VER:impe
Calme	unknown	calme	NAM	ADJ	ADJ
Une	unknown	un	ABR	DET:ART	DET:ART
Logement	unknown	logement	NAM	NOM	NOM
Clubs	unknown	club	NAM	NOM	NOM

**Figure 2**  
Examples of unknown lemmas resolved with the lowercase transposition

The *token* column enumerates the tokens in their original format. The *lemma\_1* and *pos\_1* columns show the lemma information and the part-of-speech tags obtained after running TreeTagger on the original input. The *lemma\_2* and *pos\_2* columns list the lemma information and the part-of-speech tags retrieved from the text when converted into small letters. The *man\_pos\_tagged\_1* column refers to the hand-annotated tags, which are used as indicators to check the tagger performance with the new procedures.

For example, the first token *J' 'I'*, correctly spelt and not recognised by TreeTagger, was tagged as a proper noun (NAM), which is incorrect. The lowercase conversion allows TreeTagger to recognise the token. As a result, the lemma information *je 'I'* is provided and the most likely part-of-speech tag is estimated, i.e., personal pronoun (PRO:PER), which corresponds to the hand-annotated pos-tag.

If the lemma information is still unavailable, the token is checked against an additional word bank of non ambiguous abbreviations, adjectives, adverbs, proper nouns, compound nouns or English words commonly used in the students' written documents, see Figure 3 below.

intruction	NOM	<input type="text" value="NOM"/>	unknown <input type="text" value="intruction"/>
vetements	NOM	<input type="text" value="NOM"/>	unknown <input type="text" value="vêtement"/>
comfortable	ADJ	<input type="text" value="ADJ"/>	unknown <input type="text" value="confortable"/>
peut-etre	VER:infi	<input type="text" value="ADV"/>	unknown <input type="text" value="peut-être"/>
parceque	VER:subp	<input type="text" value="KON"/>	unknown <input type="text" value="parce que"/>
francais	VER:impf	<input type="text" value="ADJ"/>	unknown <input type="text" value=""/>
<input type="button" value="submit"/>			

**Figure 3**  
Insertion of new entries into the word bank

The word bank is a list of unambiguous tokens with lemma and part-of-speech information that were not initially recognised by TreeTagger and not solved with the uppercase to lowercase conversion algorithm. If the token with an unknown lemma has an entry in the word bank, the part-of-speech and lemma information is automatically updated. Otherwise, the token may be added into the database under the condition that it has a unique part-of-speech. The token and its part-of-speech, as originally estimated by TreeTagger, are given in columns 1 and 2. The user (the person who feeds TreeTagger with the input) chooses in the drop down list the part-of-speech tag that corresponds to the entry. If the user provides the lemma in column 4, the token is considered as non-ambiguous and stored into the database. Otherwise, the application interprets an empty lemma box as the token being ambiguous, which implies that the token can be labelled with more than one tag. When a token is incorrect and ambiguous, for instance *\*francais* 'French' (adjective or noun), the part-of-speech tag can be disambiguated with the surrounding context of the token. The next step shows how a set of hand-written rules based on the observation of inconsistencies in the original tagging can further improve the tagging accuracy.

**2.2.2 Rule-based part-of-speech tagging.** The rule-based part-of-speech tag review is based on the BRILL tagging method, which assigns first default initial tags, i.e., the most frequent tag for that word, and then applies replacement rules based on the analysis of lexicon, morphological and contextual rules (Brill 1995). The method adopted here is to assign the most probable tag to each token with TreeTagger, and then to apply replacement rules depending on prior and/or subsequent part-of-speech tags, lemma information and tokens themselves. TreeTagger performance is checked against a set of 45 additional rules written by this researcher in order to exclude consistent incorrect

part-of-speech tags. The replacement rules are described under the form of phonological rules including regular expressions. While regular expressions are used to show the exact part of the string that is intended to be a search pattern, phonological rules are generally outlined as

- $A \rightarrow B/X\_Y$ , where  $A$  becomes  $B$  between  $X$  and  $Y$
- $A \rightarrow B/X\_$ , where  $A$  becomes  $B$  after  $X$
- $A \rightarrow B/\_Y$ , where  $A$  becomes  $B$  before  $Y$

The whole set of rules is classified into three categories. The first one includes rules that refine the original tag set of TreeTagger by either adding or subtracting part-of-speech tags. The second category contains rules that look at specific part-of-speech tags and update them depending on prior and/or subsequent token, part-of-speech and lemma information. Finally, the third group relates to rules that capture issues involving specific tokens. Table 2 below lists some of the rules for the three categories.

---

**Table 2**  
Classification of the hand-written rules and examples

---

*Category #1: Adding or subtracting part-of-speech tags*

e.g. Rule that refines the numeral (NUM) part-of-speech tag into adjective (ADJ), numeral adjective (ADJ:NUM):

```
^(NUM)$ → ^(ADJ)$/_  
if token=[^(.*\d.*)] AND if token!=(i(è|e|é)re*s*)$  
ELSE ^(NUM)$ → ^(ADJ : NUM)$/_  
if token=[^(.*\d.*)] AND if token!=^(c|m|x|i|l)$
```

---

*Category #2: Updating existing tags depending on context*

e.g. Rule that replaces adjective (ADJ) tags by noun (NOM) tags when situated between a determiner (DET) and a verb (VER):

```
^(ADJ)$ → ^(NOM)$/^(DET)_^(VER)
```

---

*Category #3: Capturing issues involving specific tokens.*

e.g. Rule that changes the part-of-speech of the token *pour* 'for' always misidentified as a conjunction (KON) into a noun (NOM) or a preposition (PRP) depending on context:

```
token → ^(NOM)$/^(DET)_  
if token=^(pour)$ AND if pos!=^(PRP)$  
ELSE token → ^(PRP)$/_  
if token=^(pour)$ AND if pos!=^(PRP)$
```

The first category of the above table illustrates how the original tag set is refined by adding part-of-speech tags. For example, TreeTagger classifies, without any distinction, cardinal or ordinal adjectives written either in figures or in letters as numeral (NUM). Whereas a cardinal numeral adjective indicates that the adjective is a definite number, an ordinal adjective refers to an ordering relation of elements. The following example shows that the cardinal number *trois* ‘three’, the ordinal number *première* ‘first’ and the cardinal number 500 written in figures are all tagged as numeral, which is correct but not specific enough, especially with a view to analysing possible agreement errors in the French language:

*J’ai écrit trois(NUM) fois la première(NUM) partie de 500(NUM) mots* - ‘I wrote three times the first part of 500 words’

Distinguishing between these instances of numeral tags is relevant since (a) an ordinal adjective can be compared to an adjective that qualifies a noun in terms of number and gender agreement, and (b) a cardinal adjective written in letters is most of the time invariable but susceptible to be incorrectly written. If the token is tagged as a numeral (NUM) and is not a digit and ends with a combination of *i+e+è+re+s* (e.g., *prem-ier*, *prem-ière*, *prem-iers*, ..., ‘first’), the part-of-speech tag is replaced by the adjective tag (ADJ). Otherwise, the token –if not a digit nor a Roman numeral– is tagged as a numeral adjective (ADJ:NUM). Instead of exclusively using the pos-tag NUM, the following example is now tagged as:

*J’ai écrit trois(ADJ:NUM) fois la première(ADJ) partie de 500(NUM) mots* - ‘I wrote three times the first part of 500 words’

The second category, as shown in Table 2 above, identifies inconsistencies that can be solved with general descriptions. For example, the rule  $\wedge(ADJ)\$ \rightarrow \wedge(NOM)\$/\wedge(DET)\_ \wedge(VER)$  allows the replacement of all adjective tags by the noun tag when located between a determiner and a verb. Table 3 shows an extract of a student’s incorrect sentence *Notre sujet est \*démontrer...* ‘Our subject is to show...’ where the aforementioned rule can be applied.

**Table 3**  
TreeTager output and post-editing

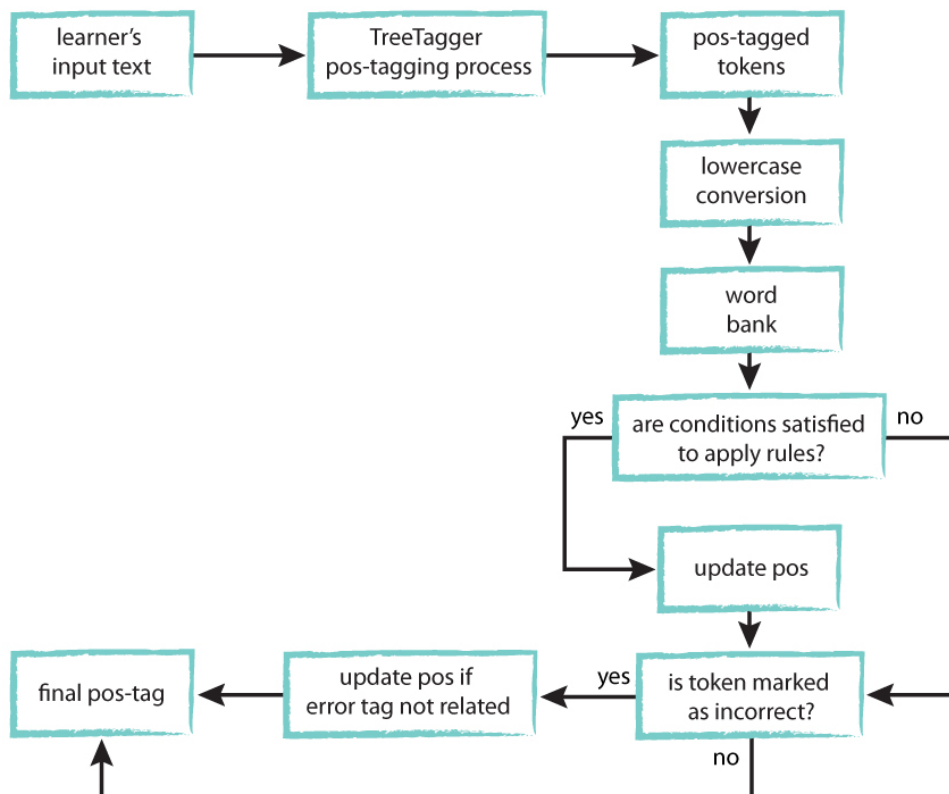
Tokens	Notre	sujet	est
TreeTager output	*NAM	*ADJ	VER:pres
Lowercase conversion	DET:pos	-	-
$\wedge(ADJ)\$ \rightarrow \wedge(NOM)\$/\wedge(DET)\_ \wedge(VER)$	-	NOM	-
Final output	DET:pos	NOM	VER:pres

The original output shows two incorrect tags. With the lowercase conversion coming first, the incorrect tag \*NAM applied to the determiner *Notre* ‘our’ is replaced by the possessive determiner (DET:POS) tag. The token *sujet* ‘subject’, incorrectly tagged as an adjective (ADJ), is with the application of the rule replaced by the noun (NOM) tag, as it is now preceded by a determiner and followed by a verb. The final output illustrates the usefulness of the lowercase conversion combined with the rule-based tagging review. After checking each token against the set of rules based on recurrent tagging errors, the



- $[\wedge(VER : (pper|conj))] \rightarrow \wedge(VER : pper)\$/\wedge(VER : conj)\$\wedge(ADV)?\_$
- $ELSE [\wedge(VER : (pper|conj))] \rightarrow \wedge(VER : conj)\$/[\wedge(VER : conj)]\_$
- $ELSE [\wedge(VER : (pper|conj))] \rightarrow \wedge(VER : pper)\$/\wedge(VER : inf)\$_$

The first rule checks whether the current tag is preceded by a conjugated verb (VER:conj) followed by zero or one adverb (ADV). If it is, the tag should be a past participle (VER:pper). Otherwise, if the current tag does not follow any conjugated verb, then the tag itself should be a conjugated verb (VER:conj). Otherwise, if the tag follows an infinitive verb, then it is most likely that the token is a past participle (VER:pper). The whole process to improve the part-of-speech tagging accuracy can be summarised as follows (Figure 5 below).



**Figure 5**  
Process to improve the tagging accuracy

The learner's raw input is first fed into TreeTagger, then information on unknown lemmas is searched within the lowercase version and the consultation of the word bank. The next step implies checking each token against a set of rules intended to reduce consistent tagging errors. If the rules can be applied, the current tag is replaced, otherwise the tag is left unchanged. Finally, each token, if marked as incorrect in the

error-annotated corpus, is (a) compared to the error tag and (b) updated if they are not related.

### 3. Inter-rater agreement analyses

To evaluate the tagging process accuracy of TreeTagger, the output produced before and after improvements is compared to a hand-annotated tagged corpus. The baseline strategy simply consisted in hand-tagging the corpus token by token using the exact same part-of-speech tag set that is given with TreeTagger. The *Petit Larousse* and *Petit Robert* dictionaries<sup>2</sup> were used as references when hesitating between different tags. A set of 10,108 tokens corresponding to 14 students' written texts, in which 8,424 words and 1,440 lexical and grammatical errors were encountered, was manually tagged by this researcher. The mean error rate per hundred words is equal to 17.11%. To compare TreeTagger with the gold standard data, the percentage of agreement excluding chance between human and machine is estimated through the means of Krippendorff's (1970) alpha and Cohen's (1960) kappa coefficients, –two measures using distinct ways of computing the expected agreement (Di Eugenio and Glass 2004). In addition, the tagging accuracy is evaluated with recall, precision and F-measures, three widely used metrics in part-of-speech tagging to estimate the effectiveness of a system (Voutilainen 2003). To ensure a high quality in tagging, this researcher's reliability was also checked against the tagging of another human; a former teacher in France, who is also a passionate of French grammar. The corpus used on this occasion is, however, much more modest than the one mentioned above; it includes 2,022 tokens in total.

#### 3.1 Kappa and Alpha coefficients

Krippendorff's (1970) alpha coefficient measures the agreement between any numbers of raters who are processing the same set of items. A strong argument in favour of this means of measurement, is that the alpha algorithm is able to calculate the reliability coefficient regardless of missing data, sample size, number of categories or coders. To make the alpha coefficient practical and easily usable, Hayes and Krippendorff (2007) describe and illustrate how to use their macro developed for SPSS<sup>3</sup>, also called KALPHA<sup>4</sup> macro. The agreement measure between two raters is generally presented under the following formulation:

$$\alpha = 1 - \frac{D_o}{D_e} \quad (1)$$

where  $D_o$  is the observed agreement and  $D_e$  is the agreement expected when the coding is attributable to chance. For example, to compute nominal data with two raters, i.e., human and machine, and no missing data, one must first construct a *reliability data*

---

<sup>2</sup> Le Petit Larousse Illustré 2007 and Le Petit Robert 2009 are both French monolingual dictionaries.

<sup>3</sup> SPSS is a registered brand name IBM SPSS Statistics (formerly SPSS Statistics): <http://www.spss.com/statistics/>

<sup>4</sup> The KALPHA macro has been written by Andrew F. Hayes to promote the use of the Krippendorff's (1970) alpha coefficient since most statistical software packages do not include this reliability coefficient. For an SPSS or SAS version of the script, and further explanations about how to execute the macro command, visit Andrew F. Hayes' website <http://www.comm.ohio-state.edu/ahayes/>.

matrix as illustrated in Figure 6. This example is a 2-by-8 matrix containing 16 values, where the first token is tagged as a noun (NOM) by both raters.

	1	2	3	4	5	6	7	8
human	NOM	NOM	ADV	ADV	NOM	ADV	NOM	ADJ
machine	NOM	NOM	ADJ	ADV	ADJ	ADJ	NOM	ADJ

**Figure 6**  
Reliability data matrix

Then, the coincidence matrix, shown in Figure 7 below, tabulates the values that are represented in the above reliability data matrix. For example, the number 6 accounts for the 3 NOM-NOM pairs  $O_{NOM-NOM}$  representing the tokens #1, #2 and #7 in Figure 6, and the numbers 1 account for the pairs  $O_{NOM-ADJ}$  and  $O_{ADJ-NOM}$  representing token #5. The number of NOMs is represented as  $n_{NOM}$ , the number of ADV as  $n_{ADV}$ , and so on.

	NOM	ADV	ADJ	
NOM	6	.	1	7
ADV	.	2	2	4
ADJ	1	2	2	5
	7	4	5	16

**Figure 7**  
Alpha coincidence matrix

The total number is  $n$  ( $n=16$ ). The calculation of the alpha coefficient is as follows:

$$\alpha = \frac{(n-1) \sum_c o_{cc} - \sum_c n_c(n_c-1)}{n(n-1) - \sum_c n_c(n_c-1)} \quad (2)$$

$$\alpha = \frac{(16-1)(6+2+2) - \langle 7(7-1) + 4(4-1) + 5(5-1) \rangle}{16(16-1) - \langle 7(7-1) + 4(4-1) + 5(5-1) \rangle} = .4578 \quad (3)$$

The percentage of agreement excluding chance would be 45.78% for this example. The calculation of the kappa coefficient differs from the alpha coefficient in the sense that the coefficient represents the expected agreement based on the preferences of the raters rather than on the coding process. The Cohen's (1960) kappa coefficient is as follows:

$$\kappa = \frac{P_o - P_c}{1 - P_c} \quad (4)$$

where  $P_o$  is “the proportion of units in which the judges agreed”, and  $P_c$  is “the proportion of units for which agreement is expected by chance” (Cohen 1960, p. 39). Taking the same example used to illustrate the alpha coefficient (Figure 6), the crosstabulation table is as follows (Figure 8). For example, the numbers 1 account for the pairs  $O_{NOM-ADV}$ ,  $O_{ADV-ADV}$  and  $O_{ADV-ADJ}$ . The total number of pairs is  $n$  ( $n=8$ ). The proportion of agreement expected by chance  $P_c$  for the category NOM is computed by multiplying the total sum of the row  $n_{NOM}$  with the total sum of the column  $n_{NOM}$ . The product is then divided by the total amount of pairs.

		machine			
		NOM	ADV	ADJ	
human	NOM	3	.	1	4
	ADV	.	1	2	3
	ADJ	.	.	1	1
		3	1	4	8

**Figure 8**  
Kappa crosstabulation table

With the values of observed count of agreement included in the crosstabulation table, the kappa coefficient (in this example) is calculated as follows:

$$\kappa = \frac{(3 + 1 + 1) - ((3 * 4)/8) + ((1 * 3)/8) + ((4 * 1)/8)}{8 - ((3 * 4)/8) + ((1 * 3)/8) + ((4 * 1)/8)} = .4666 \quad (5)$$

As for all chance corrected agreements measures, a kappa coefficient of 1.00 means perfect agreement, whereas a coefficient of 0.00 reflects no agreement but chance. Cohen’s kappa coefficient may be computed with SPSS that requires a symmetric 2-way table in which the part-of-speech tags of the first rater must match the part-of-speech tags of the second rater. Yet, the part-of-speech tags used by both raters, i.e., human and machine, to tag the corpus was unequal in this study. For example, while TreeTagger selected the VER:subi tag 5 times, this tag was never selected in the manual encoded corpus. The reverse order is also true, the interjection tag was used in the hand-annotated corpus and not once in the automatic encoded corpus. SPSS computes the kappa coefficient if and only if the two raters have exactly the same categories. The solution to this issue is to add extra entries with the missing categories for each rater, and arbitrarily declare that both raters agree on the part-of-speech. The effect of adding the missing categories may be minimized by setting a negligible weight (weight=.0001) to the additional entries and a large weight (weight=1) to other genuine entries, as seen in Figure 9.

	token	machine	human	weight
10104	manière	NOM	NOM	1.00
10105	je	PRO:PER	PRO:PER	1.00
10106	me	PRO:PER	PRO:PER	1.00
10107	repose	VER:conj	VER:conj	1.00
10108	!	SENT	SENT	1.00
10109		VER:subi	VER:subi	.00
10110		VER:impe	VER:impe	.00

**Figure 9**  
Weighing data to compute kappa

### 3.2 Recall, precision and F-measures

Whereas the recall measure computes the percentage of agreement by dividing the number of correct part-of-speech tags in TreeTagger by the number of correct part-of-speech tags in the Gold data, the precision measure calculates the percentage of agreement by dividing the number of correct part-of-speech tags in TreeTagger by the total number of part-of-speech tags in TreeTagger.

$$recall = \frac{\text{number of correct pos tags in TreeTagger}}{\text{number of correct pos tags in Gold Data}} \quad (6)$$

$$precision = \frac{\text{number of correct pos tags in TreeTagger}}{\text{number of total pos tags in TreeTagger}} \quad (7)$$

High scores in recall and precision may be considered as valuable only if the two measures are of approximate same high values. If the span between both results is too wide, for example 55% in recall and 98% in precision, both figures are combined together to normalize the compromise between recall and precision, which is called F-measure (Hull and Gomez 2002).

$$F - \text{measure} = \frac{(\beta + 1.0)(precision)(recall)}{\beta(precision) + (recall)} \quad (8)$$

Whereas  $\beta = 1$  equally considers recall and precision, a  $\beta$  value of 0.5 put the accent on precision and a  $\beta$  value of 2 emphasises the recall perspectives.

## 4. Results

### 4.1 Human versus human

The inter-human rater reliability was measured with kappa coefficient. The percentage of agreement excluding chance between both raters is equal to 95.7% (Table 4). This result, considered as slightly low by this researcher, was further investigated, and the main cause of disagreement was easily explainable.

**Table 4**  
Inter-rater reliability: Human versus human

	Value	Asymp. Std. Error	Approx. T	Approx. Sig.
Kappa	.957	.005	146.202	.000
N of Valid Cases	2004			

The tag set included a part-of-speech tag named *POS*, which was used when raters could not apply a specific tag to a particular token. For example, the token *\*Que-ce-que* 'what does' not properly tokenised, was not manually assigned with a part-of-speech tag as it is rather challenging to identify the right one. Rater #2 applied the unspecified tag (*POS*) much more frequently than rater #1, in particular when the tokens were English words. Figure 10 shows a few examples of those tokens tagged with the *POS* tag. Given the fact that most discrepancies between both raters occurred because of the frequent use of the *POS* tag by rater #2, the results obtained from the inter-human reliability analysis were considered high and extremely satisfying.

token_id	token	man_pos_tagged_1 human #1	man_pos_tagged_2 human #2
1175	hui	ADV	POS
1448	Que-ce-que	POS	POS
9429	un	DET:ART	POS
9444	En	PRP	POS
9625	Home	NOM	POS
9626	and	KON	POS
9627	Away	ADV	POS

**Figure 10**  
Unspecified part-of-speech tag

#### 4.2 Machine versus human

The results obtained by running the KALPHA macro on the data set is illustrated below. Figures 11 and 12 show that the Krippendorff's alpha coefficient rose from 89.62% ( $\alpha = .8962$ ) to 97.60% ( $\alpha = .9760$ ). Krippendorff (2004) suggests that we can "rely on variables with reliabilities above  $\alpha=.800$ " and "consider variables with reliabilities between  $\alpha=.667$  and  $\alpha=.800$  only for drawing tentative conclusions" (p. 241).

```
Run MATRIX procedure:

Krippendorff's Alpha Reliability Estimate
```

	Alpha	LL95%CI	UL95%CI	Units	Observers	Pairs
Nominal	.8962	.8898	.9025	10108.0000	2.0000	10108.0000

**Figure 11**  
Krippendorff's alpha coefficient before improvements

Run MATRIX procedure:						
Krippendorff's Alpha Reliability Estimate						
	Alpha	LL95%CI	UL95%CI	Units	Observers	Pairs
Nominal	.9760	.9729	.9792	10108.0000	2.0000	10108.0000

**Figure 12**  
Krippendorff's alpha coefficient after improvements

The proportions of agreements with kappa, before and after improvements between both human and machine after chance has been excluded, rose from 89.6% to 97.6% of agreement, Tables 5 and 6, respectively.

**Table 5**  
Cohen's kappa coefficient before improvements

	Value	Asymp. Std. Error	Approx. T	Approx. Sig.
Kappa	.896	.003	306.858	.000
N of Valid Cases	10,108			

**Table 6**  
Cohen's kappa coefficient after improvements

	Value	Asymp. Std. Error	Approx. T	Approx. Sig.
Kappa	.976	.002	326.262	.000
N of Valid Cases	10,108			

Finally, the last set of measurement (recall, precision and F-measure) also demonstrate an improvement. The figures go from 77.67% up to 97.21%. Table 7 below summarises all results obtained from the agreement analyses.

**Table 7**  
Summarising the results of the agreement analyses

	Alpha	Kappa	Precision	Recall	F-measure
Before	89.62%	89.60%	78.43%	77.67%	78.03%
After	97.60%	97.60%	97.21%	96.01%	96.61%

## 5. Discussion and limitations

Craggs and Wood (2005, p. 293) point out that "it is impossible to prescribe a scale against which all coding schemes can be judged", which suggests that the threshold at which the level of agreement will be considered as sufficient is to be determined in a subjective way. The results obtained in this article are certainly encouraging. With the alpha, kappa, and precision results placed above the upper bound of 97% –upper bound at which taggers tend to not surpass (Vanroose 2001)–, the part-of-speech tagging was deemed a reliable process to further draw conclusions from the annotated corpus, even

if an amelioration is always feasible. The primary aim of improving the part-of-speech tagging accuracy was to ensure that the learner interlanguage corpus was annotated with robust part-of-speech tags so as to draw solid conclusions from the various analyses. The secondary objective was to propose a means to improve the tagger performance when processing input from language learners. The method proposed is not without drawbacks and has to be considered with caution. The main limitations refer to the use of (a) the word bank, which considerably slows down the system and creates time-out errors, and (b) the additional hand-written rules that cannot be generalised to other languages since the rules were specifically written for the French language. In addition, the method was tested on the training data, as opposed to be tested on an unseen corpus, which could explain the excellent results obtained. A further evaluation would be to test the various rules written to improve the tagger with other data than the one used to identify issues in tagging. The strength of this method however is the cross-reference approach, which compares part-of-speech tags with error type. This orientation seems to be a direction to be further explored, as error correction either manual or automatic could provide information on lexical and syntactical formations, all certainly useful for the tagging process.

## 6. Conclusion

This article presented an approach proposed to improve the tagging accuracy when processing language learners' written documents. The approach is based on three sequential steps: Firstly, the identification of unknown lemmas through the means of a lowercase conversion and a word bank; secondly, the reduction of incorrect tags consistently applied with a set of additional rules; and finally, the elimination of a few incorrect tags due to learners' ill-formed words by cross-referencing the part-of-speech tags with the learners' error-annotated corpus. The performance of the tagger before and after improvement was analysed through the means of simple percentage measures and percentages of agreement that take into account the chance parameter. The output produced by the tagger was compared with the hand-annotated part-of-speech tags applied to the same corpus. Bearing in mind that the input text was real-life data from language learners of French, the results obtained showed a definite improvement.

## Acknowledgments

I would like to express my gratitude to all the researchers present at the Automatic Analysis of Learner Language (AALL'09) workshop, where this article was presented and discussed.

## References

- Black, E. and Jelinek, F. and Lafferty, J. and Mercer, R. and Roukos, S. 1992. Decision tree models applied to the labeling of text with parts-of-speech. In *Decision Tree Models Applied to the Labeling of Text with Parts-of-Speech*, pp. 117–121, Morristown, NJ. Association for Computational Linguistics.
- Brill, E. 1995. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Computational Linguistics*, 21(4), pp. 543–565.
- Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), pp. 37–46.
- Craggs, R. and Wood, M. 2005. Evaluating discourse and dialogue coding schemes. *Computational Linguistics*, 31(3), pp. 289–296.
- Cutting, D. and Kupiec, J. and Pedersen, J. and Sibun, P. 1992. A practical part-of-speech tagger. In *A practical part-of-speech tagger*, Morristown, NJ. Association for Computational Linguistics.
- Daelemans, W. and Van der Bosch, A. 2005. *Memory-based language processing*. Cambridge University Press, Cambridge.

- Di Eugenio, B. and Glass, M. 2004. The kappa statistic: A second look. *Computational Linguistics*, 30(1), pp. 95–101.
- Díaz-Negrillo, A. and Meurers, D. and Valera, S. and Wunsch, H. 2010. Towards interlanguage pos annotation for effective learner corpora in sla and flt. *Language Forum*, 36(1/2), pp. 1–15.
- Faaß, G. and Heid, U. and Taljard, E. and Prinsloo, D. 2009. Part-of-speech tagging of northern sotho: Disambiguating polysemous function words. In G. De Pauw, G.-M. De Schryver, and L. Levin, editors, *Inproceedings of the EACL 2009 Workshop on Language Technologies for African Languages, AfLaT 2009, 31 March 2009 Megaron Athens International Conference Centre Athens, Greece*. The Association for Computational Linguistics, Stroudsburg, PA, pp. 38–45.
- Granger, S. 2002. A bird's-eye view of learner corpus research. In S. Granger, J. Hung, and S. Petch-Tyson, editors, *Computer learner corpora, second language acquisition and foreign language teaching*. John Benjamins Publishing Company, Amsterdam, pp. 3–33.
- Granger, S. 2003. Error-tagged learner corpora and call: A promising synergy. *CALICO Journal*, 20(3), pp. 465–480.
- Greene, B.B. and Rubin, G.M. 1971. Automated grammatical tagging of english. *Department of Linguistics, Brown University*.
- Hayes, A.F. and Krippendorff, K. 2007. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1(1), pp. 77–89.
- Hull, R. and Gomez, F. 2002. Automatic acquisition of biographic knowledge from encyclopedic texts. In A. Kent and J.G. Williams, editors, *Encyclopedia of microcomputers: volume 28*. Marcel Dekker, INC, New York, pp. 1–16.
- Jurafsky, D. and Martin, J.H. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, New Jersey.
- Karlsson, F. 1995. Designing a parser for unrestricted text. In F. Karlsson, A. Voutilainen, J. Heikkilä, and A. Anttila, editors, *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin, pp. 1–40.
- Krippendorff, K. 1970. Bivariate agreement coefficients for reliability of data. *Sociological Methodology*, 2(1), pp. 139–150.
- Márquez, L. and Padró, L. and Rodríguez, H. 2000. A machine learning approach to pos tagging. *Machine Learning*, 39(1), pp. 59–91.
- Marshall, I. 1983. Choice of grammatical word-class without global syntactic analysis: Tagging words in the lob corpus. *Computers and the Humanities*, 17(3), pp. 139–150.
- Mihalcea, R. 2003. Unsupervised natural language disambiguation using non-ambiguous words. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent advances in natural language processing III: selected papers from the RANLP conference 2003*. John Benjamins Publishing Company, Philadelphia.
- Pradhan, S.S. and Ward, W. and Martin, J.H. 2008. Towards robust semantic role labeling. *Computational Linguistics*, 34(2), pp. 556–563.
- Schmid, H. 1994. Probabilistic part-of-speech tagging using decision trees. In *Probabilistic part-of-speech tagging using decision trees*, Manchester, UK.
- Shrivastava, M. and Agrawal, N. and Singh, S. and Bhattacharya, P. 2005. Harnessing morphological analysis in pos tagging task. In S.M. Bendre, A. Mukherjee, and R. Sangal, editors, *Inproceedings of the international conference on natural language processing (ICON-2005)*. Allied Publishers PVT. Limited, New Delhi.
- Thouësny, S. Forthcoming. *Modeling second language learners' interlanguage and its variability: A computer-based dynamic assessment approach to distinguishing between errors and mistakes*. Ph.D. thesis, Dublin City University, Dublin.
- Thouësny, S. and Blin, F. 2010. Modeling language learners' knowledge: What information can be inferred from learners' free written texts? In M. Levy, F. Blin, C. Bradin Siskin, and O. Takeuchi, editors, *WorldCALL: International Perspectives on Computer-Assisted Language Learning*. Routledge, New York.
- Tlili-Guiassa, Y. 2006. Hybrid method for tagging arabic text. *Journal of Computer Science*, 2(3), pp. 245–248.
- Vanroose, P. 2001. Part-of-speech tagging from an information-theoretic point of view. In *Part-of-speech tagging from an information-theoretic point of view*, pp. 9–14.
- Voutilainen, A. 2003. Part-of-speech tagging. In R. Mitkov, editor, *The Oxford handbook of computational linguistics*. University Press, Oxford, pp. 219–232.