On The Naturalness of Software

SID RASKAR
Computer And Information Science
University Of Delaware
Naturalness?

Central Hypothesis -

- Natural Languages - Simple and repetitive in practice
- Software - Natural product of human effort
- Usefully modelled by Statistical language models

- Can be leveraged to support software engineers
Motivation

- “The European Central Bank announced that interest rates remain unchanged…”
- Bank rather than fish!
- Speech Recognizer, OCR

- Similar Code Completion -
- `For(i=0;i<=10
  ;i++) {`
Language Model

- Assigns probability to an utterance
- Attempts to calculate maximum likelihood estimate of the parameter

N-gram Model -

- Token occurrence is influenced by the n-1 tokens that precede the token in consideration.

\[ p(s) = p(a_1)p(a_2 \mid a_1)p(a_3 \mid a_1a_2)\ldots p(a_n \mid a_1\ldots a_{n-1}) \]
What Makes a Good Model?

- Captures the regularities in the corpus, predicts tokens with high confidence
- Model will not find new document surprising
- In NLP term, cross entropy

$$H_M(s) = -\frac{1}{n} \sum_{i=1}^{n} \log p_M(a_i \mid a_1 \ldots a_{i-1})$$

- Good model has low entropy
- High Probability for frequent words
- Low probability for rare words
Datasets

- Natural Language:
  - Brown and Gutenberg corpus

- For code –
  - Java projects
  - Ubuntu Applications

- Removed comments, produce token sequence
- Each project concatenated as single document
10 Fold Cross Validation

- 90% corpus for training
- 10% corpus for testing
- Unseen tokens smoothed

<table>
<thead>
<tr>
<th>Java Project</th>
<th>Version</th>
<th>Lines</th>
<th>Tokens</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ant</td>
<td>20110123</td>
<td>254457</td>
<td>919148</td>
<td>270008</td>
</tr>
<tr>
<td>Batik</td>
<td>20110118</td>
<td>367293</td>
<td>1384554</td>
<td>30298</td>
</tr>
<tr>
<td>Cassandra</td>
<td>20110122</td>
<td>135992</td>
<td>697498</td>
<td>13002</td>
</tr>
<tr>
<td>Eclipse-E4</td>
<td>20110426</td>
<td>1543206</td>
<td>6807301</td>
<td>98652</td>
</tr>
<tr>
<td>Log4J</td>
<td>20101119</td>
<td>68528</td>
<td>247001</td>
<td>8056</td>
</tr>
<tr>
<td>Lucene</td>
<td>20100319</td>
<td>429957</td>
<td>2130349</td>
<td>32676</td>
</tr>
<tr>
<td>Maven2</td>
<td>20101118</td>
<td>61622</td>
<td>263831</td>
<td>7637</td>
</tr>
<tr>
<td>Maven3</td>
<td>20110122</td>
<td>114527</td>
<td>462397</td>
<td>10839</td>
</tr>
<tr>
<td>Xalan-J</td>
<td>20091212</td>
<td>349837</td>
<td>1085022</td>
<td>39383</td>
</tr>
<tr>
<td>Xerces</td>
<td>20110111</td>
<td>257572</td>
<td>992623</td>
<td>19542</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ubuntu Domain</th>
<th>Version</th>
<th>Lines</th>
<th>Tokens</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin (116)</td>
<td>10.10</td>
<td>9092325</td>
<td>41208531</td>
<td>1140555</td>
</tr>
<tr>
<td>Doc (22)</td>
<td>10.10</td>
<td>87192</td>
<td>362501</td>
<td>15373</td>
</tr>
<tr>
<td>Graphics (21)</td>
<td>10.10</td>
<td>1422514</td>
<td>7453031</td>
<td>188792</td>
</tr>
<tr>
<td>Interp. (23)</td>
<td>10.10</td>
<td>1416361</td>
<td>6388351</td>
<td>201538</td>
</tr>
<tr>
<td>Mail (15)</td>
<td>10.10</td>
<td>1049136</td>
<td>4408776</td>
<td>137324</td>
</tr>
<tr>
<td>Net (86)</td>
<td>10.10</td>
<td>5012473</td>
<td>20666917</td>
<td>541896</td>
</tr>
<tr>
<td>Sound (26)</td>
<td>10.10</td>
<td>1698584</td>
<td>29310969</td>
<td>436377</td>
</tr>
<tr>
<td>Tex (135)</td>
<td>10.10</td>
<td>1405674</td>
<td>14342943</td>
<td>375845</td>
</tr>
<tr>
<td>Text (118)</td>
<td>10.10</td>
<td>1325700</td>
<td>6291804</td>
<td>155177</td>
</tr>
<tr>
<td>Web (31)</td>
<td>10.10</td>
<td>1743376</td>
<td>11361332</td>
<td>216474</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>English Corpus</th>
<th>Version</th>
<th>Lines</th>
<th>Tokens</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown</td>
<td>20101101</td>
<td>81851</td>
<td>1161192</td>
<td>56057</td>
</tr>
<tr>
<td>Gutenberg</td>
<td>20101101</td>
<td>55578</td>
<td>2621613</td>
<td>51156</td>
</tr>
</tbody>
</table>
“Do n-gram language models capture regularities in software?”
Language model captures as much repetitive local context in Java, as it does in English.

Software is far more regular than English.

Increased similarity due to simplicity of Java?

Calculate n-gram models for English and Java.

Self cross entropy.
Is the **local regularity** that the statistical language model captures merely **language specific** or is it also **project specific**?
- Train model on one project and test on another to local regularity

- 10 Projects - Trigram model

- Avg Self entropy is always lower

- Useful language models can be built even for small projects.

- Captures significant levels of local regularity
Do n-gram models capture similarities within and differences between project domain?
- Local Regularities repeated within application domains
- Some domains have very high level of regularity eg. web
Eclipse Suggestion Plug-in

NGSE – n-gram models suggestion engine

ECSE – Eclipse’s built-in suggestion engine

NGSE –
- Tri-gram Model
- 0.2 seconds suggestion time

ECSE good at longer tokens

Simple Merge Algorithm (MSE)

Breakeven length = 7

If ECSE offers long suggestions, pick them greedily

Else

Pick half from ECSE and half from NGSE
Controlled 2 factors -
- String length of suggestions
- Number of choices

Training set – 160 files
Test set – 40 files
Tri gram model

MSE has advantage over ECSE – measured as the gain in number of correct suggestions.

Gains up through 6 character tokens – 33-67%

7 to 15 characters – 3-16%
Related and Future Work

- Naturalness of names in code
- Code Summarization
- Software Mining
- Language Models for accessibility
- Software Tools
Conclusion

Fairly simple statistical model can capture a surprising amount of regularity in natural software which can be leveraged to assist further in software development and maintenance.