Accelerated Natural Language Processing

Lecture 1

Introduction

Sharon Goldwater
(based on slides by Philipp Koehn)
Other lecturer: Henry Thompson

17 September 2018
Lecture recording

• Lectures for this course are recorded.

• The microphone picks up my voice, but not yours. (I will repeat questions/comments from students so they are recorded.)

• Signal to me if you want me to pause the recording at any time.

• Normally recording works, but can fail. Don’t rely on it.
What is Natural Language Processing?
Introducing Dragon 13
Increased speed, accuracy and flexibility make it our best Dragon yet.

Learn more

Rajendra Prasad
The 1st President of India

List of Presidents of India - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/List_of_Presidents_of_India

The President of India is the head of state and first citizen of India. The President is also the Commander-in-Chief of the Indian Armed Forces. Although the ... Zakir Hussain - Rajendra Prasad - VV Giri - R. Venkataraman

The many faces of Cortana: How Microsoft's virtual assistant wants to woo the world
Virtual assistants have become commonplace in modern technology, but Microsoft thinks it knows how to push its Cortana a step beyond the rest.
What is Natural Language Processing?

Applications

- Machine Translation
- Information Retrieval
- Question Answering
- Dialogue Systems
- Information Extraction
- Summarization
- Sentiment Analysis
- ... 

Core technologies

- Morphological analysis
- Part-of-speech tagging
- Syntactic parsing
- Named-entity recognition
- Coreference resolution
- Word sense disambiguation
- Textual entailment
- ...
This Course

Linguistics

- words
- morphology
- parts of speech
- syntax
- semantics
- discourse

Computational methods

- finite state machines (morphological analysis, POS tagging)
- grammars and parsing (CKY, statistical parsing)
- probabilistic models and machine learning (HMMS, PCFGs, logistic regression, neural networks)
- vector spaces (distributional semantics)
- lambda calculus (compositional semantics)
This is a simple sentence
Morphology

This is a simple sentence

be
3sg
present

WORDS
MORPHOLOGY
Parts of Speech

This is a simple sentence

be

3sg

present

DT VBZ DT JJ NN

PART OF SPEECH

WORDS

MORPHOLOGY
Syntax

This is a simple sentence

be
3sg
present
This is a simple sentence

be
3sg
present

SIMPLE1 having few parts

SENTENCE1 string of words satisfying the grammatical rules of a language

Sharon Goldwater ANLP Lecture 1 10
This is a simple sentence.

But it is an instructive one.
Why is Language Hard?

- Ambiguities on many levels, need context to disambiguate
- Rules, but many exceptions
- Language is infinite, cannot see examples of everything (and lots of what we do see occurs rarely)
Ambiguity

• Ambiguity is sometimes used intentionally for humor:

1. I’m not a fan of the new pound coin, but then again, I hate all change.¹

2. One morning I shot an elephant in my pajamas. How he got in my pajamas I don’t know.²

• What makes these jokes funny? Is it the same sort of ambiguity, or something different in each case?

¹Ken Cheng, 2017. (Winner of Dave’s Funniest Joke of the Fringe award.)
²Groucho Marx, in the 1930 film Animal Crackers.
Now let’s vote

Do the two jokes have the same sort of ambiguity?

1. Yes
2. No
3. I have no idea what you are talking about
Ambiguity

• However, ambiguity is much more common than jokes.

• Exercise for home: where is the ambiguity in these examples?
  Which is more like Joke 1? Joke 2?
  
  1. This morning I walked to the bank.
  2. I met the woman in the cafe.
  3. I like the other chair better.
  4. I saw the man with glasses.

• We will explain in much more detail later in the course.
Data: Words

Possible definition: strings of letters separated by spaces

- But how about:
  - punctuation: commas, periods, etc are normally not part of words, but others less clear: high-risk, Joe’s, @sloppyjoe
  - compounds: website, Computerlinguistikvorlesung

- And what if there are no spaces:
  伦敦每日快报指出,两台记载黛安娜王妃一九九七年巴黎死亡车祸调查资料的手提电脑,被从前大都会警察总长的办公室里偷走.

Processing text to decide/extract words is called tokenization.
**Word Counts**

Out of 24m total word tokens (instances) in the English Europarl corpus, the most frequent are:

<table>
<thead>
<tr>
<th>any word</th>
<th>Frequency</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,698,599</td>
<td>the</td>
<td></td>
</tr>
<tr>
<td>849,256</td>
<td>of</td>
<td></td>
</tr>
<tr>
<td>793,731</td>
<td>to</td>
<td></td>
</tr>
<tr>
<td>640,257</td>
<td>and</td>
<td></td>
</tr>
<tr>
<td>508,560</td>
<td>in</td>
<td></td>
</tr>
<tr>
<td>407,638</td>
<td>that</td>
<td></td>
</tr>
<tr>
<td>400,467</td>
<td>is</td>
<td></td>
</tr>
<tr>
<td>394,778</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>263,040</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>nouns</th>
<th>Frequency</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>124,598</td>
<td>European</td>
<td></td>
</tr>
<tr>
<td>104,325</td>
<td>Mr</td>
<td></td>
</tr>
<tr>
<td>92,195</td>
<td>Commission</td>
<td></td>
</tr>
<tr>
<td>66,781</td>
<td>President</td>
<td></td>
</tr>
<tr>
<td>62,867</td>
<td>Parliament</td>
<td></td>
</tr>
<tr>
<td>57,804</td>
<td>Union</td>
<td></td>
</tr>
<tr>
<td>53,683</td>
<td>report</td>
<td></td>
</tr>
<tr>
<td>53,547</td>
<td>Council</td>
<td></td>
</tr>
<tr>
<td>45,842</td>
<td>States</td>
<td></td>
</tr>
</tbody>
</table>
Word Counts

But there are 93638 distinct words (types) altogether, and 36231 occur only once! Examples:

- cornflakes, mathematicians, fuzziness, jumbling
- pseudo-rapporteur, lobby-ridden, perfunctorily,
- Lycketoft, UNCITRAL, H-0695
- policyfor, Commissioneris, 145.95, 27a
Plotting word frequencies

Order words by frequency. What is the freq of \( n \)th ranked word?

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Token</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,698,599</td>
<td>the</td>
<td>1</td>
</tr>
<tr>
<td>849,256</td>
<td>of</td>
<td>2</td>
</tr>
<tr>
<td>793,731</td>
<td>to</td>
<td>3</td>
</tr>
<tr>
<td>640,257</td>
<td>and</td>
<td>4</td>
</tr>
<tr>
<td>508,560</td>
<td>in</td>
<td>5</td>
</tr>
<tr>
<td>407,638</td>
<td>that</td>
<td>6</td>
</tr>
<tr>
<td>400,467</td>
<td>is</td>
<td>7</td>
</tr>
<tr>
<td>394,778</td>
<td>a</td>
<td>8</td>
</tr>
<tr>
<td>263,040</td>
<td>I</td>
<td>9</td>
</tr>
</tbody>
</table>
Plotting word frequencies

Order words by frequency. What is the freq of $n$th ranked word?
To really see what’s going on, use logarithmic axes:
Zipf’s law

Summarizes the behaviour we just saw:

\[ f \times r \approx k \]

- \( f \) = frequency of a word
- \( r \) = rank of a word (if sorted by frequency)
- \( k \) = a constant
Zipf’s law

Summarizes the behaviour we just saw:

\[ f \times r \approx k \]

- \( f \) = frequency of a word
- \( r \) = rank of a word (if sorted by frequency)
- \( k \) = a constant

Why a line in log-scales?

\[ fr = k \quad \Rightarrow \quad f = \frac{k}{r} \quad \Rightarrow \quad \log f = \log k - \log r \]

\[ y = c - x \]
Linguistics and Data

• Data
  – looking at real use of language in text
  – can learn a lot from empirical evidence
  – but: Zipf’s law means there will always be rare instances

• Linguistics
  – build a better understanding of language structure
  – linguistic analysis points to what is important
  – but: many ambiguities cannot be explained easily
Course organization

• Lecturers: Sharon Goldwater, Henry Thompson

• 3 lectures per week (Mon/Tue/Thu)

• Weekly, in alternate weeks (1st lab is this week):
  – 1.5 hr lab for exploring data and developing practical skills
  – 1 hr tutorial for working through maths and algorithms

• Labs will be done in pairs; tutorial work can be done with whomever you choose.
Course materials and communication

• Available on Learn page, even if you are not yet registered (link on course.inf.ed.ac.uk)

• Main textbook: “Speech and Language Processing”, Jurafsky and Martin. We use both 2nd Ed (2008) and 3rd Ed (draft chapters).

• Labs, assignments, code, optional readings: all on web page.

• We use the Piazza discussion forum. Sign up now using link on Learn!
Assessment

- Two assessed assignments, worth 30% altogether.
  - require some programming, but assessed on explanations and “lab-report” style write-ups.
  - You may (and are encouraged to) work in pairs.
  - Note: there will also be a zero-weighted assignment with auto-testing for feedback.

- Exam in December, worth 70% of final mark.
  - short factual answers, longer open-ended answers, problem-solving (maths, linguistics, algorithms).
British higher education system

• Main principle: self-study guided by non-assessed work (some of it used for formative feedback), final assessed exam.

• Do not expect to learn everything just by sitting in lectures and tutorials! Most of your time should be in self-study:
  – Labs: intended to be done during scheduled lab times, but you may wish to look over them in advance (or revise after).
  – Tutorial sessions: do exercises in advance, bring questions. Discussion to help answer, learn more, and provide feedback.
  – Assignments: two assessed, one not.
  – Other: reading textbook, working through examples and review questions, seeking out online materials, group study sessions.
Background needed for this course?

- Know or currently learning Python.
- Background in Linguistics and prepared to learn maths (mainly probability) and algorithms
- Background in CS and prepared to learn linguistics (and maybe maths)
Advice/warnings

• Students with little programming/maths: you can do it, but it will be very intensive.
  – Find study partners, start work early.
  – Pair up with a computer scientist.

• Students with programming but little maths or weak English: you can do it, but it will be very intensive.
  – Find study partners, start work early.
  – Pair up with a linguist or someone with stronger English.

• Students with strong programming/maths/machine learning: still fairly intense, plenty of scope for challenge. Don’t underestimate the need to develop critical thinking and writing skills.
“What would you say to students interested in taking this course?”

Do everything that you are told to do/read, do not underestimate anything, devote a lot of time.

It is a good course. Although it is very intensive, I did learn a lot of stuff than I expected. As long as you take advantage of all the learning resources provided and work hard on every assignment, you will definitely benefit a lot from it.

It’s a great course, but it’s not a walk in the park, so be prepared to work hard.

You’ll learn a lot, but it is challenging.
What this course is, and isn’t

This course is a fast-paced introduction/survey course. We will

• introduce many of the basic tasks in NLP and discuss why they are challenging

• present linguistic concepts and standard methods (maths/algorithms) often used to solve these tasks

• give you enough background to be able to read (some) current NLP research papers and take follow-on courses in sem 2

But we will not

• say too much about cutting edge methods or heavy-duty machine learning (see ML courses and NLU+)
Relationship to other NLP courses

• Alternative intro NLP course: FNLP (10 pts, yr3 UG, sem 2)
  – Covers some similar material, but not all
  – Less fast-paced, less open-ended assignments/exam.
  – But *does* assume Python programming and probability.
  – Good if you just want a taster.
  – You **may not** take both FNLP and ANLP.

• Alternative text processing course: TTDS (20 pts, MSc, full year)
  – Focuses more on web search and shallow text processing
  – Less about the subtleties of language structure and meaning
  – More weight on practicals, including team project
  – Assumes more maths and programming background
Relationship to other NLP courses

• ANLP is **required** if you want to take more NLP in sem 2:
  - Natural Language Understanding, Generation, and Machine Translation (recent advances, including lots about neural network approaches).
Preparing for next week

- We will be starting with probabilistic models next week.

- If you haven’t taken a course on probability theory (or related), start working through the tutorial now (link on week 2 of lecture schedule).

- Probabilistic material starts early to give you longer to absorb before the exam.

- In general, material is front-loaded: you’ll have more assignments from other courses later on.
Labs start this week!

- Four available times on Wed/Thu/Fri afternoons.
- To see which to attend, check on Learn this evening (and sign up for Piazza while you’re there!)
- These times are for this week only. We’ll sort out later weeks once everyone is registered for the class.
Preparing for the first lab

• **Before your lab**: do the Preliminaries section of Lab 1. That is,
  – Get your DICE account and make sure you can log in to the lab machines in AT (or find a partner who can).
  – Read/work through the Introduction to DICE (linked from the lab) while at a DICE machine.
Questions and exercises:

• What does ambiguity refer to? Does it always involve a word with two different meanings?

• Do the exercise on slide 14.

• What is a word token? A word type? How many tokens and how many types are there in the following sentence?
  the new chair chaired the meeting

• What does Zipf’s law describe, and what are its implications? (We will see more about implications in the next few lectures.)