What is Natural Language Processing?

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

Core technologies
- Language modelling
- Part-of-speech tagging
- Syntactic parsing
- Named-entity recognition
- Coreference resolution
- Word sense disambiguation
- Semantic Role Labelling
- ...

This course

NLP is a big field! We focus mainly on core ideas and methods needed for technologies in the second column (and eventually for applications).

- Linguistic facts and issues
- Computational models and algorithms

More advanced methods and specific application areas covered in 4th/5th year courses:

- Natural Language Understanding
- Machine Translation
- Topics in NLP
- Automatic Speech Recognition
What does an NLP system need to “know”?

- Language consists of many levels of structure
- Humans fluently integrate all of these in producing/understanding language
- Ideally, so would a computer!
**Syntax**

This is a simple sentence

be
3sg
present

**Semantics**

∃y(\(this_{\text{dem}}(x) \land be(e, x, y) \land simple(y) \land sentence(y)\))

**Discourse**

But it is an instructive one.

**Why is NLP hard?**

1. **Ambiguity** at many levels:
   - Word senses: bank (finance or river?)
   - Part of speech: chair (noun or verb?)
   - Syntactic structure: I saw a man with a telescope
   - Quantifier scope: Every child loves some movie
   - Multiple: I saw her duck
   - Reference: John dropped the goblet onto the glass table and it broke.
   - Discourse: The meeting is cancelled. Nicholas isn’t coming to the office today.

How can we model ambiguity, and choose the correct analysis in context?
Ambiguity

Inf2a started to discuss methods of dealing with ambiguity.

- non-probabilistic methods (FSMs for morphology, CKY parsers for syntax) return all possible analyses.

- probabilistic models (HMMs for POS tagging, PCFGs for syntax) and algorithms (Viterbi, probabilistic CKY) return the best possible analysis, i.e., the most probable one according to the model.

This “best” analysis is only good if our model’s probabilities are accurate. Where do they come from?

Why is NLP hard?

2. Sparse data due to Zipf’s Law.

- To illustrate, let’s look at the frequencies of different words in a large text corpus.
- Assume a “word” is a string of letters separated by spaces (a great oversimplification, we’ll return to this issue...)

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

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

But also, out of 93638 distinct word types, 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 frequency of nth ranked word?

Rescaling the axes

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

Why a line in log-scales? \( fr = k \Rightarrow f = \frac{k}{r} \Rightarrow \log f = \log k - \log r \)

Implications of Zipf’s Law

- Regardless of how large our corpus is, there will be a lot of infrequent (and zero-frequency!) words.
- In fact, the same holds for many other levels of linguistic structure (e.g., syntactic rules in a CFG).
- This means we need to find clever ways to estimate probabilities for things we have rarely or never seen.
Why is NLP hard?

3. Variation

- Suppose we train a part of speech tagger on the Wall Street Journal:
  
  Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsevier/NNP
  N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/NN ./.

Alex Lascarides  FNLP Lecture 1  24

Why is NLP hard?

4. Expressivity

- Not only can one form have different meanings (ambiguity) but the same
  meaning can be expressed with different forms:

  She gave the book to Tom vs. She gave Tom the book
  Some kids popped by vs. A few children visited
  Is that window still open? vs Please close the window

Alex Lascarides  FNLP Lecture 1  26

Why is NLP hard?

3. Variation

- Suppose we train a part of speech tagger on the Wall Street Journal:
  
  Mr./NNP Vinken/NNP is/VBZ chairman/NN of/IN Elsevier/NNP
  N.V./NNP ,/, the/DT Dutch/NNP publishing/VBG group/NN ./.

- What will happen if we try to use this tagger for social media??

  ikr smh he asked fir yo last name

Twitter example due to Noah Smith

Alex Lascarides  FNLP Lecture 1  25

Why is NLP hard?

5 and 6. Context dependence and Unknown representation

- Last example also shows that correct interpretation is context-dependent and
  often requires world knowledge.
- Very difficult to capture, since we don't even know how to represent the
  knowledge a human has/needs: What is the “meaning” of a word or sentence?
  How to model context? Other general knowledge?

That is, in the limit NLP is hard because AI is hard

- In particular, we've made remarkably little progress on the Knowledge
  Representation problem...
Background needed for this course
We assume you are familiar with most/all of the following:

• Basic Python programming
• Finite-state machines, regular languages
• Context-free grammars
• Dynamic programming (e.g. edit distance, Viterbi, and/or CKY algorithms)
• Concepts from machine learning (estimating probabilities, making predictions based on data)
• Probability theory (conditional probabilities, Bayes’ Rule, independence and conditional independence, expectations)
• Vectors, logarithms
• Concepts of syntactic structure and semantics and relationship between them (ideally for natural language but at least for programming languages)
• Some basic linguistic concepts (e.g. parts of speech, inflection)

Where we are headed
Informatics 2a discussed ideas and algorithms for NLP from a largely formal, algorithmic perspective. Here we build on that by

• Focusing on real data with all its complexities.
• Discussing some of the algorithms in more depth, as probabilistic inference.
• Introducing some tasks and technologies that didn’t fit into the Inf2a story.

Course organization

• Lecturer: Alex Lascarides
• Lectures: Tue/Fri 10:00-10:50
• Labs: two groups
  Reply to the email from ITO to be assigned a group.
  Labs start next week!
• Web site: for slides, lectures, labs, assignments, due dates, etc
  http://www.inf.ed.ac.uk/teaching/courses/fnlp/
• Course mailing list: fnlp-students@inf. Register ASAP to get on the list!
• Course discussion forum: Piazza.
  Link for signing up to FNLP’s piazza page is on FNLP website.

Outside work required
In addition to attending lectures, you are expected to keep up with:

• Readings from textbook: Speech and Language Processing, 3rd edition (online) and 2nd edition (paperback, International version), Jurafsky and Martin.
• Weekly (unassessed) labs (in Python). To help solidify concepts and give you practical experience. Help and feedback available from lab demonstrator.
• Lectures are being recorded. Recordings will be linked from the lectures page week by week. The audience is not in shot.
• Two assignments (in Python)
  − The second worth 30%
  − The first will be reviewed and marked, but will not contribute to your final mark
• Exam in May, worth 70% of final mark.

We will also provide some optional further readings/exercises for those who wish to stretch themselves. These will be clearly marked as optional (non-examinable).