EXPLOITING LARGE CORPORA FOR PARSING

Dominick Ng
James Curran
ə-lab, School of Information Technologies
FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGIES

BACKGROUND

- **Supervised parsers** analyse grammatical structure. Accurate parsing is a vital step for most natural language processing applications
- The defacto standard parser training corpus is based on **1 million words** of **1989 newswire text** that is outdated and domain-specific
- **We exploit very large corpora of unannotated and automatically parsed text for features to address this deficiency and incorporate more knowledge into the parser**

CORPORA

- Web1T (web text) and Google Books (scanned books) provide surface n-gram counts
  - Counts of 1 to 5 adjacent words
- Google Syntactic Ngrams provides dependency structure counts over parsed Google Books
  - Counts of Stanford dependency subtrees
  - Parsed with a "more accurate" baseline parser
- Bansal and Klein [2011] developed surface n-gram count features from Web1T, and improved parsing accuracy by 0.6%

THIS WORK

- **Substitute Google Books for Web1T as a source of surface n-grams**
- **Develop Syntactic Ngrams features**
- **Test over LTH and Stanford dependencies and newswire and web text**

Corpus	Src. Tokens	Size (gzipped)
Web1T	1,000 billion	25 GB
Google Books	468 billion	26 GB
Syntactic Ngrams	345 billion	42 GB

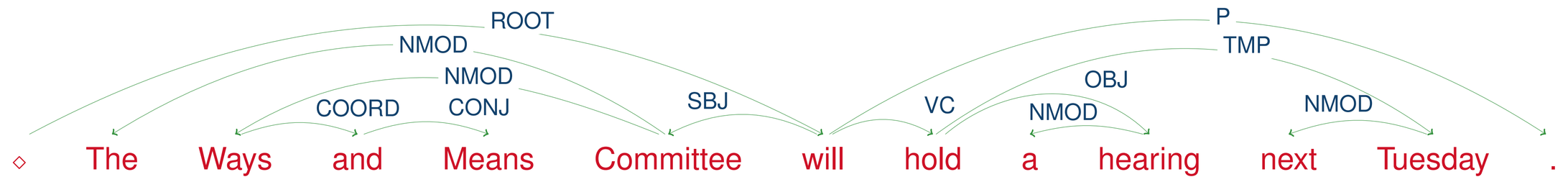

Figure 1: *An* LTH *dependency analysis for a sentence from the* WSJ *development set (section 22).*

```
hold    hold/VBP/ROOT/0 a/DT/det/3 hearing/NN/dobj/1    174    1920,3...
```

Figure 2: *A truncated line from Google Syntactic Ngrams. The fields are head word, the syntactic n-gram (word/*POS*/label/head), total frequency, and frequency by year(s).*

FEATURES

- Each feature is based on a **dependency** between a **head** word and **argument** word
- Encode the **part of speech tags** of the head and argument, and the **bucketed counts** of the dependency in Syntactic Ngrams
- Also encode **bucketed distance** between head and argument, **dependency direction**, and features for each **cumulative count bucket** up to the extracted count

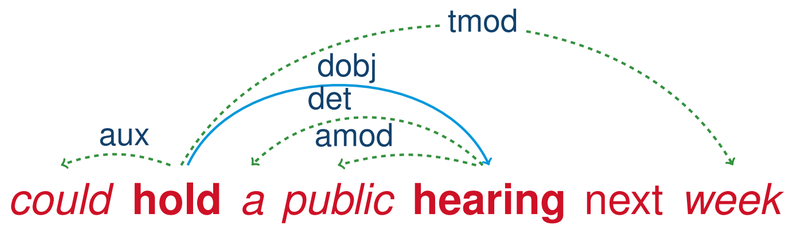

Figure 3: *The context words around* hold→hearing *for which features are extracted. Context words are italicized and their arcs bolded.*

Second Order and Context features

- Higher-order dependency structures
  - parents with two of their children
  - pairs of adjacent siblings
- Context words linked to head or argument
  - **before**, **between**, and **after**
  - head of head, other arguments of head, arguments of argument
- Encode bucketed counts of words and POS tags in before/between/after configurations

Implementation

- All experiments are based on MSTParser, a graph-based second-order dependency parser [McDonald and Pereira, 2006]
- All possible features in the WSJ corpus were extracted and cached prior to runtime

RESULTS

LTH	BASE	WEBT	BOOK	SYNT	WEBT SYNT
WSJ 22	92.3	92.9	92.9	92.8	**93.2**
WSJ 23	91.7	92.2	92.3	92.3	**92.5**
EWT	83.8	84.6	84.5	84.8	**85.2**

Table 1: LTH UAS *on the* WSJ *dev (22) and test sets (23) and averaged over the English Web Treebank (*EWT*) test corpora. All results are statistically significant improvements over the baseline.*

- Table 1 summarises unlabeled accuracy scores for the baseline, surface n-gram features (WEBT, BOOK), syntactic n-gram features (SYNT), and combined WEBT and SYNT
- Google Books is comparable to Web1T for surface n-grams (despite being half the size), with accuracies being statistically indistinguishable between the two
- Surface and syntactic n-gram features produce similar accuracy improvements on newswire (0.7%) and web text (1.0%)
- **Combining the two feature sets in a single model yields up to 1.0% improvement on newswire and 1.4% on web text**
- The Syntactic Ngrams corpus is very noisy
  - syntactic n-gram features perform best with a minimum frequency cutoff of 10,000
  - worse performance at lower frequencies suggest parser errors are being masked by the large volume of text
- Figure 4 shows that syntactic n-grams perform best on verb phrases and conjunctions, but do worse on noun and prepositional phrases – known challenges for parsers

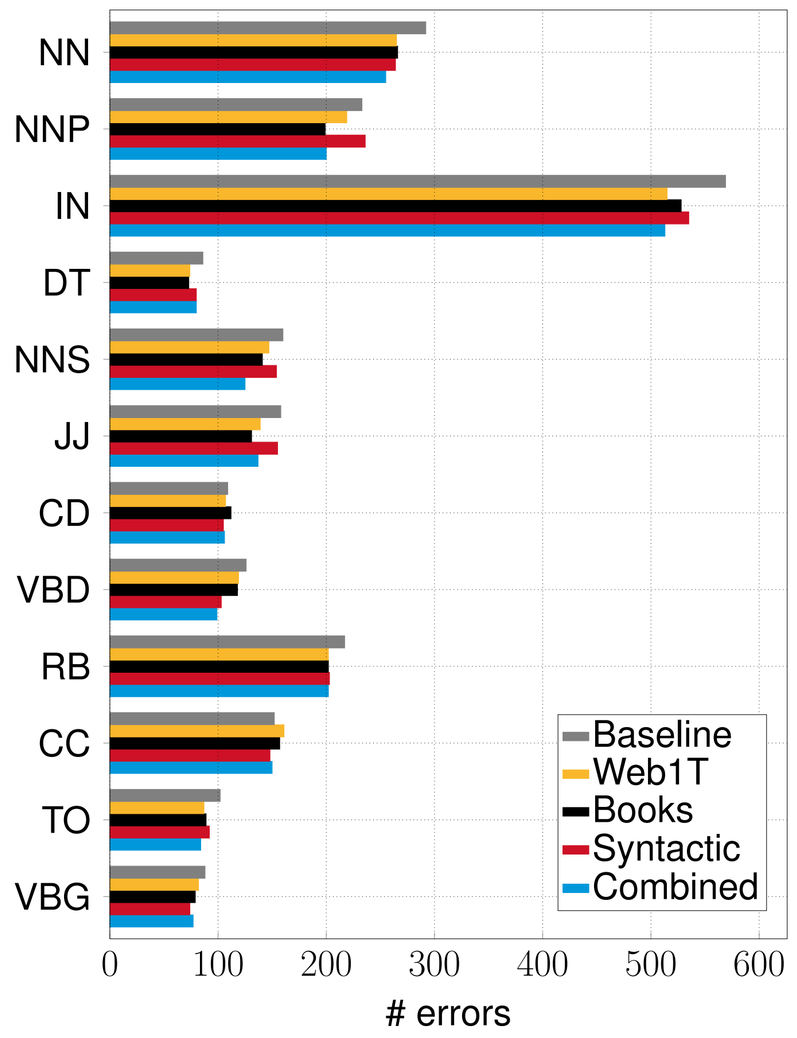

Figure 4: *Total* LTH *attachment errors by gold argument* POS *tag, sorted by the total tag frequency.*

CONCLUSION

- Combined surface and syntactic n-gram features outperform either in isolation
- **We achieve up to 1.4% improved accuracy across LTH and Stanford dependencies, and on newswire and web text**

Acknowledgements

Supported by Australian Research Council Discovery grant DP1097291, an Australian Postgraduate Award, a University of Sydney Vice-Chancellor's Research Scholarship, a Google Australia PhD Fellowship, and a Fulbright Scholarship.

References

Mohit Bansal and Dan Klein. Web-Scale Features for Full-Scale Parsing. In *ACL*, 2011.

Ryan McDonald and Fernando Pereira. Online Learning of Approximate Dependency Parsing Algorithms. In *EACL*, 2006.