The Syntactically Annotated ICE Corpus and the Automatic Induction of a Formal Grammar

Alex Chengyu Fang
Department of Chinese, Translation and Linguistics
City University of Hong Kong
acfang@cityu.edu.hk

Abstract

The International Corpus of English is a corpus of national and regional varieties of English. The mega-word British component has been constructed, grammatically tagged, and syntactically parsed. This article is a description of work that aims at the automatic induction of a wide-coverage grammar from this corpus as well as an empirical evaluation of the grammar. It first of all describes the corpus and its annotation schemes and then presents empirical statistics for the grammar. I will then evaluate the coverage and the accuracy of such a grammar when applied automatically in a parsing system. Results show that the grammar enabled the parser to achieve 86.1% recall rate and 83.5% precision rate.

1 Introduction

The International Corpus of English (ICE) is a project that aims at the construction of a collection of corpora of English in countries and regions where English is used either as a first or as an official language (Greenbaum 1992). Each component corpus comprises one million words of both written and transcribed spoken samples that are then annotated at grammatical and syntactic levels. The British component of the ICE corpus was used to automatically induce a large formal grammar, which was subsequently used in a robust parsing system. In what follows, this article will first of all describe the annotation schemes for the corpus and the evaluation of a formal grammar automatically induced from the corpus in terms of its potential coverage when tested with empirical data. Finally, this article will present an evaluation of the grammar through its application in a robust parsing system in terms of labelling and bracketing accuracies.

1.1 The ICE wordclass annotation scheme

There are altogether 22 head tags and 71 features in the ICE wordclass tagging scheme, resulting in about 270 grammatically possible combinations. Compared with 134 tags for LOB, 61 for BNC, and 36 for Penn Treebank, the ICE tagset is perhaps the most detailed in automatic applications. They cover all the major English word classes and provide morphological, grammatical, collocational, and sometimes syntactic information. A typical ICE tag has two components: the head tag and its features that bring out the grammatical features of the associated word. For instance, **`N(com,sing)`** indicates that the lexical item associated with this tag is a common (**`com`**) singular (**`sing`**) noun (**`N`**).

Tags that indicate phrasal collocations include **`PREP(phras)`** and **`ADV(phras)`**, prepositions (as in [1]) and adverbs (as in [2]) that are frequently used in collocation with certain verbs and adjectives:

[1] *Thus the dogs' behaviour had been changed because they associated the bell <u>with</u> the food.*

[2] *I had been filming The Paras at the time, and Brian had had to come <u>down</u> to Wales with the records.*

Some tags, such as **`PROFM(so,cl)`** (pronominal *so* representing a clause as in [3]) and **`PRTCL(with)`** (particle *with* as in [4]), indicate the presence of a clause; *so* in [3] signals an abbreviated clause while *with* in [4] a non-finite clause:

[3] *If so, I'll come and meet you at the station.*

[4] *The number by the arrows represents the order of the pathway causing emotion, with the cortex lastly having the emotion.*

Examples [5]-[7] illustrate tags that note special sentence structures. *There* in [5] is tagged as **EXTHERE,** existential *there* that indicates a marked sentence order. [6] is an example of the cleft sentence (which explicitly marks the focus), where *it* is tagged as **CLEFTIT**. [7] exemplifies anticipatory *it,* which is tagged as **ANTIT**:

[5] *There were two reasons for the secrecy.*

[6] *It is from this point onwards that Roman Britain ceases to exist and the history of sub-Roman Britain begins.*

[7] *Before trying to answer the question it is worthwhile highlighting briefly some of the differences between current historians.*

The verb class is divided into auxiliaries and lexical verbs. The auxiliary class notes modals, perfect auxiliaries, passive auxiliaries, semi-auxiliaries, and semip-auxiliaries (those followed by *-ing* verbs). The lexical verbs are further annotated according to their complementation types. There are altogether seven types: complex transitive, complex ditransitive, copular, dimono-transitive, ditransitive, intransitive, mono-transitive, and **TRANS**. Figure 1 shows the sub-categorisations of the verb class.

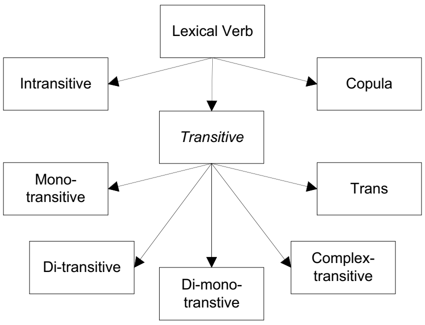


Figure 1: The ICE subcategorisation for verbs

The notation **TRANS** of the transitive verb class is used in the ICE project to tag those transitive verbs followed by a noun phrase that may be the subject of the following non-finite clause. This type of verb can be analysed differently according to various tests into, for instance, monotransitives, ditransitives and complex transitives. To avoid arbitrary decisions, the complementing non-finite clause is assigned a catch-all term 'transitive complement' in parsing, and its preceding verb is accordingly tagged as **TRANS** in order to avoid making a decision on its transitivity type. This verb type is best demonstrated by [8]-[11]:

[8] *Just before Christmas, the producer of Going Places, Irene Mallis, had asked me to make a documentary on 'warm-up men'.*

[9] *They make others feel guilty and isolate them.*

[10] *I can buy batteries for the tape - but I can see myself spending a fortune!*

[11] *The person who booked me in had his eyebrows shaved and replaced by straight black painted lines and he had earrings, not only in his ears but through his nose and lip!*

In examples [8]-[11], *asked, make, see,* and *had* are all complemented by non-finite clauses with overt subjects, the main verbs of these non-finite clauses being infinitive, present participle and past participle.

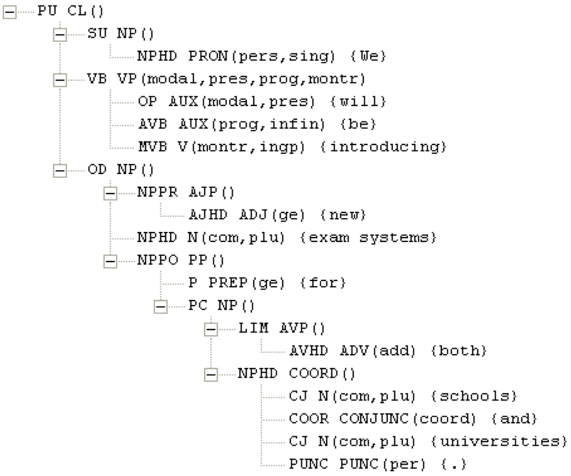
As illustrated by examples [1]-[11], the ICE tagging scheme has indeed gone beyond the wordclass to provide some syntactic information and has thus proved itself to be an expressive and powerful means of pre-processing for subsequent parsing.

1.2 The ICE parsing scheme

The ICE parsing scheme recognises five basic syntactic phrases. They are adjective phrase (**AJP**), adverb phrase (**AVP**), noun phrase (**NP**), prepositional phrase (**PP**), and verb phrase (**VP**). Each tree in the ICE parsing scheme is re-presented as a functionally labelled hierarchy, with features describing the characteristics of each constituent, which is represented as a pair of function-category labels. In the case of a terminal node, the function-category descriptive labels are appended by the lexical item itself in curly brackets. Figure 2 is such a structure for [12].

[12] *We will be introducing new exam systems for both schools and universities.*

According to Figure 2, we know that [12] is a parsing unit (**PU**) realised by a clause (**CL**), which governs three daughter nodes: **SU NP** (NP as subject), **VB VP** (VP as verbal), and **OD NP** (NP as direct object). Each of the three daughter nodes are sub-branched until the leaves nodes with the input tokens in curly brackets. The direct object node, for example, has three immediate constituents: **NPPR AJP** (AJP as NP pre-modifier), **NPHD N(com,plu)** (plural common noun as the NP head), and **NPPO PP** (PP as NP post-modifier).



```
PU CL()
  SU NP()
    NPHD PRON(pers,sing) {We}
  VB VP(modal,pres,prog,montr)
    OP AUX(modal,pres) {will}
    AVB AUX(prog,infin) {be}
    MVB V(montr,ingp) {introducing}
  OD NP()
    NPPR AJP()
      AJHD ADJ(ge) {new}
    NPHD N(com,plu) {exam systems}
    NPPO PP()
      P PREP(ge) {for}
      PC NP()
        LIM AVP()
          AVHD ADV(add) {both}
        NPHD COORD()
          CJ N(com,plu) {schools}
          COOR CONJUNC(coord) {and}
          CJ N(com,plu) {universities}
          PUNC PUNC(per) {.}
```

Figure 2: A parse tree for [12]

Note that in the same example, the head of the complementing NP of the prepositional phrase is initially analysed as a coordinated construct (**COORD**), with two plural nouns as the conjoins (**CJ**) and a coordinating conjunction as co-ordinator (**COOR**).

In all, there are 58 non-terminal parsing symbols in the ICE parsing scheme, compared with 20 defined in the Penn Treebank project. The Suzanne Treebank has 43 function/category symbols, discounting those that are represented as features in the ICE system.

2 The generation of a formal grammar

The British component of the ICE corpus, annotated in fashions described above, has been used to automatically generate a formal grammar that has been subsequently applied in an automatic parsing system to annotate the rest of the corpus (Fang 1995, 1996, 1999). The grammar consists of two sets of rules. The first set describes the five canonical phrases (AJP, AVP, NP, PP, VP) as sequences of grammatical tags terminating at the head of the phrase. For example, the sequence **AUX(modal,pres) AUX(prog,infin) V(montr,ingp)** is a VP rule describing instantiations such as *will be introducing* in [12]. The second set describes the clause as sequences of phrase types. The string in [12], for instance, is described by a sequence **NP VP NP PP** in the set of clausal rules.

To empirically characterise the grammar, the syntactically parsed ICE corpus was divided into ten equal parts according to the number of component texts. One part was set aside for testing, which was further divided into five test sets. The remaining nine parts were used as training data in a leave-one-out fashion. In this way, the training data was used to generate 9 consecutive training sets, each increased by one part over the previous set, with Set 1 formed of one training set, Set 2 two training sets, and Set 3 three training sets, etc. The evaluation thus not only aims to establish the potential coverage of the grammar but also to indicate the function between the coverage of the grammar and the training data size.

Figure 3 shows the growth of the number of phrase structure rules as a function of the growth of training data size. The Y-axis indicates the number of rules generated from the training data and the X-axis the gradual increase of the training data size.

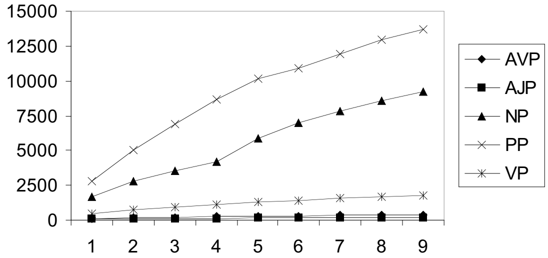


Figure 3: The number of phrase structure rules as a function of growing training data size

It can be observed that AJP and AVP show only a marginal increase in the number of different rules with the increase of training data size, therefore demonstrating a relatively small core set. In comparison, VPs are more varied but still exhibit a visible plateau of growth. The other two phrases, NP and PP, show a much more varied set of rules not only through their large numbers (9,184 for NPs and 13,736 for PPs) but also the sharp learning curve. There are many reasons for the potentially large set of rules for PPs since they structurally subsume the clause as well as all the phrase types. Their large number is therefore more or less expected. The large set of NP rules is however a bit surprising since they are often characterised, perhaps too simplistically, as comprising a determiner group, a premodifier group, and the noun head but the grammar has 9,184 different rules for this phrase type. While this phenomenon calls for further investigations, we are concerned with only the coverage issue for the moment in the current article.

3 The coverage of the formal grammar

The coverage of the formal grammar is evaluated through individual rule sets for the five canonical phrase types separately. The coverage by the clausal rules will also be reported towards the end of this section.

3.1 The coverage of AJP rules

As Figure 4 suggests, the coverage of the grammar, when tested with the five samples, is consistently high – all above 99% even when the grammar was trained from only one ninth of the training set. The increase of the size of the training set does not show significant enhancement of the coverage.

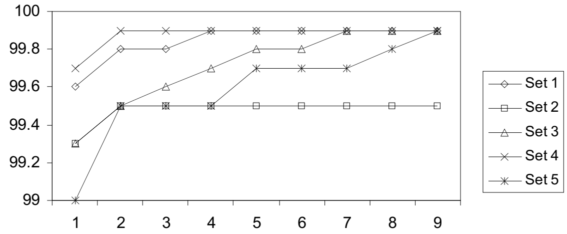


Figure 4: The coverage of AJP rules

3.2 The coverage of AVP rules

Like AJP rules, high coverage can be achieved with a small training set since when trained with only one ninth of the training data, the AVP rules already showed a high coverage of above 99.4% and quickly approaching 100%. See Figure 5.

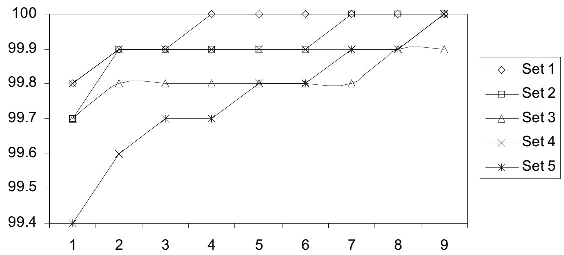


Figure 5: The coverage of AVP rules

3.3 The coverage of NP rules

Although lower than AVP and AJP discussed above, the NP rules show a satisfactorily high coverage when tested by the five samples. As can be seen from Figure 6, the initial coverage when trained with one ninth of the training data is generally above 97%, rising proportionally as the training data size increases, to about 99%. This seems to suggest a mildly complex nature of NP structures.

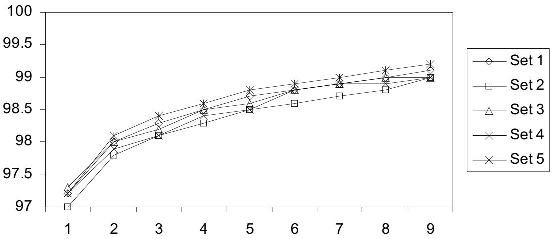


Figure 6: The coverage of NP rules

3.4 The coverage of VP rules

VPs do not seem to pose significant challenge to the parser. As Figure 7 indicates, the initial coverage is all satisfactorily above 97.5%. Set 1 even achieved a coverage of over 98.5% when the grammar was trained with only one ninth of the training data. As the graph seems to suggest, the learning curve arrives at a plateau when trained with about half of the total training data, suggesting a centralised use of the rules.

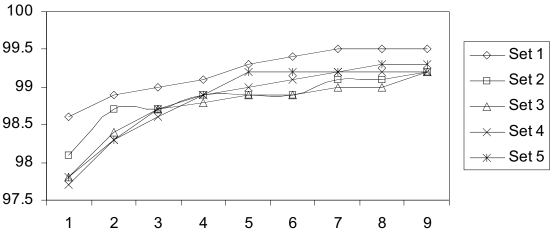


Figure 7: The coverage of VP rules

3.5 The coverage of PP rules

As is obvious from Figure 8, PPs are perhaps the most complex of the five phrases with an initial coverage of just over 70%. The learning curve is sharp, culminating between 85% and 90% with the full training data set. As far as parser construction is concerned, this phrase alone deserves special attention since it explains much of the structural complexity of the clause. Based on this observation, a separate study was carried out to automatically identify the syntactic functions of PPs.

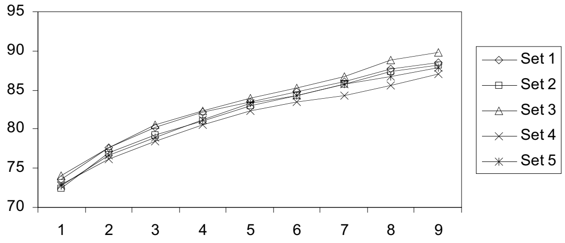


Figure 8: The coverage of PP rules

3.6 The coverage of clausal rules

Clausal rules present the most challenging problem since, as Figure 9 clearly indicates, their coverage is all under 67% even when trained with all of the training data. This observation seems to reaffirm the usefulness of rules at phrase level but the inadequacy of clause structural rules. Indeed, it is intuitively clear that the complexity of the sentence is mainly the result of the combination of clauses of various kinds.

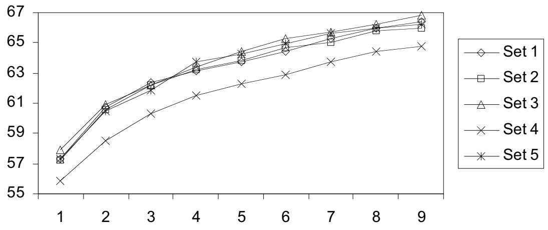


Figure 9: The coverage of CL rules

3.7 Discussion

This section presented an evaluation of the grammar in terms of its coverage as a function of growing training data size. As is shown, the parsed corpus resulted in excellent grammar sets for the canonical phrases, AJP, AVP, NP, PP, and VP: except for PPs, all the phrase structure rules achieved a wide coverage of about 99%. The more varied set for PPs demonstrated a coverage of nearly 90%, not as high as what is achieved for the other phrases but still highly satisfactory.

The coverage of clause structure rules, on the other hand, showed a considerably poorer performance compared with the phrases. When all of the training data was used, these rules covered just over 65% of the testing data.

In view of these empirical observations, it can be reliably concluded that the corpus-based grammar construction holds a promising approach in that the phrase structure rules generally have a high coverage when tested with unseen data. The same approach has also raised two questions at this stage: Does the high-coverage grammar also demonstrate a high precision of analysis? Is it possible to enhance the coverage of the clause structure rules within the current framework?

4 Evaluating the accuracy of analysis

The ICE project used two major annotation tools: AUTASYS and the Survey Parser. AUTASYS is an automatic wordclass tagging system that applies the ICE tags to words in the input text with an accuracy rate of about 94% (Fang 1996a). The tagged text is then fed into the Survey Parser for automated syntactic analysis. The parsing model is one that tries to identify an analogy between the input string and a sentence is that already syntactically analysed and stored in a database (Fang 1996b and 2000). This parser is driven by the previously described formal grammar for both phrasal and clausal analysis. In this section, the formal grammar is characterised through an empirical evaluation of the accuracy of analysis by the Survey Parser.

4.1 The NIST evaluation scheme

The National Institute of Science and Technology (NIST) proposed an evaluation scheme that looks at the following properties when comparing recognition results with the correct answer:

$$\text{Correct Match Rate} = \frac{\text{number of correct constituents in } T_P}{\text{number of constituents in } T_C}$$

$$\text{Substitution Rate} = \frac{\text{number of substituted constituents in } T_P}{\text{number of constituents in } T_C}$$

$$\text{Deletion Rate} = \frac{\text{number of deleted constituents in } T_P}{\text{number of constituents in } T_C}$$

$$\text{Insertion} = \text{number of inserted constituents in } T_P$$

$$\text{Combined Rate} = \frac{\text{number of correct constituents in } T_P - \text{number of insertions}}{\text{number of constituents in } T_C}$$

Notably, the correct match rate is identical to the labelled or bracketed recall rate. The commonly used precision score is calculated as the total number of correct nodes over the sum of correct, substituted, and inserted nodes. The insertion score, arguably, subsumes crossing brackets errors since crossing brackets errors are caused by the insertion of constituents even though not every insertion causes an instance of crossing brackets violation by definition. In this respect, the crossing brackets score only implicitly hints at the insertion problem while the insertion rate of the NIST scheme explicitly addresses this issue.

Because of the considerations above, the evaluations to be reported in the next section were conducted using the NIST scheme. To objectively present the two sides of the same coin, the NIST scheme was used to evaluate the Survey Parser in terms of constituent labelling and constituent bracketing before the two are finally combined to yield performance scores.

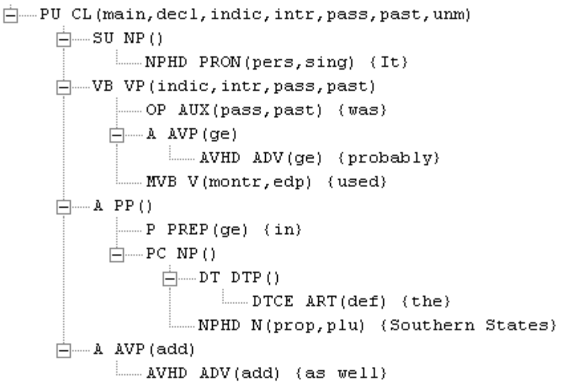
In order to conduct a precise evaluation of the performance of the parser, the experiments look at the two aspects of the parse tree: labelling accuracy and bracketing accuracy. Labelling accuracy expresses how many correctly labelled constituents there are per hundred constituents and is intended to measure how well the parser labels the constituents when compared to the correct tree. Bracketing accuracy attempts to measure the similarity of the parser tree to that of the correct one by expressing how many correctly bracketed constituents there are per hundred constituents. In this section, the NIST metric scheme will be applied to the two properties separately before an attempt is made to combine the two to assess the overall performance of the Survey Parser.

The same set of test data described in the previous section was used to create four test sets of 1000 trees each to evaluate the performance of the grammar induced from the training sets described earlier.

4.2 Labelling Accuracy

To evaluate labelling accuracy with the NIST scheme, the method is to view the labelled constituents as a linear string with attachment bracketing removed. For [12], as an example, Figure 10 is a correct tree and Figure 11 is a parser-produced tree.

[12] It was probably used in the Southern States as well.

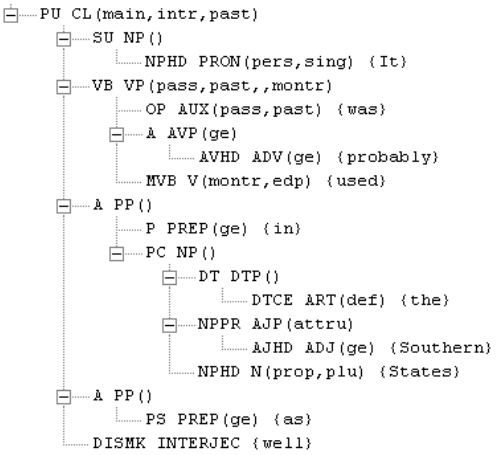


```
PU CL(main,decl,indic,intr,pass,past,unm)
  SU NP()
    NPHD PRON(pers,sing) {It}
  VB VP(indic,intr,pass,past)
    OP AUX(pass,past) {was}
    A AVP(ge)
      AVHD ADV(ge) {probably}
    MVB V(montr,edp) {used}
  A PP()
    P PREP(ge) {in}
    PC NP()
      DT DTP()
        DTCE ART(def) {the}
      NPHD N(prop,plu) {Southern States}
  A AVP(add)
    AVHD ADV(add) {as well}
```

Figure 10: A correct tree for [12]

After removing the bracketed structure, we then have two flattened sequences of constituent labels and compare them using the NIST scheme, which will yield the following statistics:

Total # sentences evaluated :	1
Total # constituent labels :	42
Total # correct matches :	37 (88.1%)
Total # labels substituted :	5 (11.9%)
Total # labels deleted :	0 (0.0%)
Total # labels inserted :	6
Overall labelling accuracy :	73.8%



```
PU CL(main,intr,past)
  SU NP()
    NPHD PRON(pers,sing) {It}
  VB VP(pass,past,,montr)
    OP AUX(pass,past) {was}
    A AVP(ge)
      AVHD ADV(ge) {probably}
    MVB V(montr,edp) {used}
  A PP()
    P PREP(ge) {in}
    PC NP()
      DT DTP()
        DTCE ART(def) {the}
      NPPR AJP(attru)
        AJHD ADJ(ge) {Southern}
      NPHD N(prop,plu) {States}
  A PP()
    PS PREP(ge) {as}
  DISMK INTERJEC {well}
```

Figure 11: A parser-produced tree for [12]

Accordingly, we may concretely claim that there are 42 constituent labels according to the correct tree, of which 37 (88.1%) are correctly labelled by the parser, with 5 substitutions (11.9%), 0 deletion, and 6 insertions. The overall labelling accuracy is then calculated as 73.8%.

A total of 4,000 trees, divided into four sets of 1,000 each, were selected from the test data to evaluate the labelling accuracy of the parser. Empirical results show that the parser achieved an overall labelling precision of over 80%.

	Test Set 1		Test Set 2		Test Set 3		Test Set 4	
	#	%	#	%	#	%	#	%
Tree	1000		1000		1000		1000	
Node	31676		34095		31563		30140	
Correct	27329	86.3	29263	85.8	27224	86.3	26048	86.4
Subs	3214	10.1	3630	10.6	3253	10.3	3084	10.2
Del	1133	3.6	1202	3.5	1086	3.4	1008	3.3
Ins	2021		2316		1923		1839	
Prec.		83.9		83.1		84.1		84.1
Overall		79.9		79.0		80.2		80.3

Table 1: Labelling accuracy

Table 1 shows that the Survey Parser scored 86% or better in terms of correct match (labelled recall) and nearly 84% in terms of labelled precision rate for the four sets. About 10% of the constituent labels are wrong (*Subs*) with a deletion rate (*Del*) of about 3.5%. Counting insertions (*Ins*), the overall labelling accuracy by the parser is around 80%.

4.3 Bracketing Accuracy

A second aspect of the evaluation of the grammar through the use of the Survey Parser involves the measuring of its attachment precision, an attempt to characterise the similarity of the parser-produced hierarchical structure to that of the correct parse tree. To estimate the precision of constituent attachment of a tree, a linear representation of the hierarchical structure of the parse tree is designed which ensures that wrongly attached non-terminal nodes are penalised only once if their sister and daughter nodes are correctly aligned.

Table 2 shows that the parser achieved nearly 86% for the bracketed correct match and 82.8% for bracketing precision. Considering insertions and deletions, the overall accuracy according to the NIST scheme is about 77%. This indicates that for every 100 bracket pairs 77 are correct, with 23 substituted, deleted, or inserted. In other words, for a tree of 100 constituents, 23 edits are needed to conform to the correct tree structure.

	Test Set 1		Test Set 2		Test Set 3		Test Set 4	
	#	%	#	%	#	%	#	%
Tree	1000		1000		1000		1000	
Node	12451		13390		12411		11858	
Correct	10679	85.8	11402	85.2	10620	85.6	10271	86.6
Subs	1088	8.7	1249	9.3	1115	9.0	968	8.2
Del	648	5.5	739	5.5	676	5.4	619	5.2
Ins	1127		1297		1092		1029	
Prec		82.8		81.7		82.8		83.7
Overall		76.7		75.5		76.8		77.9

Table 2: Bracketing accuracy

4.4 Combined accuracy

The combined score for both labelling and bracketing accuracy is achieved through representing both constituent labelling and unlabelled bracketing in a linear string described in the previous sections.

Table 3 gives the total number of trees in the four test sets and the total number of constituents. The number of correct matches, substitutions, insertions and deletions are indicated and combined scores computed accordingly. The table shows that the parser scored 86% and 83.5% respectively for labelled recall and precision. It is also shown that the parser achieved an overall performance of about 79%. Considering that the scoring program tends to underestimate the success rate, it is reasonable to assume a real overall combined performance of 80%.

	Test Set 1		Test Set 2		Test Set 3		Test Set 4	
	#	%	#	%	#	%	#	%
Tree	1000		1000		1000		1000	
Node	44127		47485		43974		41998	
Correct	38008	86.1	40665	85.6	37844	86.1	36319	86.5
Subs	4302	9.7	4879	10.3	4368	9.9	4052	9.6
Del	1781	4.0	1941	4.1	1762	4.0	1627	3.9
Ins	3148		3613		3015		2868	
Prec		83.6		82.7		83.7		83.9
Overall		79.0		78.0		79.2		79.6

Table 3: Combined accuracy

4.5 Discussion

Although the scores for the grammar and the parser look both encouraging and promising, it is difficult to draw straightforward comparisons with other systems. Charniak (2000) reports a maximum entropy inspired parser that scored 90.1% average precision/recall when trained and tested with sentences from the Wall Street Journal corpus (WSJ). While the difference in precision/recall between the two parsers may indicate the difference in terms of performance between the two parsing approaches, there nevertheless remain two issues to be investigated. Firstly, there is the issue of how text types may influence the performance of the grammar and indeed the parsing system as a whole. Charniak (2000) uses WSJ as both training and testing data and it is reasonable to expect a fairly good overlap in terms of lexical co-occurrences and linguistic structures and hence good performance scores. Indeed, Gildea (2001) suggests that the standard WSJ task seems to be simplified by its homogenous style. It is thus yet to be verified how well the same system will perform when trained and tested on a more 'balanced' corpus such as ICE. Secondly, it is not clear what the performance will be for Charniak's parsing model when dealing with a much more complex grammar such as ICE, which has almost three times as many non-terminal parsing symbols. The performance of the Survey Parser is very close to that of an unlexicalised PCFG parser reported in Klain and Manning (2003) but again WSJ was used for training and testing and it is not clear how well their system will scale up to a typologically more varied corpus.

5 Conclusion

This article described a corpus of contemporary English that is linguistically annotated at both grammatical and syntactic levels. It then described a formal grammar that is automatically generated from the corpus and presented statistics outlining the learning curve of the grammar as a function of training data size. Coverage by the grammar was presented through empirical tests. It then reported the use of the NIST evaluation metric for the evaluation of the grammar when applied by the Survey Parser on test sets totalling 4,000 trees.

Through the size of the grammar in terms of the five canonical phrases as a function of growth in training data size, it was observed that the learning curves for AJP, AVP, and VP culminated fairly rapidly with growing training data size. In contrast, NPs and PPs demonstrate a sharp learning curve, which may have suggested that there would be a lack of sufficient coverage by the grammar for these two phrase types. Experiments show that such a grammar still had a satisfactory coverage for these two with a near total coverage for the other three phrase types.

The NIST scheme was used to evaluate the performance of the grammar when applied in the Survey Parser. An especially advantageous feature of the metric is the calculation of an overall parser performance rate that takes into account the total number of insertions in the parse tree, an important structural distortion factor when calculating the similarity between two trees. A total of 4,000 trees were used to evaluate the labelling and bracketing accuracies of the parse trees automatically produced by the parser. It is shown that the LR rate is over 86% and LP is about 84%. The bracketed recall is 85.8% with a bracketed precision of 82.8%. Finally, an attempt was made to estimate the combined performance score for both labelling and bracketing accuracies. The combined recall is 86.1% and the combined precision is 83.5.

These results show both encouraging and promising performance by the grammar in terms of coverage and accuracy and therefore argue strongly for the case of inducing formal grammars from linguistically annotated corpora. A future research topic is the enhancement of the recall rate for clausal rules, which now stands at just over 65%. It is of great benefit to the parsing community to verify the impact the size of the grammar has on the performance of the parsing system and also to use a typologically more balanced corpus than WSJ as a workbench for grammar/parser development.

References

Charniak, E. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the ACL, Seattle, Washington.*

Fang, A.C. 1996a. Grammatical tagging and cross-tagset mapping. In S. Greenbaum (ed).

Fang, A.C. 1996b. The Survey Parser: Design and development. In S. Greenbaum (ed).

Fang, A.C. 2000. From Cases to Rules and Vice Versa: Robust Practical Parsing with Analogy. In *Proceedings of the Sixth International Workshop on Parsing Technologies, 23-25 February 2000, Trento, Italy*. pp 77-88.

Gildea, D. 2001. Corpus variation and parser performance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2001.*

Greenbaum, S. 1992. A new corpus of English: ICE. In *Directions in Corpus Linguistics: Proceedings of Nobel Symposium 82, Stockholm 4-8 August 1991,* ed. by Jan Svartvik. Berlin: Mouton de Gruyter. pp 171-179.

Greenbaum, S. 1996. *The International Corpus of English*. Oxford: Oxford University Press.

Klein, D. and C. Manning, 2003. Accurate unlexicalized parsing. In *Proceeding of the 41st Annual Meeting of the Association for Computational Linguistics, July 2003.* pp 423-430.