Annotation systems and automatic processing: a tight connection

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

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—use complementary to PARSEVAL—is introduced. We conclude with some suggested guidelines for the specific purpose of annotation revision 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 TreeBank 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.

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 tags system, and of the absence (or partial or total) of dash tags from the training corpus and of the test file, of course.

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 1 shows, it is highly correlated with PARSEVAL when target is the gold tree, but (as it 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-free) to a fully annotated one.

Discussion

Main aspects highlighted from the above results:

• Punctuation. The smaller improvement from base to parse 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 parse 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 -dash and -vpdash 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 noun-name) for combined conditions, but not for verb and noun (each with significant impacts, in isolation). As closer a condition gets to 80%, the smaller the impact of other changes. Maybe a limitation of the parser, at least in TBC. In Table 1 we show that dashless trees are not helping the parser, at least in TBC.

Conclusion

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 categorical equivalences and phrase-internal relations, leaving all else as features annotated as dash tags.

• Remove all superficial 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 coordinative relations and base tags that capture categorical equivalences, would improve recall and precision.

Quality control issues

In automatic inconsistency detection, we try to detect variation 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-related relations and base tags that capture categorical equivalences, would improve recall and precision.

References


Acknowledgements

Thanks to Sao Paulo Research Foundation – FAPESP – for funding this research through grants no. 12/00678-9, 13/18099-6 and 14/17172-1.

Contact Information:
Departamento de Linguística, IEL/UNICAMP
R. Sérgio de Bó, 571, Campinas, SP
Phone: +55 (19) 3521 1570
Email: pablofaria@gmail.com
charlotte.mg@gmail.com

Improving parsing

Phrase structure annotation systems, in particular, are often designed to "PRO" plus dash tags ("-CL, -SE, -DEM" respectively); and, finally, punctuation (except for parenthesis) is reduced to PUNC, instead of "(" (intermediate) and ")" (final).

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 1 shows, it is highly correlated with PARSEVAL when target is the gold tree, but (as it 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-free) to a fully annotated one.

Discussion

Main aspects highlighted from the above results:

• Punctuation. The smaller improvement from base to parse 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 parse 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 -dash and -vpdash 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 noun-name) for combined conditions, but not for verb and noun (each with significant impacts, in isolation). As closer a condition gets to 80%, the smaller the impact of other changes. Maybe a limitation of the parser, at least in TBC. In Table 1 we show that dashless trees are not helping the parser, at least in TBC.

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 categorical equivalences and phrase-internal relations, leaving all else as features annotated as dash tags.

• Remove all superficial 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 coordinative relations and base tags that capture categorical equivalences, would improve recall and precision.

Quality control issues

In automatic inconsistency detection, we try to detect variation 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-related relations and base tags that capture categorical equivalences, would improve recall and precision.

References


Acknowledgements

Thanks to Sao Paulo Research Foundation – FAPESP – for funding this research through grants no. 12/00678-9, 13/18099-6 and 14/17172-1.