Improving the performance of probabilistic parsers on non-WSJ text

Dr. Jennifer Foster

National Centre for Language Technology,

Dublin City University

Talk Outline

- Part One: Parsing Ungrammatical Language
  - Previous work (Foster 2005)
  - Ircset project proposal
  - Progress so far
    - Automatic creation of ungrammatical data
    - Automatic error detection
  - Future work

- Part Two: Parser Adaptation
  - What is parser adaptation?
  - What is self-training?
  - Some research questions
  - Manual parsing of BNC sentences

Previous Work (1)

- Linguistic evidence to test a grammatical theory
  1. Grammaticality judgements on invented sentences
     - Dominant form of evidence in generative linguistics

     **When Mary knows French, she knows it well.*

     When a Moroccan knows French, she knows it well.
     - Three problems:
       - Choice of informant
       - Lack of sentential context
       - Variance in judgement
  2. Corpus Evidence
     - Coincides with the rise of statistical techniques in NLP
     - Grammars induced directly from corpora (no grammatical/ungrammatical distinction)
  3. Combination of both types of evidence
     - Grammaticality judgments made on corpus data *in context*

Previous Work (2)

- A corpus of ungrammatical language
  - Grammaticality judgements on English sentences in context
  - Definition of “ungrammatical”
    - A sentence is ungrammatical if it contains an error and all words in the sentence are well-formed.
    - *The theory in empirical* is included. *The theory is empirrical* is not.
  - Each sentence is corrected > parallel corpus
  - 925 ungrammatical sentences, 1117 grammatical sentences

Previous Work (3)

Error Analysis

- **Replace** a word, **48%**

*His next insult **as** to call me a Republican → His next insult **was** to call me a Republican*

- **Add** a word, **24%**

*Will be declaring their undying love for each other? →Will **they** be declaring their undying love for each other?*

- **Delete** a word, **17%**

*A joint development which will **the** provide 10 new apartments → A joint development which will provide 10 new apartments*

- **Combination** of above (composite errors), **11%**

*What does a single line **yellow** mean? → What does a single **yellow** line mean?*

Previous Work (4)

- Parsing Experiment
  - Hand-crafted context-free grammar (1,705 rules)
  - Error grammar (3,715 rules)
  - Two stage bottom-up chart parser
    - Employs rules from error grammar only when no full spanning parse is produced
    - Error rules added individually to chart on the basis of the partial parse found using the well-formed grammar
  - Results
    - 80% accuracy
    - Reasons for parse failure
      - An implausible parse was found during the first phase
      - More than one error in the sentence
      - Sentence contained a composite error
    - 4-fold increase in parse time

Previous Work (5)

- Another parsing experiment
  - LKB parser, typed feature structures
  - Defined a form of typed feature structures – *robust agreement feature structures*
  - Defined a form of unification – *robust agreement unification*
  - Modified LKB parser to employ robust agreement unification – restricted form of constraint relaxation
  - Applied new parser to agreement error sentences from my corpus using the English Resource Grammar (Copestake and Flickinger, 2000)
  - 94% of sentences were correctly parsed (mean parse#: 22)
  - New version of parser is slower, for both grammatical and ungrammatical sentences

Previous Work (6)

- Parser Evaluation Method
  - Evaluate the parse produced by some parser for an ungrammatical sentence using, as a gold standard, the parse of its grammatical counterpart produced by the same parser
  - Depends on a parallel grammatical/ungrammatical corpus
  - Advantage: no compatibility issue between test parse and reference parse, since parser produces its own reference parse
  - Disadvantage: assume that the parser can accurately parse grammatical text
  - Results on two treebank parsers: (Charniak, 2000) and (Collins, 2003)

Parser	F-Score	100% Match	Problematic
Charniak	90.7	32.8	15.1
Collins	90.2	34.4	17.9

Previous Work (7)

- Some conclusions
  - Probabilistic treebank parsers will produce a parse for an ungrammatical sentence, but not always an accurate one
  - Parser should have a grammar that distinguishes the grammatical from the ungrammatical but which produces a parse for an intelligible ungrammatical sentence
  - The error grammar approach is superior to constraint relaxation
    - Explicit model of ungrammatical language
    - More flexible, suited to many different types of error

Ircset Project Proposal

- Two-stage parser:
  - A probabilistic model of well-formed English
  - A probabilistic model of ill-formed English
- Error detection is needed to trigger the second parsing stage
- Parser can be evaluated using method described previously

Talk Outline

- Part One: Parsing Ungrammatical Language
  - Previous work (Foster 2005) ✓
  - Ircset project proposal ✓
  - **Progress so far**
    - Automatic creation of ungrammatical data
    - Automatic error detection
  - Future work

- Part Two: Parser Adaptation
  - What is parser adaptation?
  - What is self-training?
  - Some research questions
  - Manual parsing of BNC sentences

Error Data Creation (1)

- Why is it **useful** to have a large amount of ungrammatical data?
  - To test a parser's ability to correctly parse ungrammatical language
  - To train machine-learning approaches to error detection
  - To induce a grammar of ungrammatical language
- Why is it **necessary** to do this automatically?
  - Finding errors is time-consuming
  - Almost 1,000 ungrammatical sentences in 18 months
- Empirically motivated method
  - Tagged corpus of grammatical language (BNC)
  - Attempt to introduce an error into each sentence
  - Use as a basis the error analysis on hand-crafted corpus

Error Data Creation (2)

- 8.8 million ungrammatical sentences
  - Extra word errors
    - repeated word errors

    *I think I 'll get Fred **to to** wash his own overalls*
    - double syntactic function errors

    *Do you ever go and visit **the any** of them?*
    - random extra word errors

    *It 'd be one thing less for Neil to worry **and** about*
  - Missing word errors

  He does not mind being butt of his colleagues ' jokes

Error Data Creation (3)

- Context-sensitive spelling errors

*I came **too** the mountain very casually*

- Agreement errors

***Other are** employed in merchant banks advising pension funds*

Error Data Creation (4)

- Limitations
  - Some ungrammatical constructions not covered
    - wrong verb form

    *Brent would often **became** stunned by resentment.*
  - Only one error per sentence
  - Only simple errors (involving one correction operation)
  - Some noise

    *Where he had touched her **her scalps** was prickling like a porcupine .*

Automatic Error Detection (1)

- A research question posed by Joachim Wagner:
  - Can the probability of a sentence's most likely parse be predicted such that the deviation between the predicted and the actual probability reflects the sentence's grammaticality?
- Basic approach:
  - Use the k-nn learning method to predict an estimated parse probability of any input sentence
  - Training data for learning method: various parse and linguistic features of *grammatical* sentences
    - parse tree height, #internal nodes
    - #words, language model probabilities, pos counts
  - If estimated probability is some factor greater than the actual probability, then the sentence is considered to be ungrammatical

Automatic Error Detection (2)

- A slightly different approach
  - Use machine learning to classify a sentence as grammatical or ungrammatical
  - Training data:
    - 100,000/200,000 parsed grammatical BNC sentences
    - 100,000/200,000 parsed ungrammatical BNC sentences (using automatic error creation method)
  - Weka implementation of support vector machines
  - Evaluation carried out using 10-fold cross validation on training data

Automatic Error Detection (3)

- Training data features currently used
  - #words
  - height of most probable parse tree
  - #internal nodes in most probable parse tree
  - probability of most probable parse tree
  - probability of 2nd most probable parse tree
  - POS counts, e.g. #IN, #TO,#DT, etc.
  - #adjacent duplicate POS tags
  - ratio of closed class to open class words in sentence
  - language model probabilities (unigram token, pcfg terminal rules)

Automatic Error Detection (4)

Results

Error Type	Precision	Recall	F-Score
Extra Word	70.7	66.7	68.6
Missing Word	61.4	59.8	60.6
Context Sensitive Spelling	70.6	68.9	69.7
Agreement	62.1	63.2	62.6

Future Work

- Automatic Error Detection
  - Distribution of first fifty parse probabilities
  - Tree height, #internal nodes of 2nd –50th parse tree
  - Vary sentence length range of training data
  - Using a more brittle, less robust grammar to detect an error
    - Hand-crafted wide-coverage grammar (XLE, Rasp)
    - PCFG with a rule threshold
- Two–stage Error Grammar Parsing
  - Create an ungrammatical version of Penn Treebank
  - Done automatically using a combination of robust evaluation software (Foster 2004) and automatic error creation software
  - This can act as training data for the second stage parser
  - Integrate this with DCU LFG parsing

Talk Outline

- Part One: Parsing Ungrammatical Language ✓
  - Previous work (Foster 2005)
  - Ircset project proposal
  - Progress so far
    - Automatic creation of ungrammatical data
    - Automatic error detection
  - Future work

- **Part Two: Parser Adaptation**
  - What is parser adaptation?
  - What is self-training?
  - Some research questions
  - Manual parsing of BNC sentences

Parser Adaptation

- The most widely used parsers of English are very good at parsing sentences from the WSJ
- Don't perform as well on out-of-domain sentences
  - Gildea 2001 – Brown Corpus
  - Judge et al. 2005 – Atis Sentences
- Parser adaptation is the process of adapting a parser to accurately parse out-of-domain data
  - Extend the training set with manually corrected parsed sentences
  - Extend the training set using *self-training*

Self-training

- Self-training involves adding parses produced by the parser itself to the training data.
- Advantage: expand the training set quickly and inexpensively
- In its most basic form, it doesn't work!
- Can be applied successfully with a two-stage parser+reranker model:
  - Sentence is parsed and its 50 best parses are re-ranked.
  - Highest ranked parse is added to training set.
  - When applied to Charniak's parser+reranker with NANC newspaper self-trained examples, performance improved significantly on WSJ23 and on Brown corpus (McCloskey et al 2006)

Some Research Questions (1)

- How do the WSJ-treebank parsers perform on sentences from the British National Corpus?
- Can performance be improved by extending the training set with *carefully selected* manually corrected parsed BNC sentences?
- Can performance be improved by using self-trained BNC examples?
- Can performance be improved by using *carefully selected* self-trained BNC examples?

Some Research Questions (2)

- What do we need to answer these questions?
  - A set of manually corrected parsed BNC sentences
  - A set of carefully selected BNC manually parsed BNC sentences
  - A large number of parsed BNC sentences
  - A large number of carefully selected BNC sentences

Careful Selection

- Generate a list of verbs which appear in the BNC but *not* in Sections 2-21 of the WSJ.
- With each verb in the list, associate its frequency count within the BNC.
- Randomly select sentences from the BNC containing these verbs, giving preference to the more frequent ones.

Gold Standard BNC Parses

- Follow guidelines used by the Penn Treebank annotators (Bies et al. 1995)
- Progress so far
  - 450 sentences
  - approximately 10 parses per hour
  - unclear cases documented to ensure consistency
- Problems/Inconsistencies
  - Quantifier Phrases constructions
    - (NP (QP just 15) months)
    - (NP just (QP a few million))
    - (NP almost two months)
  - Adverb-adjective constructions
    - (NP almost unimaginable speed)
    - (NP (ADJP very poisonous) apple)

Talk Outline

- Part One: Parsing Ungrammatical Language ✓
  - Previous work (Foster 2005)
  - Ircset project proposal
  - Progress so far
    - Automatic creation of ungrammatical data
    - Automatic error detection
  - Future work

- Part Two: Parser Adaptation ✓
  - What is parser adaptation?
  - What is self-training?
  - Some research questions
  - Manual parsing of BNC sentences

THANK YOU FOR LISTENING!