Attachment Errors of Nouns after Possessor Clitic

Ercan Solak[1], Olcay Taner Yıldız[1], Onur Görgün[1,2], and Razieh Ehsani[1]

[1] Işık University, Istanbul, Turkey
[2] Alcatel Lucent Teletaş Telekomünikasyon A.Ş., Istanbul, Turkey

Abstract. In this paper, we analyze the errors of NP attachments that occur when they combine with NP's ending in possessor clitic. We suggest a simple pattern based detection and correction solution. We illustrate the errors with examples from Penn Treebank.

1 Introduction

With the help of the annotated corpora of different size and nature, statistical NLP research has come a long way over the past three decades. Part-of-speech tagged Brown corpus [1], syntactically annotated Penn-Treebank [2] and sentence aligned Europarl corpus [3] are arguably the most-cited corpora in their respective subfields of statistical NLP research. Creating and maintaining such huge corpora need a lot of manual and semi-manual effort. The annotation errors may have a negative influence on the performance of the corpus-based research.

To overcome this problem, several error detection and/or correction algorithms are proposed [4]. Early works concentrate on the detection of errors in the POS-annotation [5], and they were followed by the works focused on the detection of errors in the syntactic structure [6]. Not only the users of the corpora, but also the maintainers of the corpora are also interested in detecting annotation errors. For example, Linguistic Data Consortium uses Tree Adjoining Grammar (TAG) to check treebank consistency, which is applied on Arabic and English treebank [7].

In this paper, we focus on the attachment errors that occur when NP attachments are combined with the possessor clitic. We give examples and statistics from the Penn Treebank. We propose a simple correction algorithm based on the specific pattern of the error.

This paper is organized as follows: We give the related work on annotation error detection/correction in Section 2. After reviewing very briefly the attachment errors in Section 3, we will discuss the relationship between NP and PP, and show how this relationship can cause errors in the annotation. We illustrate the errors with example trees from Penn Treebank in Section 5 and illustrate its implications for tree based translation. We describe an error correction algorithm in Section 6. Finally, we conclude in Section 7.

2 Related Work

[8] is one of the earliest works in the annotation error detection. They divide corpus errors into three; (i) detectable errors which can be automatically detected and fixed,

(ii) fixable errors which require human intervention at some point in the correction process, and (iii) other cases, where the markup guidelines do not give any hint to the annotators and leave them to their own intuitions.

In general, annotation error literature can be divided into two; error detection approaches and error correction approaches. [9, 10, 5] detect errors in the POS annotation; [11–14, 7] detect errors in the constituent structure; [6, 15] detect errors in the dependency structure in the treebanks. On the other hand, [9, 16] can correct POS annotation errors; [4] can correct errors in the constituent structure.

3 Attachment errors

Attachment ambiguity is a source of error for automatic constituent parsers that rely on structural information. A common example is the sentence "I saw the man with a telescope". The PP "with the telescope" can attach either under the NP "the man with a telescope" or under VP "saw the man with the telescope". Obviously, a bit of world knowledge often resolves the ambiguity. When we replace "telescope" with "suitcase", it is clear where the attachment goes.

4 NP and PP

Consider the sentence "I read John's book of quotations." The Berkeley parser [17] gives the parse tree given in Figure 1.

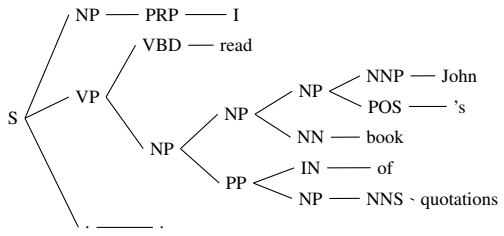


Fig. 1. The parse of the sentence "I read John's book of quotations."

The PP "of quotations" clearly qualifies the book that happens to belong to John. Now consider the alternative parse in Figure 2.

Again, "of quotations" qualifies the the book. But this time, parse tree attaches "book" with the PP first, before attaching the resulting NP subtree to "John's". The difference between the two cases seems minor. Indeed, in both cases, John has a book and the book has lots of quotations in it.

Let us slightly change the sentence while trying to keep the same semantics. "I read the book of quotations of John." So, we got rid of the possessive clitic and tried to express the sentence with a uniform use of "of"s. But this introduces a genuine ambiguity.

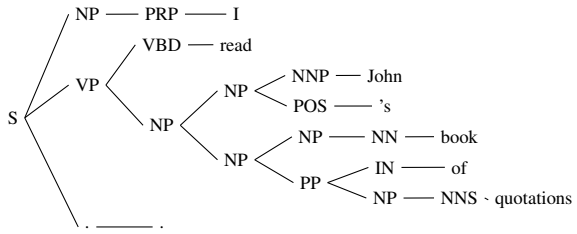


Fig. 2. An alternative parse of the sentence "I read John's book of quotations."

Does John have a book that has quotations by many famous people? Or is this a book that contains John's sayings exclusively?

Interestingly, both Berkeley and Stanford parsers [18] think the latter interpretation is more likely, although they slightly disagree on whether to use VP→VBP NP or VP→VBP NP PP. The Berkeley parse for this reading of the sentence is given in Figure 3.

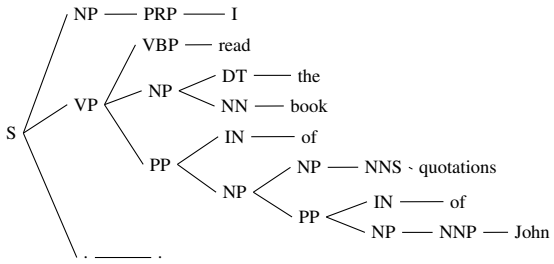


Fig. 3. The most likely Berkeley parse of the sentence "I read the book of quotations of John."

Thus, there are two opposing tendencies in the parsers for NPs expressing possession. When the possession is expressed with a clitic "'s", the parser tries to attach the next noun under the running NP. When "of" is used, the parser tries to group what comes after "of" into a NP first.

5 Penn Treebank

In this section, we focus on the case that has clitic. The incorrect constituent structure has a three layer template given in Figure 4. A,B and C are nouns and d is a preposition.

In order to see the frequency of the incorrect attachment in manual annotation, we searched for the incorrect template in Figure 4 among the 43908 sentences in the Penn Treebank. 614 of those fit the incorrect template. That is 1.4%, fairly high as the attachment errors go.

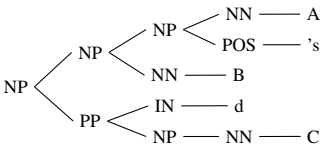


Fig. 4. Template tree structure of incorrect noun attachment after the possessive clitic.

6 Implications for translation

We noticed this NP attachment ambiguity when we were building a parallel treebank between Turkish and English [19]. In constructing the parallel corpus, we used a subset of trees in the Penn Treebank and translated them into Turkish using a sequence of only two operations. One operation permutes the children of a node and the other replaces a leaf with a stem or a morpheme.

Given the regularity of constituent order in English sentences and the regularity morphotactics of Turkish, it is possible to translate many sentences between the two languages using only the permute and replace operations.

An example translation is given in Figure 5.

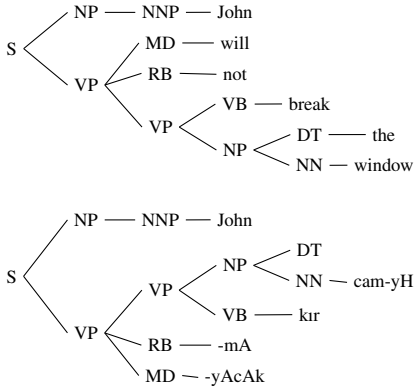


Fig. 5. The permuting of the nodes and the replacement of the leaves by the stems or morphemes.

Turkish uses postpositional morphemes to construct genitive-possessive noun phrases. In the construction A-GEN B-POSS, where A and B are nouns, A is the possessor and B is the entity possessed by A. GEN and POSS suffixes agree in their person markers. GEN-POSS construction can be chained as A-GEN B-POSS-GEN C-GEN. In this chain, A possesses B and B possesses C.

When we translate English trees to Turkish using only permutation and replacement, both the clitic "'s" and the preposition "of" is replaced with -GEN morpheme. -POSS morpheme is added to the possessed NP. Thus, food example, we have

(1) kapı-nın kol-u
door-GEN handle-POSS
handle of door

The trees for this pair are given in Figure 6.

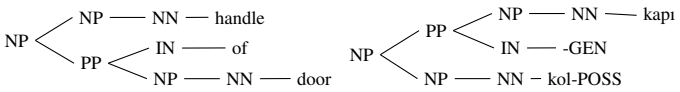
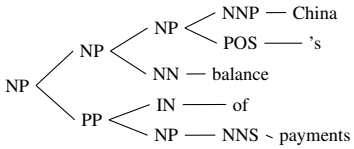


Fig. 6. Parse trees for the sentence pair in (1).

For an error prone example, consider the following NP subtree taken from the parse of the sentence "With less capital coming in, China's balance of payments will suffer".



Semantically, "of payments" qualifies the "balance" as there is no entity "China's balance" that can stand on its own. However, in the parse tree, the annotators preferred an early attachment and created this entity. For correct literal translation of this NP to Turkish we have,

(2) Çin-in ödemeler-i-nin denge-si
China-GEN payments-POSS-GEN
balance-POSS

When we replace the functional words the English tree with their Turkish morphemes only, we obtain the following tree.

Clearly, there is no way this tree can be permuted to read

(3) China-GEN payments-POSS-GEN
balance-POSS

If the original tree had the correct attachment, however, it could be easily permuted to the tree in Figure 7 with the correct order.

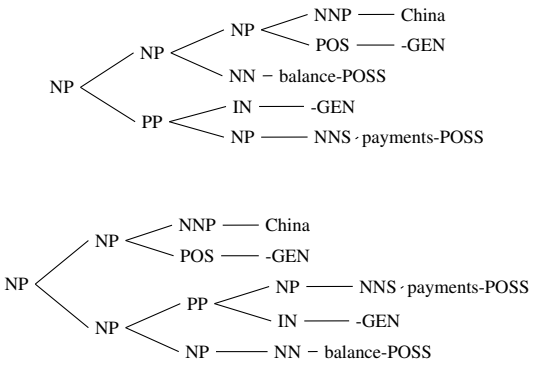
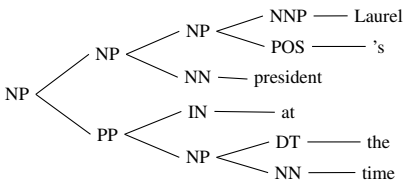


Fig. 7. Correct subtree of sentence (3).

We give two more examples of NP phrases from the Penn Treebank with their correct literal Turkish translations. These translations can not be obtained from the English tree through permutation and replacement. Note that PPs in these examples contain prepositions “at” and “for”.

(4) Laurel-in o zaman-ki başkan-ı
Laurel-GEN that time-REL
president-POSS
Laurel’s president at the time.



(5) Girişim-in Thomson için önem-POSS
Venture-GEN Thomson for
importance-POSS
The venture’s importance for Thomson

7 Remedy

We can detect with a template the particular class of attachment errors we analyzed in this work. In order to correct the relevant subtree, we need to identify and move parts of it. The general incorrect pattern is given in Figure 8.

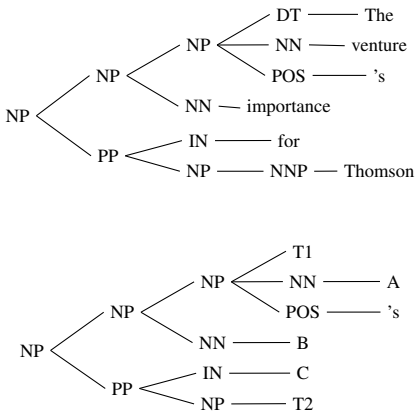


Fig. 8. Pattern for incorrect noun attachement.

Here, T1 and T2 denote nodes possibly with their own subtrees. A, B and C are terminal symbols. In order to correct the attachment error, the constituents T1, T2, A, B and C must be moved around such that now the tree has the structure in Figure 9.

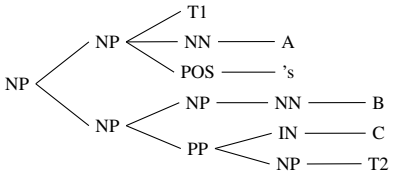


Fig. 9. Corrected tree.

Comparing the Figures 8 and 9, we see that the correction changes the counts of rules that contain NP on the left hand side. It increases the counts of NP→NP NP and NP→NN and decrease the count of NP→NP NN. We observed that this creates a tendency to introduce spurious NP hierarchies.

8 Experiment

To see the effect of these attachment errors on the parser performance, we trained the Stanford parser with the original and corrected data from WSJ section of the Penn Treebank. When we use original data for both training and test, we get an F score of 85.53. When we correct the attachment errors in both the training and test data, the F score becomes 85.4.

Interestingly, even though we corrected the attachment errors in the training treebank, the parser still consistently makes the attachment error in the test set. This is probably due to the tendency of the parser to attach NN subtree as a child to the preceding NP rather constructing a new NP above NN. The changing counts after the correction do not seem to be enough to reverse this tendency.

9 Conclusion

Phrase attachment ambiguities are sources of parse errors as well as funny newspaper clippings. In many cases, the ambiguity can be resolved using contextual and lexical constraints. In other cases, the errors are regular and follow a pattern.

In this work, we analyzed a particular class of noun phrase attachment errors. We found that this error occurs in about 1.4% of the sentences in the Penn Treebank. We showed an implication of the error for tree based translation. Finally, we suggested a template to detect the error in the Treebank and a simple rearrangement to put the constituents in their proper places. The score with the correct data is slightly lower.

References

1. Francis, W.N., Kucera, H.: Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US (1979)
2. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics **19** (1993) 313–330
3. Koehn, P.: Europarl: A multilingual corpus for evaluation of machine translation. (2002)
4. Kato, Y., Matsubara, S.: Correcting errors in a treebank based on synchronous tree substitution grammar. In: Proceedings of the ACL 2010 Conference Short Papers. ACLShort '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 74–79
5. Dickinson, M., Meurers, D.: Detecting errors in part-of-speech annotation. In: 10th Conference of the European Chapter of the Association for Computational Linguistics, The Association for Computer Linguistics (2003) 107–114
6. Boyd, A., Dickinson, M., Meurers, D.: On detecting errors in dependency treebanks. Research on Language and Computation **6** (2008) 113–137
7. Kulick, S., Bies, A., Mott, J.: Further developments in treebank error detection using derivation trees. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, European Language Resources Association (ELRA) (2012)
8. Blaheta, D.: Handling noisy training and testing data. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. EMNLP '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 111–116
9. Eskin, E.: Detecting errors within a corpus using anomaly detection. In: 1st North American chapter of the Association for Computational Linguistics Conference. (2000) 148–153
10. Nakagawa, T., Matsumoto, Y.: Detecting errors in corpora using support vector machines. In: 19th International Conference on Computational Linguistics. (2002) 709–715
11. Dickinson, M., Meurers, W.D.: Detecting inconsistencies in treebanks. In: Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003), Växjö, Sweden (2003) 45–56

12. Ule, T., Simov, K.: Unexpected productions may well be errors. In: 4th International Conference on Language Resources and Evaluation, European Language Resources Association (2004) 1795–1798
13. Dickinson, M., Meurers, W.D.: Prune diseased branches to get healthy trees! How to find erroneous local trees in a treebank and why it matters. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005), Barcelona, Spain (2005)
14. Kulick, S., Bies, A., Mott, J.: Using derivation trees for treebank error detection. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2. HLT '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 693–698
15. Volokh, A., Neumann, G.: Automatic detection and correction of errors in dependency treebanks. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2. HLT '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 346–350
16. Murata, M., Utiyama, M., Uchimoto, K., Isahara, H., Ma, Q.: Correction of errors in a verb modality corpus for machine translation with a machine-learning method. ACM Transactions on Asian Language Information Processing **4** (2005) 18–37
17. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. (2006) 433–440
18. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. (2003) 423–430
19. Yıldız, O.T., Solak, E., Görgün, O., Ehsani, R.: Constructing a Turkish-English parallel treebank. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, Association for Computational Linguistics (2014) 112–117