CompAn – A Tool for Quantitative Comparison of Corpus Annotation

Vlasta Ohlídalová

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
akai@mail.muni.cz

Lexical Computing, Brno, Czech Republic
vlasta.ohlidalova@sketchengine.eu

Abstract. The paper is focused on automatic morphological annotation and its evaluation. The most common evaluation method is described as well as its main issues. Then, based on the theoretical part, a tool for quantitative comparison of corpus annotation (CompAn) is briefly introduced as an alternative to the traditional annotation evaluation based on gold standard corpora.

Keywords: tagging evaluation, gold standard, automatic POS tagging, inter-annotator agreement

1 Introduction

POS tagging is one of the most well researched areas of NLP: the first corpus to be automatically annotated was Brown and it was tagged with the TAGGIT tagger in 1971 [2]. It is also certainly one of the most widely used NLP techniques (both as the first step for developing other tools – such as syntactic analysis – and during linguistic research itself). The accuracy of taggers has been reported at 95–97%, depending on the language and many other variables. Yet similar results have been encountered since the end of the last century [1]. Does this mean that it is sufficient? And perhaps more importantly, is this a good numerical objective indicator of the success of the tools?

In this article I will discuss both of these questions. Based on the theoretical background and issues of the automatic evaluation, which is currently used the most, a tool assisting with the tagging evaluation will be introduced.

2 Is it not good enough yet?

Even using a very simple idea and implementation, the results of POS tagging are quite good (especially when compared to other areas of natural language processing); it currently achieves a success rate just a few percents below 100% – which is also very similar to the level achieved by annotator agreement in the

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2022, pp. 97–102, 2022.

manually tagged gold standards. Yet all of us who work with corpora know that obvious errors are still around.

And considering the assumed 97% accuracy the automatic taggers achieve, we encounter these errors surprisingly often.

So what is the reason why the real results are often far below the proclaimed ones, and why the accuracy of 97% might still not be enough?

1. The taggers accuracy is calculated from all tokens in the corpus, including, for example, punctuation, on which the success rate can very easily be close to 100%. Moreover, we need to keep in mind how frequent punctuation is: for example, in the English enTenTen15 [3] corpus five of the most frequent tokens are punctuation tokens (comma and period in the second and third places respectively), in the Czech csTenTen17 [3] corpus there are 6 of them in the top 20 (comma and period taking the first two places).
2. The accuracy rate will vary considerably depending on what texts we process. Perhaps the most important problem in this area is the fact that the tagger is usually evaluated on the same type of text it was trained on (although of course on different parts of that text). Thus, it is clear that when such a tagger is run on a different type of text, especially data with a lot of noise such as social network discussions, the resulting percentages may be quite different. The difference between the accuracy of several frequently used taggers on a corpus containing newspaper texts versus a corpus generated from the Web has been addressed throughout the work Evaluation of POS Tagging for Web as Corpus by Eugenie Giesbrecht [1]. As expected, all three taggers performed worse on the Web corpus, on average by about 2 percent.
3. The accuracy will also vary considerably depending on the specific genre of the text. In the aforementioned work, an evaluation of accuracy based on genres is also found. The percentages here vary by around 10% – from 88% to 98% (accuracy on each text and its genre is in detail described in Table 1).
4. If we want to build other tools on top of the tagger results, we are often not interested in the accuracy on token level, but rather accuracy on whole sentences (because even one incorrectly annotated token might confuse the tools working with the output). If we take a tagger success rate of 97% and the average sentence length according to the Brown corpus – which is 20 words – the probability of having an error in a sentence is close to 50% (precisely 45.6%). Looking on the issue from the other side, to achieve 95% correctness at the sentence level, we would need an accuracy of 99.6% at the token level – and this is perhaps the number which shows the best how far from it we are.

3 POS tagging evaluation

Evaluation can be theoretically run in many ways (automatic versus manual, formative versus summative, intrinsic versus extrinsic), but in reality, it is

Table 1: Statistics of TreeTagger POS tagging accuracy on various texts in the corpus DeWaC by their genres [1].

genre	overall accuracy
child infections (report)	98.25%
political speech (labor union)	97.52%
job market news	97.46%
news report (school district)	97.10%
scientific news/medicine	96.88%
history (Gold War) report	96.67%
story about Holy Paul	95.42%
biological exposition	94.23%
movie description	93.89%
IT news/Cebit	93.69%
news report (Archbishop)	91.97%
information about a conference	90.98%
Rolling Stones tour (forum)	88.01%

usually reported by comparing the tagger results to a gold standard. That is, the tagger is trained on a part of the manually tagged text and evaluated on another part of the same text (on a part which was not seen before by the tool). Success rate is then reported using accuracy.

Using this method, we might run into the following problems:

- As mentioned above, the genre and type of text plays a role in the final result. So we can assume that whenever a tagger is used in practice and the corpus is not very similar to the one the tagger was trained on, the results will differ. However, this is something which is not recognized at all in the result of this evaluation method.
- Since it is cheap to compare results of a tagger against a gold standard, the comparison can be run as many times as it takes to get the number you are happy with. The focus might therefore easily switch from actually improving the tool to having a number to publish.
- The correctness of the gold standard.

4 Gold standard

The problem with gold standard is that it is considered a fundamental truth and there is no mechanism to deal with the possibility of incorrect annotation. We assume that the labels are always right and never question it, because it is needed both for training a tagger and evaluating their results. A nice example of what inconsistent tagging (for which the Penn Treebank has been known) will ultimately produce is given by Manning [5]. In this paper, 100 mistakes made by the tagger were studied and categorized according to what caused the errors. Of the seven categories, 28% of the errors fell into the

category of "inconsistent/non-existent standard" and another 15.5% even into the "wrong gold standard" category. Together, these accounted for almost half of the errors (43.5% in total). In practice, we see both situations: inconsistency among annotators and mistakes in the gold standard.

4.1 Inter-annotator agreement

A huge problem in many NLP tasks is the (dis)agreement of the linguists themselves. And although we know that this is not as common in POS tagging as in other NLP tasks (especially those involving semantics), the problem exists here too, and given the usually high accuracy of POS tagging, it is important to address it; when we are at 95–97% accuracy, disagreement in 1% of all cases is still a lot and it might make someone wonder how reliable the measured accuracy actually is.

4.2 Incorrect annotation

In addition to disagreements, the ambiguity of the language might also lead to entirely incorrect annotations in the gold standard. A thorough examination of the manually annotated DESAM corpus [6] shows many errors too. For example, the following CQL query run on DESAM returns 13 sentences; of which in 10 cases adjectives are incorrectly annotated as nouns:

```
1:[tag="k1.*" & lemma="[[:lower:]].*ý"] 2:[tag="k1.*" &
lemma="[[:lower:]].*"] & 1.c=2.c within < s/>
```

Fig. 1: A few lines showing incorrectly annotated tokens in DESAM.

5 A tool for quantitative comparison of corpus annotation

The previous sections have described problems in automatic evaluation of morphological annotation that can – and often do – lead to inaccurate results. For this reason, the CompAn tool (the name comes from "compare annotations") has been created. Although the tool does not evaluate the quality of the morphological annotation, it compares the annotation of any attribute (at the

	Freq	rftagger	rftagger_synt	Conc	Conc
1	112	k1gInSc1	k4	🔗	🔗
2	69	k1gMnSc1	k1gInSc1	🔗	🔗
3	59	kF	k4	🔗	🔗
4	45	k1gInSc4	k4	🔗	🔗
5	39	k7c6	k7c4	🔗	🔗

Fig. 2: The example output of the tool when comparing attribute value (tags in this case)

moment it can be used for token, lemma or tag) in a single corpus processed by two different tools (such as a tokenizer, part-of-speech tagger or lemmatizer).

In practice, this means that the tool will list the most frequent differences in the annotations of the two tools, either by attribute value (i.e. which values were most often interchanged) or by word (i.e. which words were most often annotated differently). Examples of how the results are displayed in the interface are shown in Figure 2 and Figure 3.

	Freq	Word	rftagger	rftagger_synt	Conc	Conc
1	26	v	k7c6	k7c4	🔗	🔗
2	14	pondělí	k1gNnSc6	k1gNnSc4	🔗	🔗
3	8	na	k7c6	k7c4	🔗	🔗
4	7	top	kA	k5nSp2	🔗	🔗
5	7	to	k3gNnSc1	k3gNnSc4	🔗	🔗

Fig. 3: The example output of the tool when comparing words

Thus, the tool is not designed to produce one particular number that can be used as a universal indicator of annotation quality, but rather to assist in the manual comparison of two tools. The results are first pre-calculated and cached, so that they can retrieved instantly to be searched by different criteria.

The tool was developed as a RiotJS web application with a Python backend that uses corpora indexed by Manatee which is part of the (No)Sketch Engine corpus management suite [4,7]. Any indexed corpus can be instantly loaded and evaluated by CompAn.

6 Conclusions

In this paper CompAn is presented, an online tool for comparing annotations between two corpora. The main motivation behind the tool is comparison of part-of-speech annotation, lemmatization or tokenization, but it can be easily generalized for other purposes as well.

References

1. Giesbrecht, E.: Evaluation of POS Tagging for Web as Corpus. Master's thesis
2. Greene, B.B., Rubin, G.M.: Automated grammatical tagging of English. Dept. of Linguistics, Brown University (1971)
3. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen corpus family. In: 7th International corpus linguistics conference CL. pp. 125–127. Lancaster University (2013)
4. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The Sketch Engine: ten years on. Lexicography pp. 7–36 (2014)
5. Manning, C.: Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? vol. 6608, pp. 171–189 (05 2011). https://doi.org/10.1007/978-3-642-19400-9_14
6. Pala, K., Rychlý, P., Smrž, P.: DESAM – Annotated Corpus for Czech. In: Proceedings of SOFSEM '97. pp. 523–530. Springer-Verlag (1997)
7. Rychlý, P.: Manatee/Bonito-A Modular Corpus Manager. In: RASLAN. pp. 65–70 (2007)