Treebanks Gone Bad: Generating a Treebank of Ungrammatical English

Jennifer Foster
National Centre for Language Technology
School of Computing
Dublin City University
jfoster@computing.dcu.ie

Abstract

This paper describes how a treebank of ungrammatical sentences can be created from a treebank of well-formed sentences. The treebank creation procedure involves the automatic introduction of frequently occurring grammatical errors into the sentences in an existing treebank, and the minimal transformation of the analyses in the treebank so that they describe the newly created ill-formed sentences. Such a treebank can be used to test how well a parser is able to ignore grammatical errors in texts (as people can), and can be used to induce a grammar capable of analysing such sentences. This paper also demonstrates the first of these uses.

1 Introduction

This paper describes how a treebank of ungrammatical sentences can be created from a treebank of well-formed sentences. The treebank creation procedure involves the automatic introduction of frequently occurring grammatical errors into the sentences in an existing treebank, and the minimal transformation of the analyses in the treebank so that they describe the newly created ill-formed sentences. Such a treebank can be used to test how well a parser is able to ignore grammatical errors in texts (as people can), and can be used to induce a grammar capable of analysing such sentences. This paper demonstrates the first of these uses – a popular Wall-Street-Journal-trained statistical parser[Bikel, 2004] is evaluated on an ungrammatical version of Section 23 of the Wall Street Journal (WSJ) portion of the Penn-II Treebank [Marcus *et al.*, 1993; 1994].

The idea of an ungrammatical treebank is motivated in Section 2 of this paper, the process of creating such a treebank is described in Section 3, and, in Section 4, the results of creating an ungrammatical WSJ Section 23 and using this to evaluate a widely used parser's ability not to be side-tracked by grammatical errors are presented. Section 5 proposes further work in this area.

2 Motivation

A corpus of ungrammatical sentences is a useful resource, both as a source of evidence for the kind of ill-formed structures that tend to occur in language, and as a source of test and training data for parsers which aim to accurately analyse sentences containing grammatical errors. Since people are able to comprehend text containing grammatical errors, it is reasonable to expect a parser to behave in the same way. An error corpus can take the form of a learner corpus [Granger, 1993; Emi *et al.*, 2004] or a more general form of error corpus, created by scanning texts for errors [Becker *et al.*, 1999; Foster, 2005]. Learner corpora are particularly useful in the study of second language acquisition since they provide insight into the difficulties faced by native speakers of a particular language when attempting to learn the corpus language. The more general form of error corpus is unconcerned with whether an error reflects linguistic competence or performance, it merely records that an error has occurred. Unfortunately, the compilation of both kinds of error corpus is a slow process, because it is not enough to merely collect a body of sentences, the grammaticality of each sentence must also be judged in order to determine whether an error has occurred. If an error has occurred, it then must be classified according to some error taxonomy.

A usefully large error corpus, in which every sentence is guaranteed to contain a grammatical error, can be quickly created by automatically introducing errors into a corpus of grammatical sentences. In order to ensure that this transformation process is rooted in linguistic reality, it should, of course, be based on an analysis of naturally produced grammatical errors. An interesting aspect of the automatically induced error corpus is its parallel nature, since the meaning of the ungrammatical sentence can be found by looking at its grammatical counterpart.

An even more useful resource for the devising and testing of robust parsers, is a treebank of ungrammatical sentences. Of course, the creation of any treebank is a costly, laborious task. However, assuming the existence of a treebank of grammatical sentences and a corpus of ungrammatical sentences derived automatically from the sentences in the grammatical treebank, it is possible to automatically create a treebank of ungrammatical sentences. This treebank can then be partitioned in the usual way, into a set of gold standard reference parses and a set of training parses for any data-driven probabilistic parser.

The idea of an automatically generated error corpus is not new. [Bigert, 2004], for example, automatically introduces context-sensitive spelling errors into texts. The idea of a tree-

bank of ungrammatical sentences has been explored before by [Kepser *et al.*, 2004], who were responsible for compiling SINBAD, a treebank of German sentences which have been judged to be grammatically deviant by linguists. The SINBAD treebank differs from the type of ungrammatical treebank which would be produced by the method described in this paper because it is designed to be used more as an informational source for generative linguists rather than as a set of training/test data for a robust parser. It is created manually rather than automatically, and is, thus, limited in size.

3 Creating a Treebank

This section describes the procedure for creating an ungrammatical treebank. This procedure involves two steps: the first is the introduction of grammatical errors into the sentences in a treebank; the second is the transformation of the original analyses into gold standard analyses for the newly created ungrammatical sentences. The first step is described in Section 3.1, and the second in Section 3.2. The procedure is discussed in a manner independent of any particular treebank or annotation scheme because it is theoretically possible to apply it to any kind of treebank.

3.1 Automatic Error Creation

The error creation procedure takes as input a part-of-speech tagged corpus of sentences which are assumed to be well-formed, and outputs a part-of-speech tagged corpus of ungrammatical sentences. The automatically introduced errors are divided into the following classes:

1. missing word errors: *She didn't* ***want*** *to face him* > *She didn't to face him*
2. extra word errors: *Do you ever go and visit* ***any*** *of them?* > *Do you ever go and visit* ***the any*** *of them?*
3. context-sensitive spelling errors: *I love* ***them*** *both* > *I love* ***then*** *both*
4. agreement errors: *The* ***contrast was*** *startling* > *The* ***contrasts was*** *startling*

The decision to introduce errors of the above types was made on the basis of the error analysis carried out by [Foster, 2005] which found that 72% of all errors occurring in a manually constructed 20,000 word corpus of naturally occurring ungrammatical written English sentences (from newspapers, emails, internet forums and academic papers) fall into one of these four classes. An example which does not fall into one of these classes is a more complex error requiring more than one application of the insertion/substitution/deletion correction operations to be corrected, e.g. *Phrase structure trees* ***allows describing*** *the syntax of sentences*. Another example which does not fall into one of the above four classes is: *It captures quite* ***plausibility*** *a form of life today.* Like the agreement and context-sensitive spelling errors, this is corrected by substituting one word for another, but the relationship between the substituted and substituting word does not involve grammatical agreement or easily confused spelling.

For each sentence in the original tagged corpus, an attempt is made to automatically produce four ungrammatical sentences, one for each of the four error types. Thus, the output of the error creation procedure is, in fact, four error corpora.

Missing Word Errors

Missing word errors can be classified on the basis of the part of speech of the missing word. In the error corpus described by [Foster, 2005], 90% of the missing word errors involve the omission of the following parts of speech (ordered in decreasing frequency): *det* > *verb* > *prep* > *pro* > *"to"* > *conj*. Most of the remaining 10% involve missing nouns, but nouns cannot be omitted automatically in a straightforward manner, because, in the case of noun-noun compounds, for example, the omission will still result in a well-formed sentence.[1] Missing word errors are introduced by searching a part-of-speech tagged sentence for all occurrences of words with the above part-of-speech tags and then deleting one from the sentence. The frequency ordering shown above is respected so that the resulting error corpus will contain, for example, more missing determiners than missing pronouns. In the unlikely event that a sentence contains none of the above parts of speech, no ungrammatical sentence is produced.

Extra Word Errors

Based on the error analysis carried out by [Foster, 2005], extra word errors are divided into the following classes:

1. repeated word errors: *All this is five or* ***or*** *six years ago.*
2. double syntactic function errors: *the draught coming in* ***of*** *under the door*
3. unnecessary word errors: *The link between social status and government appointments* ***and*** *was less rigid.*

The procedure considers each of these subclasses of extra word error equally likely, and attempts to insert one of them into a grammatical sentence. It uses a pre-compiled list of function words generated from a part-of-speech tagged corpus to introduce double syntactic function errors, and it uses a pre-compiled list of function and content words to introduce an unnecessary word error. It will not be possible to insert a double syntactic function error into a sentence that contains no function words, but it will always be possible to insert errors of the other two subclasses, since these involve the random insertion of an arbitrary word or a word already in the sentence.

Context Sensitive Spelling Errors

An error is classified as a context-sensitive spelling error[2] if it can be corrected by a word similar to it in spelling. Two words are considered similar in spelling if the Levenshtein distance between them is one (e.g. *to* and *too*) ([Foster, 2005]). One could argue that sentences containing context-sensitive spelling errors are not ungrammatical because the error involves the orthography of the word rather than some syntactic feature such as case. However, they should be considered ungrammatical because they are a real problem for

[1] The error creation procedure could, of course, be improved by deleting a noun from a sentence when it occurs on its own in a noun phrase.

[2] Context-sensitive spelling errors are also known as *real word* errors (see for example [Ingels, 1996]).

parsers. Again following the error analysis carried out by [Foster, 2005], a list of candidate English context-sensitive spelling errors is pre-compiled. The error creation procedure searches for all words in the input sentence which can be replaced by a word similar in spelling (subject to the pre-compiled list): one of these is then randomly selected and replaced. The pre-compiled list contains very common English words such as *a*, *the* and *he*, and an ungrammatical sentence can be generated from most sentences.

Agreement Errors
The error creation procedure attempts to introduce subject-verb or determiner-noun agreement errors into a sentence. For English, the procedure is at its least productive for this error type, because it can only introduce a subject-verb agreement error when the sentence contains a present tense verb, and a determiner-noun agreement error when the sentence contains a determiner which is marked for number (e.g. a demonstrative or indefinite article). It would produce more ungrammatical sentences if applied to a more morphologically rich language.

"Grammatical" Erroneous Sentences
The error creation procedure can sometimes introduce an error into a grammatical sentence in such a way that, instead of producing an ungrammatical sentence, it produces another grammatical sentence, often with a different (and usually implausible) meaning. The extent to which this occurs was estimated by carrying out the following small experiment: over 100 sentences were randomly extracted from the British National Corpus [Burnard, 2000] and the error creation procedure applied to them. 400 of the resulting ungrammatical sentences (the first 100 for each error type) were then manually inspected to see if the sentence structures were grammatical. The percentage of grammatical structures that are inadvertently produced for each error type and an example of each one are shown below:

- Agreement Errors, 7%
  *Mary's staff **include** Jones,Smith and Murphy* > *Mary's staff **includes** Jones,Smith and Murphy*
- Context-Sensitive Spelling Errors, 10%
  *And **then**?* > *And **them**?*
- Extra Word Errors, 5%
  in defiance of the free rider prediction > *in defiance of the free rider **near** prediction*
- Missing Word Errors, 13%
  *She steered **Melissa** round a corner* > *She steered round a corner*

The occurrence of these *covert errors* [James, 1998] can be reduced by fine-tuning the error creation procedure but they can never be completely eliminated. Indeed, they occur even in manually created error corpora, containing real errors.

3.2 Gold Standard Transformation

The gold standard transformation procedure takes an ungrammatical sentence and a gold standard syntactic analysis of the grammatical sentence from which the ungrammatical one has been generated, and outputs a gold standard syntactic analysis of the ungrammatical sentence. The transformation method is based on three assumptions, the third assumption following on from the first two:

1. At the heart of every ungrammatical sentence, there is a grammatical sentence which expresses the same "intended" meaning as the ungrammatical sentence.
2. The role of a parser is to produce an analysis for a sentence which reflects that sentence's "intended" meaning.
3. A parser which aims to be robust to errors should produce an analysis for an ungrammatical sentence which is as close as possible to the analysis it produces for the corresponding grammatical sentence.

In keeping with these assumptions, the transformation procedure operates by changing as little as possible in the original grammatical sentence analysis to produce the analysis of the ungrammatical sentence. Examples are provided for the error types described in Section 3.1. For each example, a phrase-structure analysis *and* a dependency analysis is shown. Both types of analysis are shown to emphasize that ungrammatical treebanks can be automatically generated from any type of treebank, regardless of the syntactic annotation scheme it employs.

Consider the grammatical sentence (1) and the ungrammatical sentence (2):

(1) *A romance **is** coming your way.*

(2) *A romance **in** coming your way.*

Fig. 1 depicts a Penn-Treebank-style gold standard parse tree[3] for the grammatical sentence (1) and, underneath it, the parse tree which will be produced by the transformation procedure for the ungrammatical sentence (2). This is considered to be the gold standard parse for the ungrammatical sentence because it makes the crucial recognition that the word *in* is part of a verb phrase. A parser which produces this parse is robust to errors since it is able to see right through an ungrammatical sentence to the grammatical sentence at its heart, and produce a parse which reflects the meaning of the grammatical sentence.

Of course, a parse can be represented using a dependency analysis instead of a phrase structure tree. Following [Lin, 1998], a dependency analysis consists of a set of tuples where each tuple represents a word in the sentence and has the form:

(`Word`,`Category`,[`Head`], [`Relationship`]).

Word is the actual word in the sentence, *Category* is its part of speech category, *Head* is another word in the sentence upon which *Word* is dependent. *Relationship* specifies the nature of the dependency relationship between *Word* and *Head*. *Head* and *Relationship* are optional and can be omitted for words in the sentence which are not dependent on any other word. An example is the *head* word of the sentence which is not dependent on any other word and upon which all other words are directly or indirectly dependent. Fig. 2 shows a gold standard dependency analysis for the grammatical (1), and, underneath it, the gold standard analysis for the ungrammatical

[3] Penn-II functional tags and null elements have been omitted, since they are not needed to explain the tree transformations.

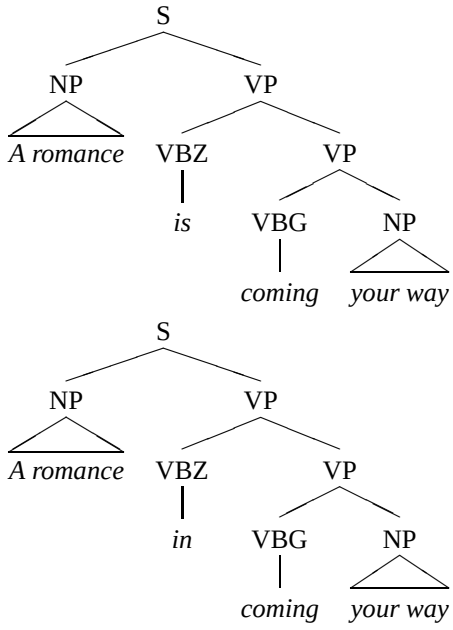


Figure 1: Gold Standard Parse Trees for Sentences (1) and (2)

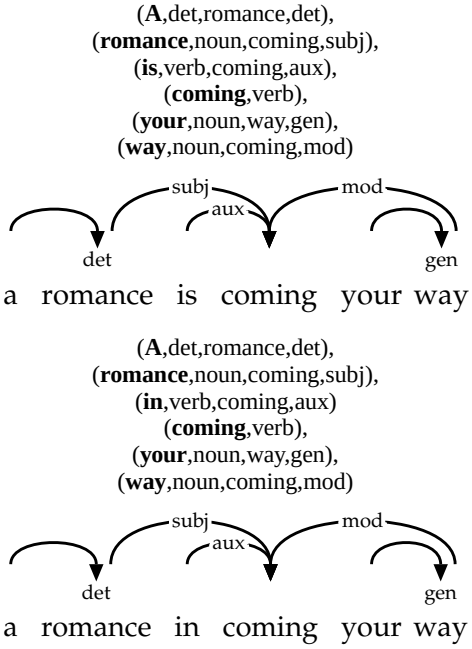


Figure 2: Gold Standard Dependency Analyses for Sentences (1) and (2)

(2). The top analysis specifies that *coming* is the head of the sentence, its subject is headed by *romance* and its modifier is headed by *way*. The bottom analysis also recognizes that the word *in* is dependent on the verb *coming* and the nature of the dependency is an auxiliary verb relationship. The example sentence (2) contains a context-sensitive spelling error but the same transformation would apply to any error correctable by a substitution, e.g. an agreement error.

Consider the grammatical sentence (3) and its ungrammatical counterpart (4):

(3) *Total revenues are expected **to be about** EUR 1 billion.*

(4) *Total revenues are expected **to about** EUR 1 billion.*

A gold standard parse tree for the grammatical (3) is shown in Fig. 3, with the gold standard parse tree which will be automatically generated for the ungrammatical (4) underneath. The bottom tree is produced by replacing the pre-terminal category *(VB be)* in the top tree in Fig. 3 with the trace *(-T- 0)*. A gold standard dependency analysis of the grammatical (3) is shown in Fig. 4, with the gold standard analysis for the ungrammatical (4) underneath. In this analysis, "()" is used to indicate a non-overt element in the sentence. This analysis should be considered to be accurate since it captures all and only the dependencies present in the gold standard analysis of the grammatical sentence.

As a final example, consider the grammatical sentence (5) and the ungrammatical sentence (6):

(5) *Annotators parse the sentences in a corpus.*

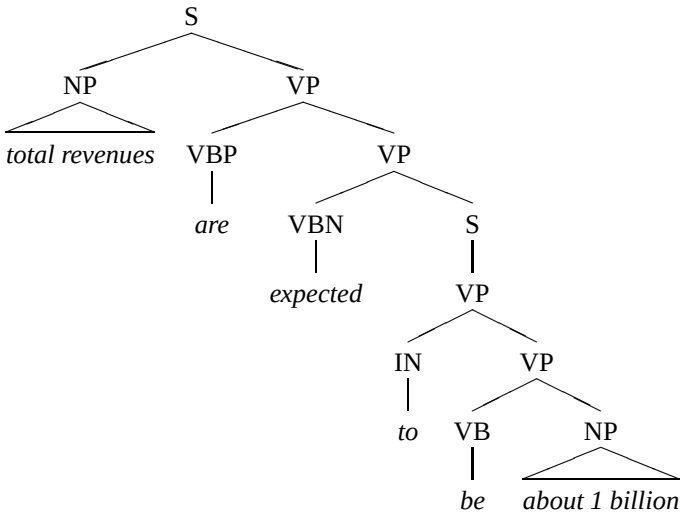
(6) *Annotators parse **to** the sentences in a corpus.*

Fig. 5 shows the gold standard parse tree for the grammatical (5), along with the *three* gold standard parse trees which will be generated automatically by the transformation procedure for the ungrammatical (6). In the ungrammatical gold standard trees, the superfluous *to* does not affect the constituent structure of the sentence (above the pre-terminal level). The only difference between the three trees is the level where the word *to* is attached. In all three, *to* has not introduced any extra structure, which is a desirable result since the word does not contribute to the sentence's meaning. A gold standard dependency parse for the grammatical (5) is shown in Fig. 6, and a gold standard dependency analysis for the ungrammatical (6) is shown underneath. In the ungrammatical analysis, *to* is not linked to the other words in the sentence since it is not dependent on any of them and none are dependent on it. Thus, this analysis preserves all the dependencies present in the grammatical analysis and introduces no others.

4 A Parser Evaluation Experiment using the Penn Treebank

In this section, the usefulness of an automatically created ungrammatical treebank is demonstrated by describing a small parser evaluation experiment which was carried out using an ungrammatical version of section 23 of the Wall Street Journal portion of the Penn Treebank [Marcus *et al.*, 1993; 1994]. The aim of this experiment is to evaluate how well a popular lexicalized generative statistical parser copes with errors in text: a parser that copes well with errors produces, for an ungrammatical sentence, an analysis which closely resembles the analysis it would produce for the sentence without the error.

Section 4.1 contains a description of how the experiment was carried out and Section 4.2 presents the results, which are then discussed briefly in Section 4.3.



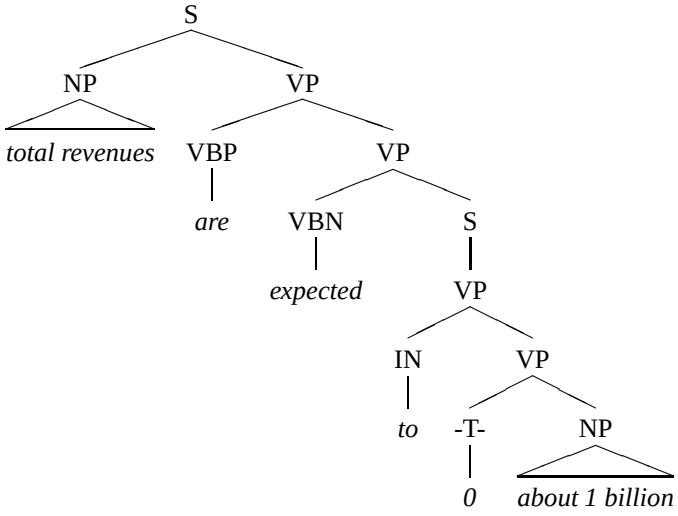
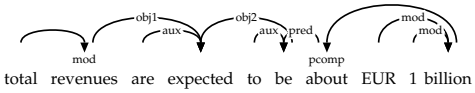


Figure 3: Gold Standard Parse Trees for Sentences (3) and (4)

(**Total**,adjective,revenues,mod),
(**revenues**,noun,expected,obj1),
(**are**,verb,expected,aux),
(**expected**,verb),
(**to**,infmarker,be,aux),
(**be**,verb,expected,obj2),
(**about**,prep,be,pred),
(**EUR**,noun,billion,mod),
(**1**,noun,billion,mod),
(**billion**,noun,about,pcomp)



(**Total**,adjective,revenues,mod),
(**revenues**,noun,expected,obj1),
(**are**,verb,expected,aux),
(**expected**,verb),
(**to**,infmarker,be,aux),
(**()**,verb,expected,obj2),
(**about**,prep,(),pred),
(**EUR**,noun,billion,mod),
(**1**,noun,billion,mod),
(**billion**,noun,about,pcomp)

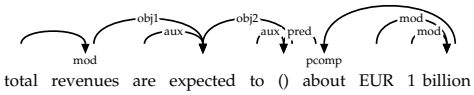


Figure 4: Gold Standard Dependency Analyses for Sentences (3) and (4)

4.1 Method

The error creation procedure described in Section 3.1 was applied to the 2330 sentences in Section 23 of the WSJ portion of the Penn Treebank [Marcus *et al.*, 1993; 1994], resulting in an error corpus of 8550 sentences (1704 sentences containing an agreement error, 2214 sentences containing a context sensitive spelling error, 2304 sentences containing an extra word and 2328 sentences with a missing word). The gold standard transformation procedure described in Section 3.2 was then applied, resulting in an ungrammatical version of Section 23.

The generative lexicalized statistical parser described in [Bikel, 2004], trained on the original grammatical Sections 2-21 of the WSJ, was used to parse the ungrammatical sentences. The input to Bikel's parser was untagged. These parses were evaluated against the ungrammatical gold standard WSJ23 parses using the Parseval [Black *et al.*, 1991] labelled precision/recall measures. In the case of extra word errors, there is potentially more than one gold standard analysis for each sentence, and therefore the test sentence parse is evaluated against each of its gold standard parses, and the highest f-score is chosen.

4.2 Results

The table in Fig. 7 shows labelled precision, recall and f-score results calculated by evaluating Bikel's parser against the ungrammatical WSJ23 using the Parseval measures. The table in Fig. 8 shows other interesting statistics: the percentage of analyses in the test sentence set which completely match the

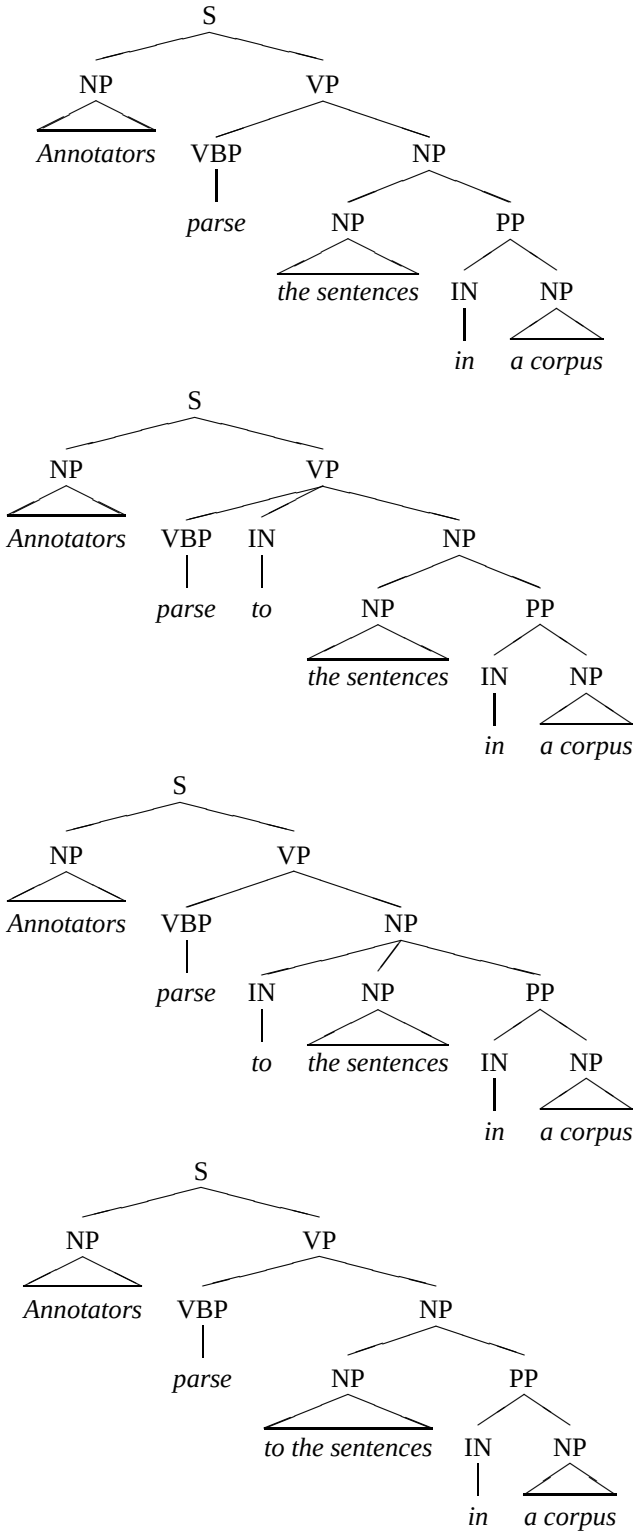


Figure 5: Gold Standard Parse Trees for Sentences (5) and (6)

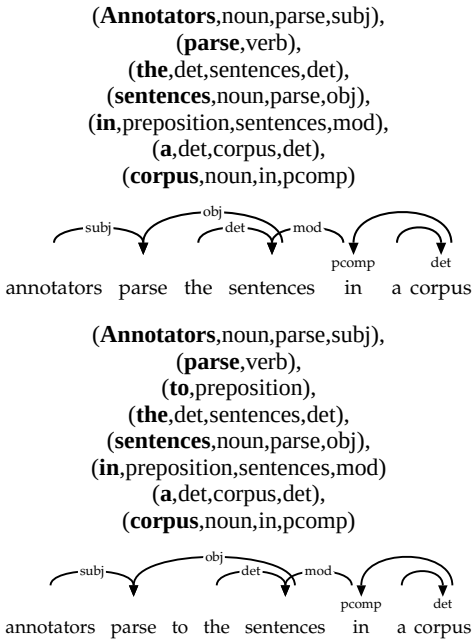


Figure 6: Gold Standard Dependency Analyses for Sentences (5) and (6)

Error Type	Precision	Recall	F-Score
No Error	84.9	84.8	84.9
Agreement	83.7	83.3	83.5
Context-Sensitive Spelling	79.0	78.6	78.8
Extra Word	79.4	82.5	80.9
Missing Word	81.5	77.7	79.6

Figure 7: Bikel Parser on Ungrammatical WSJ23

corresponding gold standard analyses, and the percentage of analyses in the test sentence set which achieve a relatively low f-score of under 75%.

The first row in Figs. 7 and 8 indicate the scores received by the parser on the original Section 23 WSJ sentences.[4] The first row figures represent an upper bound for the ungrammatical sentence results, since the grammatical and ungrammatical gold standard trees are isomorphic above the pre-terminal level and pre-terminal constituents are ignored in calculation of precision and recall.

4.3 Discussion

The results in Figs. 7 and 8 show that ungrammatical sentences containing agreement errors achieve scores which are the closest to the upper bound, suggesting that this type of error does not generally distract this parser from finding the

[4] A higher f-score of 87.5% can be achieved by ignoring punctuation in the evaluation. However, in this evaluation, punctuation is not ignored because the error creation procedure treats all tokens including punctuation symbols as candidates for errors, e.g. an extra word error can be created by inserting an unnecessary punctuation symbol.

Error Type	100% Match	Low Scoring
No Error	24.6	21.4
Agreement	19.7	24.0
Context-Sensitive Spelling	11.5	36.3
Extra Word	10.0	31.2
Missing Word	9.0	35.6

Figure 8: Bikel Parser on Ungrammatical WSJ23

*(S (NP American) (VP (VP **is/are** (VP preparing (S (VP to (VP take ...))))) and (VP is n't (VP anticipating ....))))*

*(S (NP American) (VP (VP **is** (VP preparing (S (VP to (VP take ...))))) and (VP is n't (VP anticipating ....))))*

*(S (NP American) (VP **are** (VP preparing (S (VP to (VP (VP take ...) and (VP is n't (VP anticipating ....))))))))*

Figure 9: Low scoring parse due to agreement error

correct analysis. However, there are cases, such as the example shown in Fig. 9, where the presence of an agreement error does cause the parser to perform worse on the ungrammatical sentence than on its grammatical counterpart. In Fig. 9, the first parse is the gold standard analysis of the grammatical and ungrammatical sentences, the second parse is the parse produced by Bikel's parser for the grammatical sentence and the third parse is the parse produced by Bikel's parser for the ungrammatical sentence.

The worst-performing error type is the context-sensitive spelling error. It is not surprising that this error type performs the worst, since it often involves a part-of-speech change of one of the words in the sentence. Fig. 10 shows three parses: the first, topmost parse is the gold standard parse for the original grammatical WSJ sentence and the ungrammatical sentence derived from it, the second parse is the parse produced by Bikel's parser for the grammatical sentence, and the bottom parse is the parse produced by the same parser for the ungrammatical sentence.

Extra word errors achieve a higher recall score in comparison to their precision score which suggests that this kind of error tends to introduce unwanted structure into a parse. Similarly, missing word errors achieve a higher precision score in comparison to their recall score, suggesting that a lack of relevant structure is associated with this kind of error. This is expected. An example of an extra word error causing a misparse is shown in Fig. 11. The first parse is the gold standard parse for the grammatical sentence, the next three parses are the gold standard parses for the ungrammatical sentence, the fifth parse is the parse produced by Bikel's parser for the grammatical sentence and the sixth parse is the parse produced by Bikel's parser for the ungrammatical sentence. A similar example is shown in Fig. 12 for a missing word error: the first two parses are the gold standard parses for the grammatical and ungrammatical sentences, respectively, and the last two parses are the parses produced by Bikel's parser for the grammatical and ungrammatical sentences, respectively.

*(S (ADVP Just) (VP thought (SBAR (S (NP **you/your**) (VP 'd (VP like (S (VP to (VP know))))))))*

*(S (NP Just) (VP thought (SBAR (S (NP **you**) (VP 'd (VP like (S (VP to (VP know))))))))*

*(SINV (ADVP Just) (VP thought (S (ADJP **your**))) (VP 'd (VP like (S (VP to (VP know))))))*

Figure 10: Low scoring parse due to context-sensitive spelling error

(S (NP Ports...) (VP reached (NP agreements (S (VP to (VP sell) (NP its remaining seven aircraft) (PP to (NP (NP buyers) (SBAR (WHNP that) (S (VP were n't (VP disclosed))))))))))

*(S (NP Ports...) (VP reached (NP agreements (S (VP to (VP sell) (NP its remaining seven aircraft) (PP to (NP (NP buyers) (SBAR (WHNP that) (S (VP **said** were n't (VP disclosed))))))))))*

*(S (NP Ports...) (VP reached (NP agreements (S (VP to (VP sell) (NP its remaining seven aircraft) (PP to (NP (NP buyers) (SBAR (WHNP that) (S **said** (VP were n't (VP disclosed))))))))))*

*(S (NP Ports...) (VP reached (NP agreements (S (VP to (VP sell) (NP its remaining seven aircraft) (PP to (NP (NP buyers) (SBAR (WHNP that **said)** (S (VP were n't (VP disclosed))))))))))*

(S (NP Ports...) (VP reached (NP (NP agreements) (S (VP to (VP sell (NP its remaining seven aircraft) (PP to (NP (NP buyers) (SBAR (WHNP that) (S (VP were n't (VP disclosed))))))))))

*(S (NP Ports...) (VP reached (NP (NP agreements) (S (VP to (VP sell (NP its remaining seven aircraft) (PP to (NP (NP buyers) (SBAR (WHNP that) (S (VP **said**))))))))) (VP were n't (VP disclosed)))*

Figure 11: Low scoring parse due to extra word error

*(S (NP Several fund managers) (VP expect (NP (NP a rough market) (NP this morning)) (SBAR **before** (S (NP prices) (VP stabilize)))))*

(S (NP Several fund managers) (VP expect (NP (NP a rough market) (NP this morning)) (SBAR (S (NP prices) (VP stabilize)))))

*(S (NP Several fund managers) (VP expect (NP a rough market) (NP this morning) (SBAR **before** (S (NP prices) (VP stabilize)))))*

(S (NP Several fund managers) (VP expect (NP (NP a rough market) (SBAR (S (NP this morning prices) (VP stabilize))))))

Figure 12: Low scoring parse due to missing word error

5 Future Work

This paper has introduced the concept of a treebank of ungrammatical sentences, explained how one can be automatically derived from any treebank, and then described an experiment which evaluates a statistical parser [Bikel, 2004] on an ungrammatical version of Section 23 of the Wall Street Journal.

The experiment described in Section 4 is only a starting point to illustrate a use of an ungrammatical treebank in the area of parser robustness evaluation. It is clear that the results in Fig. 7 need to be analysed so that, within each error type, the problematic ungrammatical constructions can be identified. The performance of other parsers on the ungrammatical WSJ23 could also be tested. Another obvious use of an ungrammatical treebank would be to improve a parser's performance on ungrammatical sentences. For example, Bikel's parser could be re-trained on an ungrammatical version of WSJ2-21 and then evaluated against the ungrammatical WSJ23 (as in Fig. 7) *and* against naturally occurring ungrammatical errors. It would be interesting to see how a parser behaves on well-formed and ill-formed text when trained on a grammatical *and* ungrammatical treebank. Ideally, the induced probabilistic grammar could be partitioned in such a way that the parser not only correctly parses an ungrammatical sentence, but also recognizes that the sentence is ungrammatical, and locates the error. Finally, there is room for improvement in the error creation procedure – it could be extended to introduce less common errors.

Acknowledgments

Thank you to Joachim Wagner and Josef van Genabith for their helpful comments on this paper. I am also grateful to the IRCSET Embark Initiative Postdoctoral Fellowship Award Scheme for supporting this research.

References

[Becker *et al.*, 1999] Markus Becker, Andrew Bredenkamp, Berthold Crysmann, and Judith Klein. Annotation of error types for german news corpus. In *Proceedings of the ATALA Workshop on Treebanks*, Paris, France, 1999.

[Bigert, 2004] Johnny Bigert. Probabilistic detection of context-sensitive spelling errors. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-04)*, volume Five, pages 1633–1636, Lisbon, Portugal, 2004.

[Bikel, 2004] Dan Bikel. *On the Parameter Space of Generative Lexicalized Parsing Models*. PhD thesis, University of Pennslyvania, 2004.

[Black *et al.*, 1991] E. Black, Steve Abney, Dan Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, Fred Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the 1991 DARPA Speech and Natural Language Workshop*, pages 306–311, 1991.

[Burnard, 2000] L. Burnard. User reference guide for the british national corpus. Technical report, Oxford University Computing Services, 2000.

[Emi *et al.*, 2004] Izumi Emi, Kiyotaka Uchimoto, and Hitoshi Isahara. The overview of the sst speech corpus of japanese learner english and evaluation through the experiment on automatic detection of learners' errors. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-04)*, volume Four, pages 1435–1439, Lisbon, Portugal, 2004.

[Foster, 2005] Jennifer Foster. *Good Reasons for Noting Bad Grammar: Empirical Investigations into the Parsing of Ungrammatical Written English*. PhD thesis, University of Dublin, Trinity College, 2005.

[Granger, 1993] Sylviane Granger. International corpus of learner english. In J. Aarts, P. de Haan, and N.Oostdijk, editors, *English Language Corpora: Design, Analysis and Exploitation*, pages 57–71. Rodopi, Amsterdam, 1993.

[Ingels, 1996] Peter Ingels. *A Robust Text Processing Technique Applied to Lexical Error Recovery*. PhD thesis, Linkoping University, Sweden, 1996.

[James, 1998] Carl James. *Errors in Language Learning and Use: Exploring Error Analysis*. Addison Wesley Longman, 1998.

[Kepser *et al.*, 2004] Stephan Kepser, Ilona Steiner, and Wolfgang Sternefeld. Annotating and querying a treebank of suboptimal structures. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT2004)*, pages 63–74, Tuebingen, Germany, December 2004.

[Lin, 1998] Dekang Lin. Dependency-based evaluation of minipar. In *Proceedings of Workshop on The Evaluation of Parsing Systems*, pages 48–56, 1998.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[Marcus *et al.*, 1994] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *Proceedings of the 1994 ARPA Speech and Natural Language Workshop*, pages 114–119, Princeton, New Jersey, 1994.