Annotation systems and automatic processing:
a tight connection

Pablo Faria & Charlotte Galves
University of Campinas, Department of Linguistics

Contact Information:
Departamento de Linguística, IEL/UNICAMP
R. Sérgio B. de Holanda, 571, Campinas, SP

Phone: +55 (19) 3521 1570
Email: pablofaria@gmail.com
charlotte.mgc@gmail.com

Abstract

In this work, parsing and inconsistency detection results demonstrate that annotation systems can be revised to improve automatic processing of treebanks. In addition, an alternative parsing evaluation measure – used complementary to PARSEVAL – is introduced. We conclude with some suggested guidelines for the specification of annotation systems and the preparation of training corpora, which aim to benefit parsing and quality control tasks.

Introduction

In a moment at which we put our efforts in the processing and availability of massive amounts of linguistic data, it is important to avoid wasting time and resources by making certain mistakes in the process of building a treebank. Below, we explore parsing evaluation and inconsistency detection results for the Tycho Brahe Parsed Corpus of Historical Portuguese (TBC, [3]) to show that parsing and quality control are more efficient if annotation systems are more consistent, informative and concise, in some very specific ways.

Less tight than it should

Phrase structure annotation systems, in particular, are often designed without deep concern of how its properties may affect different kinds of automatic processing. In TBC there is a "tightness" mismatch between the pos tag system and the syntactic layer annotated on top of it. Various factors contribute(d) to this (unawareness of the problem, independent development of each annotation, etc.). Let us take the current use of *base* and *dash* tags in TBC, for instance:

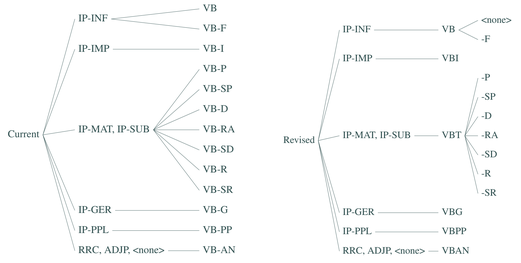

Figure 1: Revision of the TBC verbal part-of-speech tag system.

Could parsing and inconsistency detection tasks benefit from the revised system? Three TBC subsystems of part-of-speech (pos) tags were chosen for investigating this: verbal, nominal, and punctuation systems. The guiding principle: keep in (or add to) the *base tag* only the information directly relevant for the syntactic layer and leave others as dash tags. Relative to the nominal system and to punctuation, common and proper nouns are reduced to "N" (plus a dash tag "-PR" for the latter); clitics, the clitic "se", and demonstratives are reduced to "PRO" plus dash tags ("-CL, -SE, -DEM" respectively); and, finally, punctuation (except for parenthesis) is reduced to PUNC, instead of "," (intermediary) and "." (final).

Improving parsing

A series of parsing evaluations (using Bikel's parser, [1]) were conducted to test both the effects of changing the nominal and verbal pos tag system, and of the absence (partial or total) of dash tags from the training corpus (and of the test file, of course).

Conditions	N/W	Cross	F1	F1_40	wF1	RC	RC_40 ↑	wRC
verb-pdash	1.69	1.15	77.70	80.03	82.62	0.291	0.263	0.234
verb-un-rev-rvpdash	2.05	2.02	76.86	80.26	83.51	0.308	0.268	0.233
verb-un-rev-pdash	2.05	2.05	76.83	80.16	83.48	0.309	0.271	0.234
verb-rvpdash	1.69	1.24	76.79	79.31	82.25	0.304	0.273	0.239
verb-un-rvpdash	2.05	2.05	76.74	80.13	83.51	0.313	0.273	0.237
verb-un-pdash	2.05	2.06	76.89	80.08	83.47	0.312	0.275	0.239
verb-noun-rvpdash	1.69	1.24	76.73	79.05	82.04	0.306	0.278	0.243
base-un-rev-pdash	2.05	2.07	76.05	79.32	82.78	0.319	0.280	0.242
base-un-pdash	2.05	2.10	75.92	79.20	82.67	0.323	0.285	0.249
verb-noun-pdash	1.69	1.31	75.87	78.35	81.27	0.317	0.286	0.251
verb	1.69	1.42	75.08	**77.59**	80.86	0.330	0.299	0.261
punc-pdash	1.69	1.35	74.82	77.40	80.51	0.331	0.300	0.263
punc	1.69	1.43	74.54	**77.35**	80.23	0.336	0.301	0.266
verb-noun	1.69	1.43	74.87	**77.36**	80.36	0.334	0.304	0.274
noun-rvpdash	1.69	1.44	74.34	77.06	80.12	0.338	0.305	0.267
noun	1.69	1.45	74.17	**77.06**	79.69	0.342	0.307	0.280
noun-pdash	1.69	1.40	74.13	76.84	79.75	0.341	0.307	0.272
verb-un	2.05	2.07	74.64	78.47	81.95	0.352	0.308	0.267
base-un	2.05	2.08	74.51	78.30	81.71	0.352	0.309	0.269
base-pdash	1.69	1.82	67.58	70.76	75.82	0.431	0.394	0.335
base	1.69	1.88	67.49	**70.63**	75.77	0.437	0.401	0.339
verb-dash	1.69	1.39	80.30	83.43	86.72	0.623	0.608	0.607
verb-noun-dash	1.69	1.39	80.24	83.26	86.63	0.623	0.609	0.608
noun-dash	1.69	1.43	79.80	82.87	86.54	0.627	0.612	0.606
punc-dash	1.69	1.42	79.96	83.00	86.52	0.627	0.613	0.607
base-dash	1.69	1.74	74.01	77.38	83.15	0.688	0.670	0.641

Table 1: Parsing evaluation conditions: current (*base*), punctuation revised (*punc*), revised verbal system (*verb*), and revised nominal (*noun*), the latter two based on the *punc* condition. Other derived conditions: syntactic dash tags as unary projections (*-un*/*-un-rev*), all dash tags removed (*-dash*), only pos dash tags removed (*-pdash*), and only revised dash tags removed (*-rvpdash*).

Figure 2: Some examples of experimental conditions: (a) *base*; (b) *base-dash*; (c) *verb*; (d) *verb+unary*; (e) *verb+unary(reverse)*

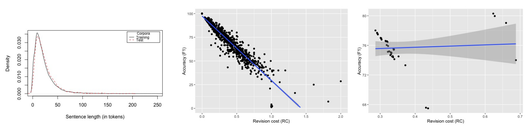

Figure 3: (a) Sentence length distributions in training and test corpora; (b) Correlation between the F-*score* measure (F1) and the revision cost (RC), for condition "punc" (-0.9488554, $p < 2.2e\text{-}16$); (c) Accuracy x Revision Cost between conditions.

The revision cost measure

It simulates the revision process operations (node insertion, deletion, movement, and relabeling). The cost is calculated as the ratio of the set of operations needed to fix a tree (given a source and a target tree) by the cost of building it manually. It is complementary to PARSEVAL ([6]). As Figure 1b shows, it is highly correlated with PARSEVAL when *target* is the *gold* tree, but (as 1c shows) it is not correlated when *target* is not. This non-correlation is interesting, for instance, when calculating how much work is needed to go from a dashless tree (*-dash* above) to a fully annotated one.

Discussion

Main aspects highlighted from the above results:

- **Punctuation**. The surprising improvement from *base* to *punc* condition demonstrates the importance of punctuation in the annotation. Our hypothesis is that, by simplifying its annotation, its impact on the probabilities for other items is minimized.
- **Unary projections**. An even better improvement than *punc* for the *base* condition, with reversed conditions showing higher accuracy. Some studies for German suggest that unary projections may benefit parsing ([4], but see [5] for contrary results). It is not clear why there was no improvement for the *verb* condition.
- **POS dash tags**. These are not helping the parser, at least in TBC. In Table 1 we see that *-pdash* and *-rvpdash* conditions produce better results overall, both for accuracy and revision cost.
- **Dashless training**. Although obtaining the highest accuracy results, revision costs show that they are not worth it.
- **Nouns x Verbs**. It is interesting that revision of verbal tags improved parsing but the revised nominal system did not. One possibility is that, in TBC, it is very likely to find more than one nominal element immediately dominated by an NP, contrary to verbs. Then, instead of highlighting the one element (the head) in its context, the revision is actually increasing the number of same tag elements.
- **Limited improvement**. There seems to be some slight additive effects (e.g., *noun x verb-noun*) for combined conditions, but not for *verb* and *-un* (each with strong impacts, in isolation). As closer a condition gets to 80%, the smaller the impact of other changes. Maybe a limitation of the parser, of the training corpus, or both.

Quality control issues

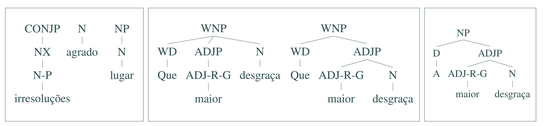

Figure 4: (a) A variation nucleus with an N tag as a non-head; (b) Another nucleus which excludes NP variants, because of its W feature as part of the base tag; (c) A variant excluded from (b).

In automatic inconsistency detection, we try to detect variance in annotation that could be, if not a case of ambiguity, a result of error in annotation or of inconsistency in the annotation system itself. In [2], Faria's method detects inconsistencies by comparing annotations for repeating sequences of tokens. The goal is to detect nuclei of variation, that is, two or more variants of annotation for equivalent strings. Thus, a consistent and tight annotation system is crucial to improve generalization routines. For instance, more explicitness in phrase-head relations and base tags that capture categorial equivalences, would improve recall and precision.

Conclusions

There are several ways of improving the automatic processing of treebanks. Revising the annotation system and trying different training strategies are two of them. After all, better automatic processing means faster treebank building. In sum, we suggest some general guidelines:

- When planning or revising the annotation system, use base tags (pos and syntactic) as the locus of categorial equivalences and phrase-head relationships, leaving all else as features annotated as dash tags.
- Remove all superfluous and automatically recoverable material from the training corpus, for instance, pos dash tags.
- Consider the possibility of using unary projections to represent information in syntactic dash tags. The conversion in both directions is straightforward.

For the future, we plan to investigate the learning curve of the parser (for different conditions), effects of alternative annotation of coordinations, and effects of a deep revision of inconsistencies in TBC.

References

[1] D Bikel. *On The Parameter Space of Generative Lexicalized Statistical Parsing Models*. PhD thesis, Computer and Information Science, University of Pennsylvania, 2004.

[2] Pablo Faria. Increased recall in annotation variance detection in treebanks. In *Text, Speech, and Dialogue*, pages 578–586. Springer International Publishing, 2015.

[3] Charlotte Galves and Pablo Faria. Tycho brahe parsed corpus of historical portuguese. Available online at http://goo.gl/cu4N6w., 2010.

[4] Sandra Kübler. How do treebank annotation schemes influence parsing results? or how not to compare apples and oranges. In *Proceedings of RANLP, Borovets, Bulgaria, September 2005.*, 2005.

[5] Ines Rehbein and Josef van Genabith. Treebank annotation schemes and parser evaluation for german. In *Proceedings of the Joint Conference EMNLP and CNLL, Prague, June 2007.*, pages 630–639, 2007.

[6] Satoshi Sekine and Michael John Collins. Evalb software. Online at http://nlp.cs.nyu.edu/evalb/, 2013.

Acknowledgements

Thanks to Sao Paulo Research Foundation – FAPESP – for funding this research through grants no. 12/06078-9, 13/18090-6 and 14/17172-1.