LIN7076 – Foundations of Computational Linguistics

General Introduction

Adèle Hénot-Mortier

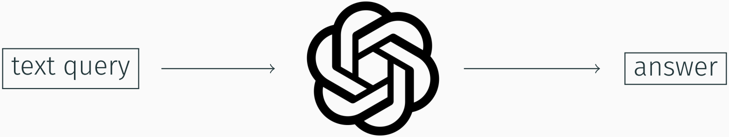
23/09/2025

Queen Mary University of London

Computational Linguistics and applications

What is computational linguistics?

The goal of Computational Linguistics is to make natural language **interpretable to computers**, to perform **automated and sometimes complex operations** on this input, and produce high-level, **interpretable outputs**.

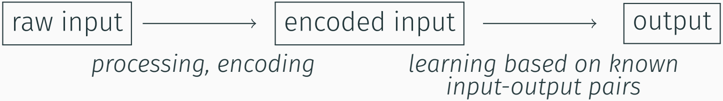


text query → → answer

preprocessing
encoding
decoding
reinforcement learning...

Supervised vs. unsupervised methods

- Many methods we will see in this course are **supervised**, which means that a model learns to generalize to new inputs from a bunch of **known input-output pairs**.
- Supervised methods requires **labeling**: you explicitly tell your model that certain inputs are this way, others are that way.



raw input → encoded input → output

processing, encoding

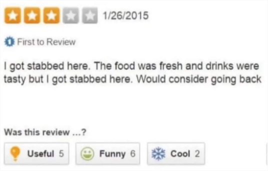
learning based on known input-output pairs

- Methods that are not based on labeling and only exploit the inherent organization of the input data are **unsupervised**.
- For instance, one can learn a lot from word cooccurrences!

What computational linguistics is not

- Computational Linguistics is usually **not intended as a model of human language faculty**.
- Instead, it applies **general-purpose techniques** to linguistic inputs, in order to solve **real-world application**.
  - The input passed to such systems is often linguistically impoverished: just raw strings of characters!
  - The systems themselves do not inherently feature the phonology/morphology/syntax/semantics divide that linguists had good reasons to posit. Whether they learn it remains quite an open question.
- CompLing models can get very complex, and it can become challenging to understand the motivations behind them.
- Those are amazing (and imo beautiful) mathematical tools fit to **process linguistic inputs**, but are not seen as models of our grammar.

Sentiment analysis



1/26/2015

First to Review

I got stabbed here. The food was fresh and drinks were tasty but I got stabbed here. Would consider going back

Was this review ...?

- Reviews are encoded (“tokenized”/“featurized”) and assigned a label by the model: 0 (negative) or 1 (positive).
- It’s a kind of classification task – binary classification.
- Challenges: borderline cases; polarity (negation); irony; register; socio-economic factors.

Authorship identification

The Washington Post
House Plan Would
Unabomber Manuscript Is Published
'Public Safety Reasons' Cited in Joint Decision By Post, N.Y. Times
A SUPPLEMENT TO THE WASHINGTON POST
INDUSTRIAL SOCIETY AND ITS FUTURE
INTRODUCTION

- A document is encoded and assigned a label by the model, encoded as a vector whose *i*-th component encodes label *i* (an author).
- Challenges: critical (e.g. legal) decisions; overfitting, few-shot learning.

Word vectors / embeddings

Sunsets are **so**	pretty	
The **red dress** is	pretti-	**er** than the **blue** one
Jo finds **Crocs**	pretty	
Anglerfish do **not** look	pretty	
This is a	pretty	**ugly** way to say it

- Words are assigned a vector of real numbers “compressing” the linguistic environments in which they tend to appear.
- The “compression” technique ensures that *pretty* will be close to *lovely* or *beautiful*, and distant from e.g. *engine* or *the*. Check out the Embedding Projector!
- Challenges: morphological variation; lexical ambiguity; language shifts; entanglement between syntax and semantics; “non-linear” behavior of natural language.

Question answering

Q : Would you still love me if I was a worm?
A : …

- Option 1: the question is encoded and mapped to an answer.
- Option 2: the question is not directly encoded as such, but “tagged” with “Q” and “A” labels, and the model iteratively generates a continuation.
- Challenges: LLM “hallucinations”; source retrieval (see Retrieval-Augmented Generation or RAG); pragmatics; ethical considerations…

Beyond CompLing

- The following techniques are used in a wide variety of domains beyond CompLing:
  - regression: associate a bunch of input features with a numerical or categorical variables.
  - Bayesian classification: determine the most likely label, given a bunch of input features.
  - embeddings: map objects (words, images...) into a space in which distance is meaningful.
  - neural nets: with a sufficiently complex structure and the right activation, can approximate any continuous function!

Ethical and environmental challenges

Bias

- Models are designed to learn statistical patterns: for instance that a determiner is often followed by a noun; that auxiliaries precede subjects in questions; that *any* likes to be within the scope of negation etc.
- But models learn primarily from naturalistic human data, which may is known to contain a lot of stereotypes, biased or even harmful content.
- Models may therefore internalize harmful patterns, among all the other useful patterns they learn. They can't tell the difference!
- Using these models to solve real-life applications may then have harmful consequences: discrimination, exclusion, filter bubbles...
- Controlling for harmful biases without compromising performance too much is a challenging yet necessary enterprise.

Representation

- Until recently, CompLing had remained very English centric; so called low-resource languages were entirely neglected.
- This has changed a little due the increase of data availability, and also thanks to models and techniques allowing to learn from less, sometimes piggybacking on other better-endowed languages.
- Still, even within the English language, minoritized variants are not so well-represented.
- This leads to models to perform poorly on such variants, again, with potential real-life consequences (misunderstanding, censoring, discrimination...).

Environmental considerations

- The models we will see this semester are still relatively “light”: we’ll train most of them in just minutes.
- But most recent models took months to train, costing several millions.
- This Medium article attempts to estimate the carbon footprint of GPT-4, based on unverified leaked data. Depending on where the datacenters were, it could be up to 15,000 metric tons of CO_2, around 2500 flights from London to SF. And that’s just one model!
- There is a push to come back to comparably smaller models, for environmental reasons, but also for efficiency reasons (bigger models don’t get indefinitely better!), and accessibility reasons.

Class Logistics

Goals

- Understand the **core intuitions** behind fundamental NLP algorithms.
- Not all the algorithms we will see are still used in modern applications; but they are still interesting to **understand how we got where we currently are**. Many modern tools, including LLMs, are based on elaboration of the core intuitions we will see this semester!
- They are also interesting as **baselines**, and constitute quick, cheap, accessible, and environment-friendly ways to build simple applications!
- Lastly, many methods we will explore are fairly **transparent**: you can grasp most of what's happening under the hood. So it's easier to connect these methods to intuitions about human reasoning, and linguistic facts.

Main textbook

- Most of the content we will cover is discussed in depth in Jurafsky and Martin's *Speech and Language Processing* textbook.
- This textbook is an amazing resource, and is freely available online.
- We'll however use the January 2025 version, because it is a bit less LLM-centric than the most recent August 2025 version, and as such I find it better suited to this intro course.

Organization of the classes

- The main concepts underlying the topic of the week will be presented in a traditional lecture format.
    - I encourage you to read the relevant chapter of J&M's textbook to deepen your understanding of the topic! The more advanced subsections can be skipped.
    - I also encourage note-taking, preferably pen and paper, but eventually it's your call.
- A "lab" will demonstrate how the concepts can be implemented or used, usually in Python, sometimes in R.
    - We will use Google Colab notebooks for these labs.
    - You should bring your laptop – we will mostly reflect on the existing code, but a little bit of extra coding will be involved each time.

Participation

- Questions and comments are welcome at any time during class!
- It’s my first time teaching this so **getting feedback from you is essential**.
- Asking questions is also a **favor you do to your peers** who might be too shy to ask (I was definitely such a student).

Assessment

- There will be two assessments, each representing 50% of the grade.
- The first assessment ($\sim$ 2000 words) will be a homework covering the weeks before Reading Week. It will be due by the end of Reading Week.
- The second assessment ($\sim$ 2000 words) will be a small project or a detailed project proposal. It will be due by the end of the semester. Feel free to meet with me prior to submission!

Use of generative AI

- You are allowed to use generative AI in this class. But if you do, I'd like to know, and also would like to know what you used it for (spelling/style/code/more...)! It's a matter of honesty and equity.
- ChatGPT has gotten pretty good with Python and even R. I've used it myself, especially with R stuff. But it's important to understand what it spits out!
- This course is less about learning to code from scratch, than about learning to understand and adapt existing code. Please don't forget this modest but important objective!