POS tagging of read speech
A tools comparison and an adaptation proposal

Chiara Pesenti[1,2]
[1]Department of Humanities, University of Turin
[2]CLST (Center for Language and Speech Technology), Radboud University

Abstract

This study aims to investigate the effectiveness of the POS tagging process with regard to read speech, focusing on two main aspects. On one hand, it wants to compare the functioning and the output's differences of two different parsing tools for Dutch language. On the other hand, it focuses on how the divergences between standard written language and read speech could possibly affect a correct POS tagging of read speech. The POS tags of read speech and standard written language obtained from the two parsers are observed and compared, and adaptation regarding the punctuation relevance on read speech is proposed and discussed. More specifically, the POS tagging process improved with the automatic insertion of punctuation marks in the orthographic transcriptions of read speech. The present research showed both expected and unexpected results: the POS tagging of read speech reported less accuracy than the POS tagging of standard written language, and the two parsing tools showed different outputs not only within the read speech but also within the standard written language. Moreover, a parsing tool proved to perform more accurately than the other regarding the correctness of the POS tags, and at the same time seemed more sensitive to the insertion of punctuation in read speech. Therefore, the results showed that the specific mechanism underlying the two Dutch parsing tools sometimes can lead to misleading results. The findings also showed how written language is more easily interpreted by parsing tools than read speech, and confirmed that punctuation marks play a rather important role in the POS tagging process.

1 Introduction

1.1 Background

Part-Of-Speech tagging (POS tagging) is the process of assigning a special label to each token in a text corpus, by specifying the part of speech of the token according to the word's definition and syntactic context [1, 2, 3]. As pointed out in [4], a parsing tool might perform differently according to the specific type of language analysed (i.e., written or spoken language). Read speech shares some cha-

racteristics with both written language and spoken language. The syntactic structure of a read speech is rather consistent and not fragmented, similarly to written language. Yet, some typical elements of spoken language are also present in read speech, such as no punctuation, stuttering, repetitions, bribes, filled pauses, and lengthened vowels.

In this study, two parallel comparisons are made: the comparison between the POS tagging outputs of two different parsing tools for Dutch language, and the comparison between the POS tags obtained from written language and the POS tags obtained from orthographic transcriptions of read speech. More specifically, a text containing the sentence prompts (SPs) is first parsed by the tools, as standard written language, and then a text containing the orthographic transcriptions (OTs) from recordings of speakers reading the SPs is parsed as read speech, with the same parsing tools.

Afterwards, to obtain a better performance of the POS tagging tools, a computational adaptation involving the automatic addition of punctuation marks is applied on the OTs, and its results are discussed.

1.2 Research questions and hypothesis

The research questions (RQs) of this study are the following:

* do the two parsers for Dutch language perform POS tagging similarly?

* is the POS tagging of SPs more accurate than the POS tagging of OTs?

* how and to what extent the presence of punctuation affects the POS tagging process?

According to the initial assumptions, the two parsing tools are expected to return a similar output. Moreover, since the parsing tools used in this study are designed with a reference point more akin to written language rather than read speech [4, 2, 3], like most parsing tools, greater accuracy is assumed to be performed for the POS tags of the SPs rather than for the POS tags of OTs. Also, it is expected that the POS tagging of the OTs would return a more accurate output after the insertion of punctuation marks in the OTs. A similar behavior is expected from the two parsing tools.

2 Materials and method

2.1 Materials and method: parsing tools, read speech, scripts

2.1.1 Parsing tools

The software used in this research are named Frog [5] and Alpino [6], and they are both avail- able at CLST, Center for Language and Speech Technology of Radboud University (https://webservices.cls.ru.nl/). Alpino is a hybrid dependency parser for Dutch, which uses rule-based constraints combined with corpus-based

statistics, while Frog is an NLP suite based on memory-based learning and trained on large quantities of manually annotated data. A basic knowledge of the SSH (Security SHell) and the access to LaMachine, a unified Natural Language Processing (NLP) open-source software distribution, were required to proceed with the parsing. Once the POS tags were extracted with Alpino and Frog, they were moved to an Excel file. For Alpino, this last step required the use of the Python script FoLiA [7] [8] written by Van Gompel and Bloem [9].

2.1.2 Read speech

The read speech used in this study comes from a larger research project which involves atypical speech analysis [10]. A group of 8 Dutch native speakers suffering from dysarthria was asked to read a set of sentences before and after a therapy treatment. From the recordings, orthographic transcriptions (OTs) were later annotated manually. The OTs, together with the text file of the SPs, were POS tagged with Frog and Alpino. Note that the OTs do not report phenomena like filled pauses and lengthened vowels and do not have any punctuation marks, but contain repeated, stuttered, or fragmented words. In this work, only the OTs of the pre-processed recordings are taken into account, and the adaptation proposed in the second stage of this study case concerns only the punctuation, while all the other typical elements of the read speech reported by the OTs (repeated words, stuttering, fragmented words) remain unchanged.

The sentence prompts are seven, and are taken from the story «Papa en Marloes» and from apple pie recipes also used by Ganzeboom et al. [11] [12]. The prompts include 32 sentences for a total of 250 words. Therefore, besides the prompts text, all the words annotated in the OTs, uttered by 8 speakers who were asked to read the prompts, were POS tagged separately for each speaker.

2.1.3 Scripts

Two Python scripts were created specifically for this work. One script was used to compare the different POS tags, grouped by speakers, tools, and kind of text (SPs and OTs), and one script was used to automatically insert the punctuation in the OTs.

The first code was made with the purpose to return the number of mismatches between two given groups of POS tags. More specifically, what the code did was to observe the words and POS tags columns of two different Excel files, e.g., a file reporting the SPs POS tags obtained with Alpino and a file reporting the SPs POS tags obtained with Frog. The code was meant to check row by row if, for instance, the POS tag «*ALPINO-tag1*» corre-

sponding to the word» «*SPsWord1*» in the first Excel file equaled or differed from the POS tag «*FROG-tag1*» corresponding to the word «*SPsWord1*» of the second Excel file. Similarly, for the comparison between an Excel file showing POS tags obtained from SPs and a file showing POS tags obtained from OTs (with the POS tags being extracted by the same parsing tool), the code would check row by row whether the POS tag «*ALPINOtag1*», corresponding to the word «*SPsWord1*» in the first Excel file equaled or differed from the POS tag «*ALPINOtag1*» corresponding to the word «OTsWord1» of the second Excel file. To do this, the script also considered the order in which the words were listed, thus avoiding an overlap between identical words. Henceforth, those POS tags that according to the script differed between each other (e.g. «*ALPINOtag1*» from the first file differs from «*FROGtag1*» or «*ALPINOtag1*» from the second file) will be referred as *mismatches*.

The second script was made to automatically insert the punctuation in the OTs. The code considered the text of the SPs, which contained punctuation, and the text of the OTs, with no punctuation. The code first located and identified the punctuation marks showed in the SPs and then reinserted them into the OTs. It did so by establishing the insertion spot on the basis of the first three words preceding and the first three words following the punctuation marks. Considering the sequence of the three words preceding and following the full stops and the commas was crucial for a proper functioning of the coding. Otherwise, by looking only at the first word preceding and following the punctuation, there would have been the concrete risk to automatically add extra punctuation whenever a word was repeated more than once in the sentences. The same script was used for full stops and commas.

2.2 Pipeline

According to the adopted pipeline, the first step was to extract all POS tags at SPs and OTs level through Frog and Alpino.

Once the POS tags were collected for each word in Excels files, the comparison code was used for two main comparisons: a comparison between SPs POS tags obtained with Alpino and Frog, to observe the functioning of the two tools, and a comparison between the SPs and the OTs POS tags before and after the insertion of the punctuation, to evaluate the punctuation impact. The accuracy of the SPs POS tags was further checked by Dutch native speakers studying at Radboud University.

Afterwards, the punctuation was added to the OTs with the second script,

and the coding effectiveness was evaluated by counting the inserted punctuation. The comparison code was used again for the SPs POS tags and the OTs obtained both with Alpino and Frog, before and after the insertion of the punctuation.

Finally, all the comparison results were collected, and the behavior of the parsing tools was observed together with the impact of the punctuation in the OTs.

3 Observations and results

3.1 Alpino vs. Frog

The SPs POS tags were used as control sample to evaluate the functioning of the two parsing tools. The comparison code revealed that amongst the 250 words included in the prompts, 27 were tagged differently by Alpino and Frog, hence 27 mismatches were found.

Table 1: Prompts POS tags mismatches. Comparison between parsing software

Detected mismatches	Alpino correct mismatches	Frog correct mismatches
27 (out of 250)	23	4

To understand the possible causes of the mismatches and which tool performed better, Dutch grammar skills proved necessary, therefore some native Dutch students from Radboud University were consulted. As shown in Table 1, with their help it turned out that amongst the 27 mismatches revealed, 24 of them were due to a Frog misinterpretation, while only 3 of them were attributed to Alpino.

Table 2: Comparison between prompts and one (random) speaker

Parsing tool	**Punctuation in OTs**	**POS tag mismatches**
Frog	no punctuation	20 out of 250
Frog	punctuation	17 out of 250
Alpino	no punctuation	24 out of 250
Alpino	punctuation	7 out of 250

Thus, Alpino seemed to report a higher number of correct POS tags with respect to Frog. A further confirm of these results was given by the accuracy scores showed in the outputs of Frog. The scoring indeed was always low in correspondence of the mismatches.

3.2 SPs vs. OTs POS tagging, before and after the insertion of punctuation

Also the comparison between SPs and OTs POS tags revealed interesting results. Table 2 shows the number of mismatches revealed from 4 different comparisons: first the number of mismatches revealed with the comparison between the SPs POS tags and the OTs POS tags of a random speaker before and after the insertion of punctuation marks. This first comparison was made twice, once with Alpino and once with Frog. Table 2 reports the number of POS tag mismatches of only one random sampled speaker, since it can be considered significant and

indicative for the entire set of speakers.

The number of mismatches confirmed again a different behavior of the two parsing tools. Before the automatic insertion of the punctuation marks, indeed, Alpino reported 4 mismatches more than Frog. However, once the punctuation was added in the OTs, the results showed a decrease of the number of mismatches for both Alpino and Frog, but the change was much more evident for Alpino. Indeed, if with Frog only 3 mismatched POS tags changed to correct POS tags after the insertion of punctuation, thus decreasing from 20 to 17 mismatches, with Alpino 17 mismatches turned correct, thus decreasing from 24 to 7 after the addition of punctuation.

4 Discussion

The findings of this study case partly met the given assumptions and partly diverged from them.

The most unexpected result is undoubtedly such a noticeable difference between the two software's behaviour. While it was assumed that the results of Alpino and Frog would have been roughly similar, the findings proved otherwise, with regard to the comparison between the POS tagging of standard written language (SPs) and of read speech (OTs), as well as the sensitivity to punctuation marks in read speech. Indeed, it seemed that Alpino performed a more accurate POS tagging than Frog on written language. Considering only the read speech instead, Alpino and Frog's POS tags did not show great difference, but they both reported a 9% of mismatches. After the insertion of punctuation, the difference between the two tools was again accentuated: although both parsers proved to be sensitive to punctuation, Alpino's POS tagging showed a stronger and more significant improvement than the Frog's minimal one. This clear difference between the two tools probably lies in the different operating mechanisms underlying the parsing functioning, which unfortunately are not investigated in detail in this work.

Nevertheless, a closer observation of the mismatches can lead to some interesting considerations. Out of the 27 mismatches revealed amongst SPs, 12 tags concern cases where a verb, correctly interpreted by Alpino, was recognised as a noun or an adjective by Frog. Since this seemed to be a recurrent error in Frog, its origins can be traced back to the syntactic structure of the sentences. Indeed, the sentence prompts from Apple Pie Recipe, also used in [11], are part of culinary recipes, therefore very often start with a verb in the infinitive form in the leftmost position of the sentence. Thus, considering the order of the syntactic elements of a sentence in Dutch language (SVO/ SOV)[12], the elements

in those sentences appear in a non-standard order, since the subject is omitted. That could be a cause of the software misinterpretation of the verb as a noun.

On the other hand, the read speech proved to be more problematic for the POS tagging process, thus requiring some adaptations for a more correct parsing.

In short, the answers to the RQs of this study are as follows: the two parsing tools showed different behaviours, and proved to be more appropriate for written language than for read speech. Alpino turned to be the parser that most has met the initial assumptions, also showing that adding the punctuation to the OTs of a read speech could be an adaptation capable of clearly increasing the accuracy of POS tagging.

Table 3: Punctuation code accuracy in Ots

Speakers	Full stops	Commas
sp01	26	3
sp02	26	3
sp03	25	3
sp04	27	4
sp05	27	3
sp06	27	2
sp07	28	3
sp08	26	4
Prompts	**30**	**4**

The following observations on the limits of this research should be made:

- the accuracy of the punctuation script is limited to the size of the small corpus used for the experiment. By way of example, Table 3 reports the number of commas and full stops that were successfully added with the script. Therefore, this script can be extended only to corpora of similar size (32 sentences), since the previous and subsequent words of the text are the parameters used to state where punctuation should be added. The more words the corpus have, the greater the risk of script malfunctioning.
- a statistical analysis would be the most appropriate way to analyse the data of this study and to obtain a scientifically valid result, but given the small sample size even a manual observation has allowed some interesting considerations.
- this work investigates the read speech of dysartrhic speakers, but for a complete analysis and more defined conclusions, also healthy speakers should be examined.

This study provides an insight on only one of the possible methods for a more accurate POS tagging of read speech, and points out some flaws of the POS tagging process, that in the case of Dutch language did not seem to have a uniform mechanism based on standardized criteria. Future research amongst different kinds of language might extend the explanations of the punctuation relevance and of the aspects that distinguish a kind of

language from the other (written language, spoken language, read speech etc.). Future studies on POS tagging could also investigate whether it is possible to have a unified parsing mechanism working properly for every kind of language or whether it would be more appropriate to have different mechanisms for each kind of language.

5 References

[1] Kumawat D. Jain V. (2015) POS tagging approaches: A comparison. International *Journal of Computer Applications*, 118(6).

[2] Brill E. (1994) Some advances in transformation-based part of speech tagging. arXiv preprint cmp-lg/9406010.

[3] Manning C.D. (2011) Part-of-speech tagging from 97% to 100%: is it time for some linguistics? Proc. of the *International conference on intelligent text processing and computational linguistics*, 171-189.

[4] Van den Bosch A., Schuurman I., Vandeghinste V. (2006) Transferring PoS-tagging and lemmatization tools from spoken to written Dutch corpus development. *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (LREC'06). Available from: http://www.lrec-conf.org/proceedings/ lrec2006/pdf/167_pdf.pdf.

[5] Bosch A. van den, Busser B., Canisius S., Daelemans W. (2007) An efficient memory-based morphosyntactic tagger and parser for Dutch. *LOT Occasional Series*, 7, 191-206.

[6] Bouma G., Van Noord G., Malouf R. (2000) Alpino: Wide-coverage Computational Analysis of Dutch, 37, 45-59.

[7] Van Gompel M., Reynaert M. (2013) FoLiA: A practical XML format for linguistic annotation – a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3, 63-81.

[8] Van Gompel M., Sloot K., Reynaert M., Van den Bosch A. (2017) FoLiA in Practice. The Infrastructure of a Linguistic Annotation Format.

[9] Van Gompel M., Bloem J. (2021) folia2columns. Script available at https://github. com/proycon/foliatools/blob/master/ foliatools/folia2columns.py.

[10] Pesenti C., van Bemmel L., van Hout R., Strik H. (2022). The effect of eHealth training on dysarthric speech. Paper accepted for the LREC 2022 workshop RAPID.

[11] Ganzeboom M., Bakker M., Beijer L., Rietveld T., Strik H. (2018) Speech training for neurological patients using a serious game. *British Journal of Educational Technology*, 49(4), 761-774.

[12] Ganzeboom M., Bakker M., Beijer L., Strik H. & Rietveld T. (2022) A serious game for speech training in dysarthric speakers with Parkinson's disease: Exploring therapeutic efficacy and patient sati-

sfaction. *International Journal of Language & Communication Disorders*, forthcoming.

[13] Koster J. et al. (1975) Dutch as an SOV language. *Linguistic analysis*, 1(2), 111-136.