Data Intensive Linguistics — Lecture 1
Introduction (I): Words and Probability

Philipp Koehn
9 January 2006

Welcome to DIL

- Lecturer: Philipp Koehn
- TA: Sebastian Fiedel
- Lectures: Mondays and Thursdays, 14:00, FH Room A9/11
- Practical sessions: 4 extra sessions
- Project (worth 30%) will be given out next week
- Exam counts for 70% of the grade

Outline

- Introduction: Words, probability, information theory, n-grams and language modeling
- Methods: tagging, finite state machines, statistical modeling, parsing, clustering
- Applications: Word sense disambiguation, Information retrieval, text categorisation, summarisation, information extraction, question answering
- Statistical Machine Translation

References

- Manning and Schütze: “Foundations of Statistical Language Processing”, 1999, MIT Press, available online
- Jurafsky and Martin: “Speech and Language Processing”, 2000, Prentice Hall.
- Also: research papers, other handouts

MSc Dissertation Topics

- Lattice Decoding for Machine Translation
- Word Alignment for Machine Translation
- Exploiting Factored Translation Models
- Discriminative Training for Machine Translation
- Discontinuous phrases in Statistical Machine Translation
- Learning in factored paradigms using parallel corpora
- Harvesting multi-lingual comparable corpora from the web
- Syntax-Based Models for Statistical Machine Translation

What is Data Intensive Linguistics?

- Data: work on corpora using statistical models or other machine learning methods
- Intensive: fine by me
- Linguistics: computational linguistics vs. natural language processing

Quotes

It must be recognized that the notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term.
Noam Chomsky, 1969

Whenever I hire a linguist our system performance improves.
Frederick Jelinek, 1988

Conflicts?

- Scientist vs. engineer
- Explaining language vs. building applications
- Rationalist vs. empiricist
- Insight vs. data analysis
Why is Language Hard?

- Ambiguities on many levels
- Rules, but many exceptions
- No clear understanding of how humans process language
- Ignore humans, learn from data?

Language as Data

A lot of text is now available in digital form
- Billions of words of news text distributed by the LDC
- Billions of documents on the web (trillion of words?)
- Ten thousands of sentences annotated with syntactic trees for a number of languages (around one million words for English)
- 100~100s of million words translated between English and other languages

Word Counts

One simple statistic: counting words in Mark Twain's Tom Sawyer:

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>3332</td>
</tr>
<tr>
<td>and</td>
<td>2973</td>
</tr>
<tr>
<td>a</td>
<td>1775</td>
</tr>
<tr>
<td>of</td>
<td>1440</td>
</tr>
<tr>
<td>was</td>
<td>1161</td>
</tr>
<tr>
<td>it</td>
<td>1027</td>
</tr>
<tr>
<td>that</td>
<td>877</td>
</tr>
</tbody>
</table>

Counts of counts

<table>
<thead>
<tr>
<th>count</th>
<th>count of count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3993</td>
</tr>
<tr>
<td>2</td>
<td>1292</td>
</tr>
<tr>
<td>3</td>
<td>644</td>
</tr>
<tr>
<td>4</td>
<td>410</td>
</tr>
<tr>
<td>5</td>
<td>243</td>
</tr>
<tr>
<td>6</td>
<td>199</td>
</tr>
<tr>
<td>7</td>
<td>172</td>
</tr>
<tr>
<td>100</td>
<td>99</td>
</tr>
<tr>
<td>&gt; 100</td>
<td>102</td>
</tr>
</tbody>
</table>

Zipf's Law

Zipf's law: \( f \propto r^{-k} \)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Word</th>
<th>Count</th>
<th>( f \times r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the</td>
<td>3332</td>
<td>3332</td>
</tr>
<tr>
<td>2</td>
<td>and</td>
<td>2973</td>
<td>5944</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>1775</td>
<td>5235</td>
</tr>
<tr>
<td>10</td>
<td>he</td>
<td>877</td>
<td>8770</td>
</tr>
<tr>
<td>20</td>
<td>but</td>
<td>410</td>
<td>8200</td>
</tr>
<tr>
<td>30</td>
<td>be</td>
<td>294</td>
<td>8820</td>
</tr>
<tr>
<td>100</td>
<td>two</td>
<td>104</td>
<td>10400</td>
</tr>
<tr>
<td>1000</td>
<td>family</td>
<td>8</td>
<td>8000</td>
</tr>
<tr>
<td>8000</td>
<td>applause</td>
<td>1</td>
<td>8000</td>
</tr>
</tbody>
</table>

Probabilities

- Given word counts, we can estimate a probability distribution:
  \[ P(w) = \frac{\text{count}(w)}{\sum \text{count}(w)} \]
- This type of estimation is called maximum likelihood estimation. Why? We will get to that later.
- Estimating probabilities based on frequency is called the frequentist approach to probability.
- This probability distribution answers the question: If we randomly pick a word out of a text, how likely will it be word \( w \)?

A bit more formal

- We introduced a random variable \( W \).
- We defined a probability distribution \( p \), that tells us how likely the variable \( W \) is the word \( w \):
  \[ \text{prob}(W = w) = p(w) \]

Joint probabilities

- Sometimes, we want to deal with two random variables at the same time.
- Example: Words \( w_1 \) and \( w_2 \) that occur in sequence (a bigram)
  - We model this with the distribution: \( p(w_1, w_2) \)
  - If the occurrence of words in bigrams is independent, we can reduce this to \( p(w_1, w_2) = p(w_1)p(w_2) \) Intuitively, this is not the case for word bigrams.
  - We can estimate joint probabilities over two variables the same way we estimated the probability distribution over a single variable:
    \[ p(w_1, w_2) = \sum_{w_3=\text{count}(w_1, w_2)} p(w_1, w_2, w_3) \]
Conditional probabilities

- Another useful concept is conditional probability
  \[ p(w_2|w_1) \]
  It answers the question: If the random variable \( W_1 = w_1 \), what is the value for the second random variable \( W_2 \)?

- Mathematically, we can define conditional probability as
  \[ p(w_2|w_1) = \frac{p(w_1,w_2)}{p(w_1)} \]

- If \( W_1 \) and \( W_2 \) are independent: \( p(w_2|w_1) = p(w_2) \)

Bayes rule

- Finally, another important rule: Bayes rule
  \[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} \]

- It can easily derived from the chain rule:
  \[ p(x,y) = p(x,y) \]
  \[ p(x|y)p(y) = p(y|x)p(x) \]
  \[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} \]

Chain rule

- A bit of math gives us the chain rule:
  \[ p(w_3|w_1) = \frac{p(w_1,w_3)}{p(w_1)} \]
  \[ p(w_1)p(w_2|w_1) = p(w_1,w_2) \]

- What if we want to break down large joint probabilities like \( p(w_1, w_2, w_3) \)?
  We can repeatedly apply the chain rule:
  \[ p(w_1, w_2, w_3) = p(w_1) p(w_2|w_1) p(w_3|w_1, w_2) \]