ECS763P
Natural Language Processing

Mehrnoosh Sadrzadeh
Massimo Poesio

Week 1: Introduction

NLP Applications: 1

NLP Applications 2: managing big (textual) data

- CLASSIFY text so as to identify relevant content / quickly assess this content
  - E.g., SENTIMENT ANALYSIS
- EXTRACT structured information from unstructured textual data
- SUMMARIZING text

SENTIMENT ANALYSIS (Esp. on social media)

> Id: Abc123 on 5-1-2008 "I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too.
>
> It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, …"

SENTIMENT ANALYSIS

Id: Abc123 on 5-1-2008 “I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too.

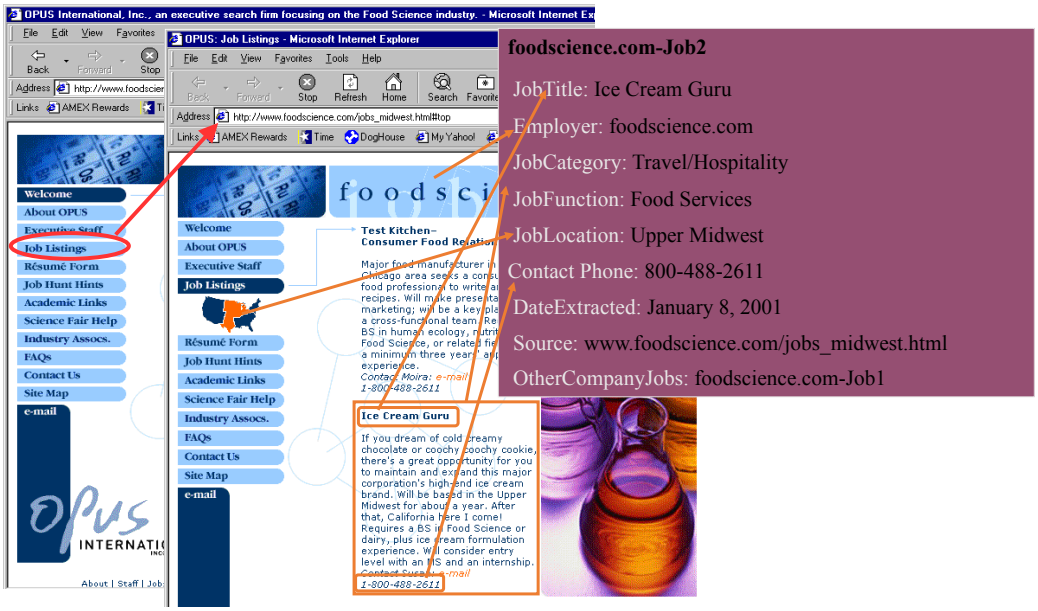
It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, …”

SENTIMENT ANALYSIS

Id: Abc123 on 5-1-2008 “I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too.

It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, …”

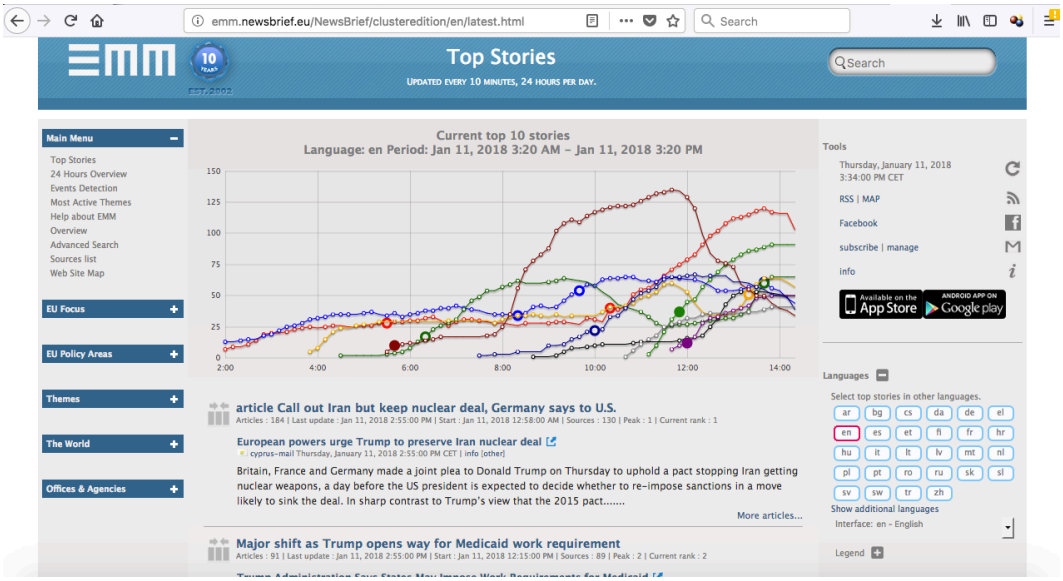
INFORMATION EXTRACTION: FINDING JOBS ON THE WEB



Summarization

- Summarization is the production of a summary either from a single source (single-document summarization) or from a collection of articles (multi-document summarization)

Clustering and summarization



Example of NLP application: Sentiment Analysis

- A basic NLP task
- Automatic decision:
  - positive vs negative
    - *I'm really happy!*
    - *I'm having a terrible day*
    - *Oh man this is so great <3*
    - *I just can't believe it*
- How could we go about this?
- What's the simplest way you can think of?

Pre-processing

- We're going to have to use the words
  - (what else is there?)
- But how do actually we get to them?
- At least:
  - Sentence segmentation
    - (split? At what?)
  - Word tokenisation
    - (split? At what?)
- And maybe:
  - Normalisation, spelling correction
    - (how?)
  - Stop word removal
    - (really?)

Sentiment analysis with words 1: Dictionaries

- We could build dictionaries:
  - List of "positive" words
  - List of "negative" words
- Score outputs based on:
  - number of words
  - weights …
  - ... etc

Example code

- dict1

Words 1: Dictionaries

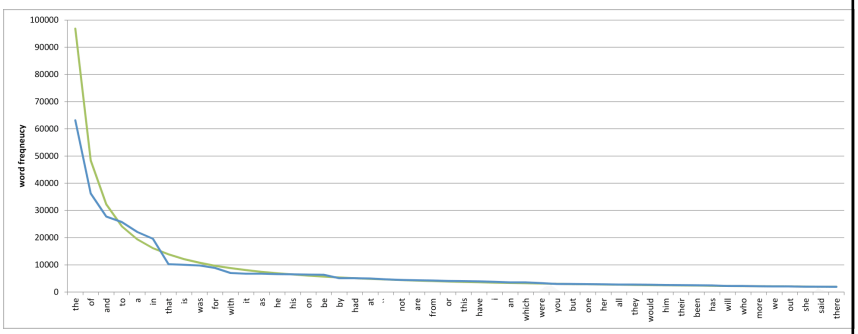
- We could build dictionaries:
  - List of “positive” words
  - List of “negative” words
- Score outputs based on:
  - number of words
  - weights …
  - ... etc
- Problems?

KEY POINT:

Language is Zipfian

Zipf's Law

- The frequency of any word is inversely proportional to its rank in the frequency table
  - Brown corpus:
    - rank 1 'the': 7%
    - rank 2 'of':
    - rank 3 'and': 2.9%



shadycharacters.co.uk

- This means:
  - We can capture most of the data easily
  - But there is a **very** long tail
  - And however big your corpus …
  - … you will see new words as soon as you look outside it!

KEY POINT:

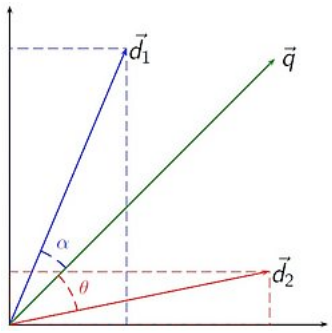
Words are not independent

Words 2: Statistical Models

- We could **learn** these dictionaries
- Or we could train a classifier:
  - List of “positive” examples
  - List of “negative” examples
- Learn a decision function based on observed words … and combinations thereof

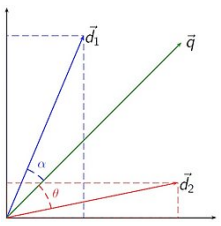
Texts as Feature Spaces

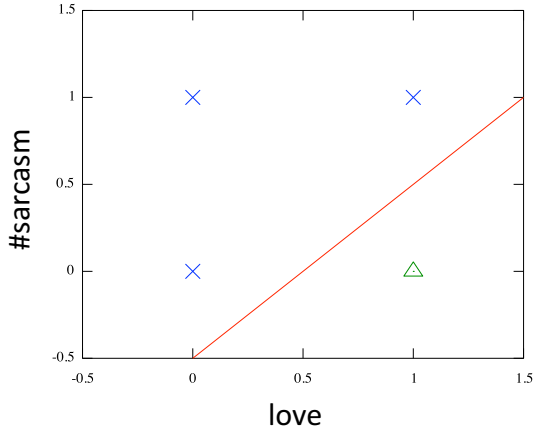
- We can characterise a text in terms of its words
- Vector space models
  - words = dimensions
- "Bag of words" model



Classification in Feature Spaces

- E.g. binary classification with SVMs
  - `i love @justinbieber #sarcasm`





Example code

- svm1

A few further issues with word-based models

- Language identification
- Tokenization
- Normalization

Tokenization

- Issues in tokenization:
  - ***Finland's capital*** → ***Finland? Finlands? Finland's***?
  - ***Hewlett-Packard*** → ***Hewlett*** and ***Packard*** as two tokens?
    - ***state-of-the-art***: break up hyphenated sequence.
    - ***co-education***
    - ***lowercase***, ***lower-case***, ***lower case*** ?
    - It's effective to get the user to put in possible hyphens
  - ***San Francisco***: one token or two? How do you decide it is one token?

Normalization

- Need to "normalize" terms in indexed text as well as query terms into the same form
  - We want to match ***U.S.A.*** and ***USA***
- We most commonly implicitly define equivalence classes of terms
  - e.g., by deleting periods in a term
- Alternative is to do asymmetric expansion:
  - Enter: ***window*** Search: ***window, windows***
  - Enter: ***windows*** Search: ***Windows, windows, window***
  - Enter: ***Windows*** Search: ***Windows***
- Potentially more powerful, but less efficient

Normalization: other languages

- Accents: ***résumé*** vs. ***resume***.
- Most important criterion:
  - How are your users like to write their queries for these words?
- Even in languages that standardly have accents, users often may not type them
- German: ***Tuebingen*** vs. ***Tübingen***
  - Should be equivalent

What about …

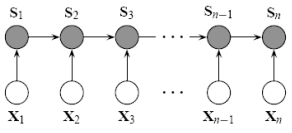
- Milk is good and not expensive
- Milk is expensive and not good

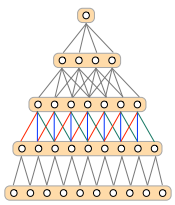
KEY POINT:

Language is not just a bag of words

Sequence modelling

- We can get a long way by using **sequence**
  - N-grams
    - [milk is], [is good], [good and], [and not], [not expensive]
    - [milk is], [is expensive], [expensive and], [and not], [not good]
  - Sequence models
    - Markov models
    - Conditional random fields
  - Convolutional / recurrent neural nets





What about …

- Milk is not very good
- Milk is not really very good
- Milk is not bad but good
- As bad as milk is, good things can come from it

- I hate happy birthdays and fluffy clouds
- I love disaster movies

- I like milk
- I like dairy products

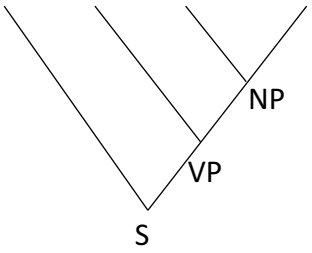
KEY POINT:

Language has hierarchical structure

Levels of language interpretation

words:	Mary	hires	a	detective	
parts of speech:	PN	VBZ	DET	CN	TAGGING
lemmata:	mary	hire	a	detective	STEMMING

syntax:



PARSING

semantics: $\exists$ x.detective(x) & hire(mary,x)

e,x	
hire(e)	detective(x)
subj(e,mary)	obj(e,x)

SEMANTIC PARSING

Beyond simple words

- Trying to capture the MEANING of words
- Recognize that texts are composed of PHRASES

Distributional semantics

- One way in which current NLP systems go beyond simple words is by attempting to model the MEANING of such words
- The most widely used approach to this is based on the principles of DISTRIBUTIONAL SEMANTICS:
  - "Thou shall know a word by the company it keeps" (Firth)

Distributional semantics

It is difficult to make a single, definitive description of the **folkloric** [redacted] though there are several elements common to many European **legends**. [redacted] were usually reported as bloated in appearance, and **ruddy**, **purplish**, or dark in colour; these characteristics were often attributed to the drinking of **blood**. [· · ·] Indeed, **blood** was often seen seeping from the mouth and nose of the [redacted] when it was seen in its **shroud** or **coffin** and its left eye was often open. [· · ·] In Christianity, the [redacted] was viewed as "a **dead** person who retained a semblance of life and could leave its **grave**-much in the same way that Jesus had risen after his **death** and **burial** and appeared before his followers. In Asia, [· · ·] a [redacted] wanders around animating **dead bodies** at night, attacking the living much like a **ghoul**.

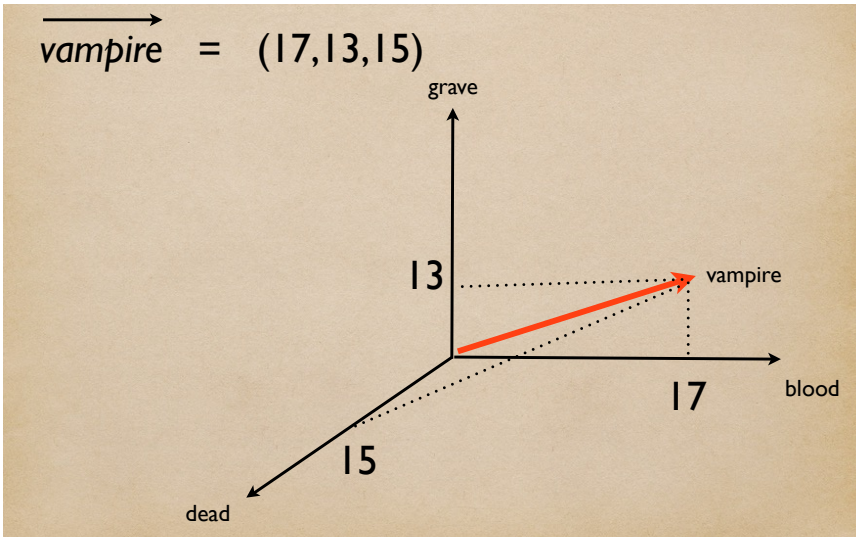
Distributional semantics

Butterflie are beautiful, flying insects with large scaly wings. Like all insects, they have six jointed legs, 3 body parts, a pair of antennae, compound eyes, and an exoskeleton. The three body parts are the head, thorax (the chest), and abdomen (the tail end). The butterfly's body is covered by tiny sensory hairs. The four wings and the six legs of the [illegible] are attached to the thorax. The thorax contains the muscles that make the legs and wings move. Butterflies are very good fliers. They have two pairs of large wings covered with colorful, iridescent scales in overlapping rows. Lepidoptera (butterflies and moths) are the only insects that have scaly wings. The wings are attached to the butterfly's thorax (mid-section). Veins support the delicate wings and nourish them with blood.

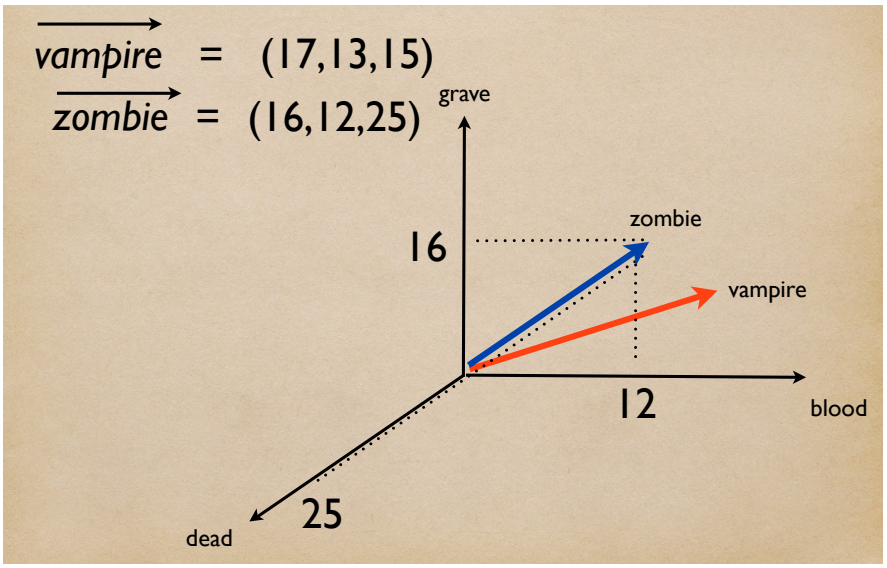
It is difficult to make a single, definitive description of the **folkloric** **vampire**, though there are several elements common to many European **legends**. **Vampire** were usually reported as bloated in appearance, and **ruddy**, **purplish**, or dark in colour; these characteristics were often attributed to the drinking of **blood**. [· · ·] Indeed, **blood** was often seen seeping from the mouth and nose of the **vampire** when it was seen in its **shroud** or **coffin** and its left eye was often open. [· · ·] In Christianity, the **vampire** was viewed as "a **dead** person who retained a semblance of life and could leave its **grave**-much in the same way that Jesus had risen after his **death** and **burial** and appeared before his followers. In Asia, [· · ·] a **vampire** wanders around animating **dead bodies** at night, attacking the living much like a **ghoul**.

Butterflie are beautiful, flying insects with large scaly wings. Like all insects, they have six jointed legs, 3 body parts, a pair of antennae, compound eyes, and an exoskeleton. The three body parts are the head, thorax (the chest), and abdomen (the tail end). The butterfly's body is covered by tiny sensory hairs. The four wings and the six legs of the butterfly are attached to the thorax. The thorax contains the muscles that make the legs and wings move. Butterflies are very good fliers. They have two pairs of large wings covered with colorful, iridescent scales in overlapping rows. Lepidoptera (butterflies and moths) are the only insects that have scaly wings. The wings are attached to the butterfly's thorax (mid-section). Veins support the delicate wings and nourish them with blood.

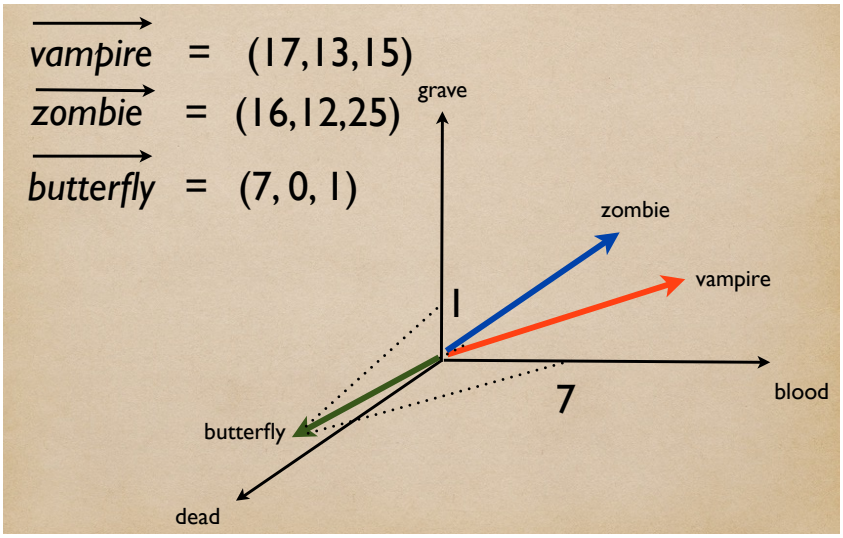
Words as vectors



Words as vectors

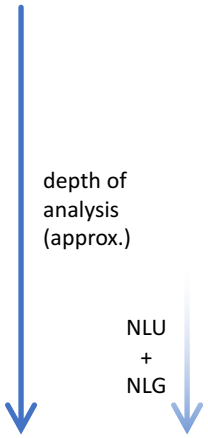


Words as vectors



What are the main applications of NLP?

- Some examples:
  - Search
  - Spam filtering
  - Document classification
  - Language modelling
  - Author identification
  - Sentiment analysis
  - Information extraction
  - Question answering
  - Machine Translation
  - Summarisation
  - Dialogue Systems



Contents of the module

- **Week 1**: outline, a simple NLTK example
- **Week 2**: statistical 1: classification/regression, ngram models
- **Week 3**: statistical 2: sequence models (HMMs, CRFs)
- **Week 4**: statistical 3 : topic models (latent variable models, LDA)
- **Week 5**: formal 1: syntax: generative and logical systems
- **Week 6**: formal 2 : semantics: lambda calculus and composition
- **Week 7**: *review*
- **Week 8**: formal 3: parsing algorithms and tools
- **Week 9**: adv dialogue & discourse
- **Week 10**: adv dialogue & discourse
- **Week 11**: adv lexical and distributional semantics
- **Week 12**: adv neural nets

Lectures and Labs

- LECTURES
  On Fridays, 1-3, Graduate Centre 2.01

- LABS
  On Mondays, 10-12, ITL 2F_Lab

Assessment

- COURSEWORK 40%
  - 3 Assignments
  - Weekly Lab projects

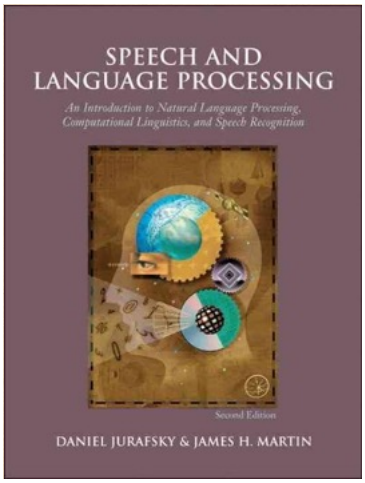
- EXAMINATION 60%
  - Covering both theory and practice

Readings

- Main text: a theoretically-oriented general reference on NLP
  - D. Jurafsky & J. Martin – *Speech and Language Processing*, 3rd edition – Prentice-Hall
- Practical intros to the topics of this module
  - Using Python:
    - Richert and Coelho - *Building Machine Learning Systems with Python (2nd ed)* - Pack Press (RC)
- A general intro to NLP with Python with a practical bend:
  - S. Bird, E. Klein & E. Loper, Natural Language Processing with Python, O'Reilly http://www.nltk.org/book/
- A more theoretical intro to some of the topics of the module:
  - C. Manning, P. Raghavan & H. Schuetze – *Introduction to Information retrieval* – Cambridge, 2008
  - http://nlp.stanford.edu/IR-book/

Reading material

- Main text:



- https://web.stanford.edu/~jurafsky/slp3/

Initial readings

- If you aren't familiar with Python / don't know much about language or corpora:
  - NLTK book, chapters 1 and 2