Approximating Parse Probabilities with Simple Probabilistic Models

Joachim Wagner
2006-02-08
Supervisors: Josef van Genabith and Monica Ward

Layout of the Talk

- Motivation
  - grammar checker
  - grammaticality and parse probability
  - my approach to detecting ungrammatical sentences
  - requirements
- Models
  - simple models
  - language modelling
  - combining models
- Conclusions

Motivation

- Grammar checkers are useful in
  - Word processors
  - Computer-assisted language learning
- Current grammar checkers
  - Low-level techniques from computational linguistics, e.g. part-of-speech patterns
  - Hand-crafted grammars → parsing

Hand-Crafted Grammars

- Grammar writing is labour-intensive
  - Needs to be repeated for each language
- Only few grammars with good coverage available for English
- For grammar checkers:
  - Reject ungrammatical sentences
  - 2nd stage: add rules to analyse the error

Data-Driven Methods

- Various methods to train or induce grammars from
  - Labelled corpora, especially treebanks
  - Unlabelled corpora
- Over-generalisation
  - Fail to reject ungrammatical sentences
  - Produce many analyses
- To date, probability models address the latter
- In my research, they will address the former

Research Question

- Can the output of existing probabilistic, data-driven parsers be exploited to:
  - judge grammaticality of sentences
  - locate errors within sentences?
Grammaticality and Parse Probability

- How does grammaticality influence the probability of the most likely parse?
- Parallel error corpus (Foster 2005)
  - 923 ungrammatical sentences
  - 1 or 2 corrections each
  - 2048 sentences in total
- 50% development set

Parallel Error Corpus - Example

- Ungrammatical sentence
  “Does your circles overlap?”
  \( \rightarrow 3.3 \times 10^{28} \)
- Corrected sentence
  “Do your circles overlap?”
  \( \rightarrow 23.0 \times 10^{28} \)

[Student assignment 22/12/04]

Observations 1

Rise of Parse Prob., all 568 pairs

Influence of Sentence Length

Average logarithmic parse probability

Observations 2

Same Sentence Length (250 pairs)
Summary

Sentence
Probabilistic parsing
Probability of most likely parse tree

• Manually correcting an ungrammatical sentence increases probability
• Big variance among different sentences
• Too much overlap for a simple threshold method

Detecting Grammatical Sentences

Do your circles overlap?

Simple probabilistic model

23.0 x 10^{-28} = 23.0 x 10^{-28}

Detecting Ungrammatical Sentences

Does your circles overlap?

Correction

Do your circles overlap?

Simple probabilistic model

23.0 x 10^{-28} = 23.0 x 10^{-28}

Requirements for the Probabilistic Model

• Predict parse probability of grammatical sentence accurately (log. error < 5)
• Fail to emulate the lower probability of ungrammatical sentences
• May look at the parse of the sentence
• Not really predicting the probability of a hypothetical correction

Layout of the Talk

• Motivation
  – grammar checker
  – grammaticality and parse probability
  – my approach to detecting ungrammatical sentences
  – requirements
• Models
  – simple models
  – language modelling
  – combining models
• Conclusions

Instance-Based Learning

• Retrieve similar sentence from training corpus
• Choose parse probability based on these sentences, e.g. average
• k-nearest neighbour method
  – simple implementation
  – few parameters
  – assumes Euclidean space
**k-Nearest Neighbour Method**

- **Input item**
- **Feature extraction**
- **Training data**
- **Retrieval**
- **Best matches**
- **Distance measure**
- **Kernel function**
- **Weighting function**
- **Regression**
- **Target value**
- **Parameter optimisation**
- **Function class**

---

**Example:**

**Does your circles overlap?**

<table>
<thead>
<tr>
<th>Distance</th>
<th>Sentence</th>
<th>Log. Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>Is Mr Fatuzzo there ?</td>
<td>-60.33</td>
</tr>
<tr>
<td>0.44</td>
<td>Is Burma really isolated ?</td>
<td>-82.00</td>
</tr>
<tr>
<td>0.45</td>
<td>Mr Crowley refused )</td>
<td>-56.11</td>
</tr>
<tr>
<td>0.57</td>
<td>Structural Funds ( continuation )</td>
<td>-57.08</td>
</tr>
<tr>
<td>0.60</td>
<td>Euro-Mediterranean cooperation ( continuation )</td>
<td>-67.12</td>
</tr>
<tr>
<td>0.60</td>
<td>Have I understood correctly ?</td>
<td>-49.56</td>
</tr>
<tr>
<td>0.60</td>
<td>Have I understood correctly ?</td>
<td>-49.56</td>
</tr>
<tr>
<td>0.60</td>
<td>Have I understood correctly ?</td>
<td>-49.56</td>
</tr>
<tr>
<td>0.60</td>
<td>Should we reprimand ministers ?</td>
<td>-59.84</td>
</tr>
<tr>
<td>0.63</td>
<td>Loud sustained applause )</td>
<td>-57.88</td>
</tr>
</tbody>
</table>

---

**Training Data**

- ½ of English EuroParl (1.1M sentences)
  - different domain
- Presumably grammatical
- Excluded sentences containing quotes
- Parser crashes and hang-ups
- 409,736 sentences

---

**Evaluation Measures**

- Mean square error of prediction of logarithmic Parse probability of grammatical sentences
  - EuroParl corpus, cross-validation N=10
- Precision and recall in task of classifying sentences
  - Parallel error corpus

---

**Results (1a)**

- Sentence Length only

---

**Example:**

**Do your circles overlap?**

<table>
<thead>
<tr>
<th>Distance</th>
<th>Sentence</th>
<th>Log. Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.24</td>
<td>Is Mr Fatuzzo there ?</td>
<td>-60.33</td>
</tr>
<tr>
<td>0.42</td>
<td>Is Burma really isolated ?</td>
<td>-82.00</td>
</tr>
<tr>
<td>0.68</td>
<td>Should embryos be cloned ?</td>
<td>-75.50</td>
</tr>
<tr>
<td>0.73</td>
<td>( Mr Crowley refused )</td>
<td>-88.11</td>
</tr>
<tr>
<td>0.74</td>
<td>Should we reprimand ministers ?</td>
<td>-59.84</td>
</tr>
<tr>
<td>0.76</td>
<td>Subject : Phare - Poland</td>
<td>-71.85</td>
</tr>
<tr>
<td>0.77</td>
<td>Subject : ASEAN and Burma</td>
<td>-70.43</td>
</tr>
<tr>
<td>0.80</td>
<td>Should we reprimand ministers ?</td>
<td>-59.84</td>
</tr>
<tr>
<td>0.81</td>
<td>Structural Funds ( continuation )</td>
<td>-57.08</td>
</tr>
<tr>
<td>0.81</td>
<td>Have I understood correctly ?</td>
<td>-49.56</td>
</tr>
<tr>
<td>0.81</td>
<td>Structural Funds ( continuation )</td>
<td>-61.16</td>
</tr>
</tbody>
</table>
Results (1b)

Sentence Length Only

Results (2)

Using All 3 Features (Sentence Length, Tree Height and # of internal nodes)

Results (3)

Scaling TH & IN

Character Trigrams

- Easy to integrate into k-NN model (vector of normalised trigram counts)
- Prov
- Ideally, add all trigrams
- K-NN slow with high-dimensional data

Character Trigrams (2)

Trigram Counts / N

Summary – Simple Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean Square Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence length only</td>
<td>640</td>
</tr>
<tr>
<td>Adding tree height and number of internal nodes</td>
<td>580</td>
</tr>
<tr>
<td>Scaling / Weighting</td>
<td>570</td>
</tr>
<tr>
<td>Adding Trigrams</td>
<td>502</td>
</tr>
</tbody>
</table>
Statistical Language Modelling

- Works on raw string of tokens
- Markov assumption: probability of token only depends on previous (n-1) tokens
- N = 1 and MLE: token frequencies
- Unseen events
  - discounting / smoothing

Results

- Mean Square Error
  - 1-gram: 3,464
  - 2-gram: 20,890
  - POS tagged token: 6,778

Terminal Rules of a PCFG

- Condition token probability on POS tag
- Corresponds to terminal rule of PCFG
- Motivation:
  - might be more related to terminal probabilities in parse tree than ordinary language models
- Mean Square Error: 23,352

Combining Models – Method 1

Input Sentence

Feature Extraction

PCFG Model

Language Model

Results (Method 1)

Using Rules and Tokens with k-NN

Combining Models – Method 2

k-NN method

- Training data: extract (divide by) language model output
- Making predictions: include (multiply with) language model output
Results (Method 2)

- Combined with simple model (SL/TH/IN)
- Mean Square Error
  - 1-gram: 200
  - 2-gram: 610
  - PCFG rules: 220
  - POS tagged: 690

Combining Both Methods

Result Broken Down by Sentence Length

Result Broken Down by Sentence Length
Conclusion

• Each idea improved MSE
• Eventually precision increased noticeable above baseline
• Precision not yet very useful
• Many possible ways to further improve the model

Thank you!

• Any questions?