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

Philipp Koehn

9 January 2006

Welcome to DIL

- Lecturer: Philipp Koehn
- TA: Sebastian Riedel
- 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 inflectional 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 fire 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 understand 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)
- 10s–100s of million words translated between English and other languages

---

Word Counts

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

Word	Count
the	3332
and	2973
a	1775
to	1725
of	1440
was	1161
it	1027
in	906
that	877

from Manning+Schütze, page 21

Counts of counts

count	count of count
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
...	...
10	91
11-50	540
51-100	99
> 100	102

- 3993 singletons (words that occur only once in the text)
- Most words occur only a very few times.
- Most of the text consists of a few hundred high-frequency words.

Zipf's Law

Zipf's law: $f \times r = k$

Rank r	Word	Count f	$f \times r$
1	the	3332	3332
2	and	2973	5944
3	a	1775	5235
10	he	877	8770
20	but	410	8400
30	be	294	8820
100	two	104	10400
1000	family	8	8000
8000	applausive	1	8000

Probabilities

- Given word counts we can estimate a probability distribution:

  $P(w) = \frac{count(w)}{\sum_{w'} count(w')}$

- This type of estimation is called *maximum likelihood estimation*. Why? We will get to that later.
- Estimating probabilities based on frequencies 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:

  $prob(W = w) = p(w)$

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 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:

  $p(w_1, w_2) = \frac{count(w_1, w_2)}{\sum_{w_{1'}, w_{2'}} count(w_{1'}, w_{2'})}$

Conditional probabilities

- Another useful concept is **conditional probability**

  $p(w_2|w_1)$

  It answers the question: If the random variable $W_1 = w_1$, how 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: $p(w_2|w_1) = p(w_2)$

Chain rule

- A bit of math gives us the chain rule:

  $p(w_2|w_1) = \frac{p(w_1,w_2)}{p(w_1)}$

  $p(w_1)\ p(w_2|w_1) = p(w_1, w_2)$

- 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:

  $p(w_1, w_2, w_3) = p(w_1)\ p(w_2|w_1)\ p(w_3|w_1, w_2)$

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:

  $p(x, y) = p(x, y)$

  $p(x|y)\ p(y) = p(y|x)\ p(x)$

  $p(x|y) = \frac{p(y|x)\ p(x)}{p(y)}$