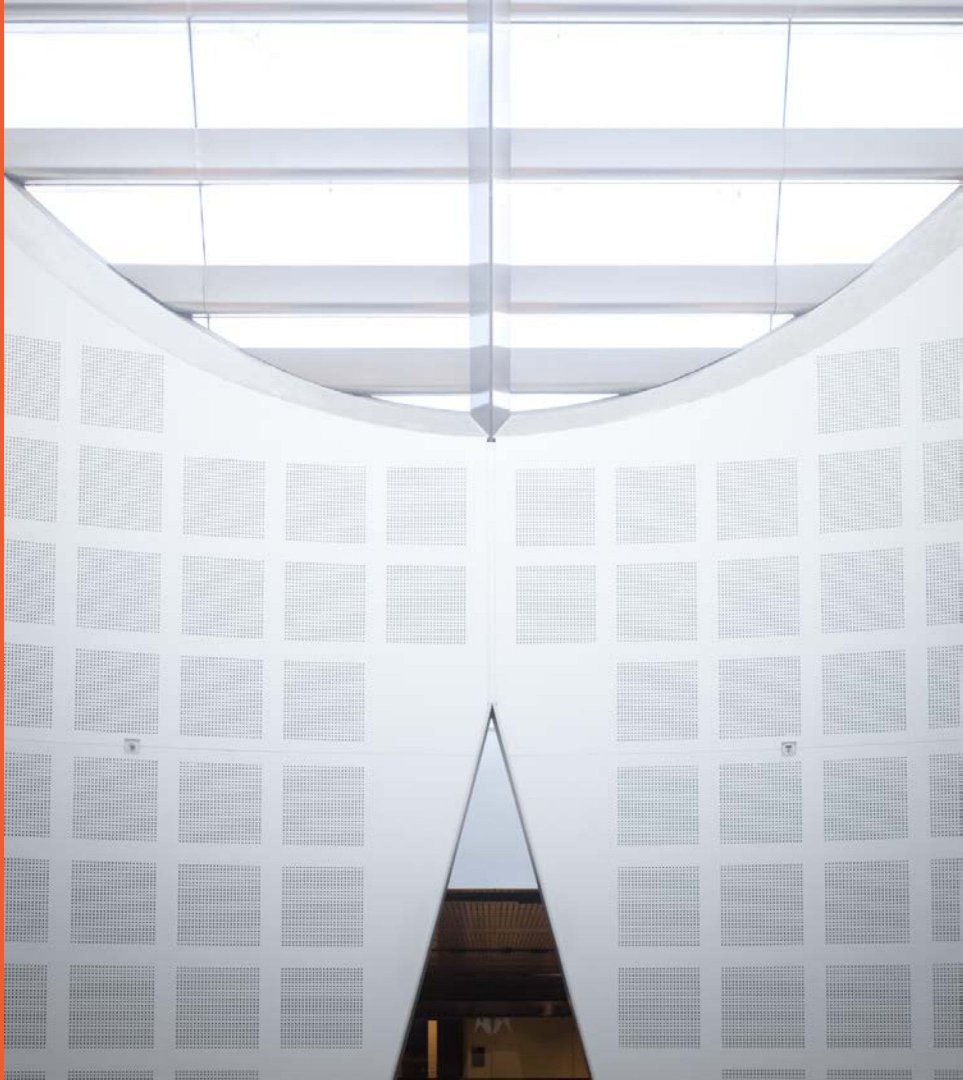
Big Data in Business

BUSS6002 week 11

Presented by

Dr Fabian Held

Fabian.Held@Sydney.edu.au



Natural Language Processing

Definition: NLP is the automated handling of natural human language like speech or text.

Goal: Make accessible for computers all the information that is stored in text (unstructured data) as opposed to data tables (structured data)

Challenge: Deal with the complexity that comes with human language

- Different languages
- Large, diverse vocabularies
- Words with multiple meanings
- Anaphora
- Accents and idioms,
- Irony, word play
- Errors…

Examples of Ambiguity

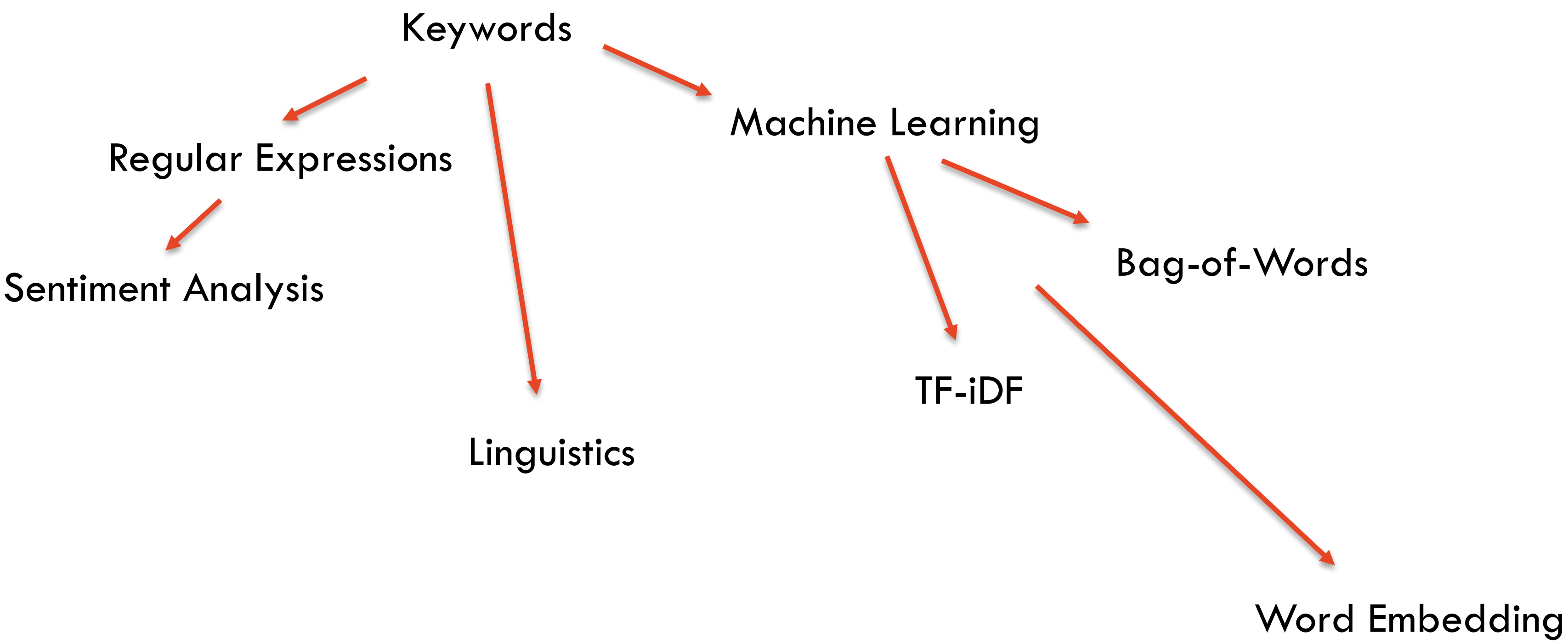
- **Lexical/morphological:**
  - change (V,N); training (V,N); even (ADJ, ADV) , present (N, V, ADJ, mean “time” or “gift”)
- **Semantic:**
  - He saw her duck.
  - She knows a little Greek.
- **Syntactic:**
  - The robber attacked the student with a book.
  - He put the ketchup on himself.
  - He watched her paint with enthusiasm.
- **Discourse:** anaphora
  - Margaret invited Susan for a visit, and she gave her a good lunch.
- **Pragmatic:**
  - The participation in this class is extraordinary.

https://cs.nyu.edu/faculty/davise/ai/ambiguity.html

Some NLP Applications

- Text as input to supervised learning:
  - Recommender systems in medicine and law
- Sentiment analysis
- Spam filters
- Identification of fake news
- Voice driven interfaces (Siri, Cortana, Alexa…)
- Financial trading based on news and social media
- Recruitment
- Chatbots
- …

Overview



Keywords

Regular Expressions

Sentiment Analysis

Linguistics

Machine Learning

Bag-of-Words

TF-iDF

Word Embedding

Regular Expressions

(not really NLP)

THE UNIVERSITY OF SYDNEY

Simple Search for Keywords

DON'T do text analytics with SQL

– I've seen complex queries like this with dozens of search terms:

```
SELECT * FROM customer_surveys WHERE survey_text LIKE "%angry%“…
```

– This is about as sophisticated as SQL gets with text!

– Don't ask me what I've seen in Excel…

– Basic functionality in Python

Simple Search for Keywords
How many paragraphs contain the word “late”?

- Chocolate is a typically sweet, usually brown, food preparation of roasted and ground cacao seeds. It is made in the form of a liquid, paste, or in a block, or used as a flavoring ingredient in other foods. The earliest evidence of use traces to the Olmecs (Mexico), with evidence of chocolate beverages dating to 1900 BC.[1][2] The majority of Mesoamerican people made chocolate beverages, including the Maya and Aztecs.[3]
- The seeds of the cacao tree have an intense bitter taste and must be fermented to develop the flavor. After fermentation, the beans are dried, cleaned, and roasted. The shell is removed to produce cacao nibs, which are then ground to cocoa mass, unadulterated chocolate in rough form. Once the cocoa mass is liquefied by heating, it is called chocolate liquor. The liquor also may be cooled and processed into its two components: cocoa solids and cocoa butter.
- Baking chocolate, also called bitter chocolate, contains cocoa solids and cocoa butter in varying proportions, without any added sugar. Much of the chocolate consumed today is in the form of sweet chocolate, a combination of cocoa solids, cocoa butter or added vegetable oils, and sugar. Milk chocolate is sweet chocolate that additionally contains milk powder or condensed milk. White chocolate contains cocoa butter, sugar, and milk, but no cocoa solids.
- Although cocoa originated in the Americas, recent years have seen African nations assuming a leading role in producing cocoa. Since the 2000s, Western Africa produces almost two-thirds of the world's cocoa, with Ivory Coast growing almost half of that amount.

https://en.wikipedia.org/wiki/Chocolate

Simple Search for Keywords
How many paragraphs contain the word “late”?

- Choco**late** is a typically sweet, usually brown, food preparation of roasted and ground cacao seeds. It is made in the form of a liquid, paste, or in a block, or used as a flavoring ingredient in other foods. The earliest evidence of use traces to the Olmecs (Mexico), with evidence of choco**late** beverages dating to 1900 BC.[1][2] The majority of Mesoamerican people made choco**late** beverages, including the Maya and Aztecs.[3]
- The seeds of the cacao tree have an intense bitter taste and must be fermented to develop the flavor. After fermentation, the beans are dried, cleaned, and roasted. The shell is removed to produce cacao nibs, which are then ground to cocoa mass, unadulterated choco**late** in rough form. Once the cocoa mass is liquefied by heating, it is called chocolate liquor. The liquor also may be cooled and processed into its two components: cocoa solids and cocoa butter.
- Baking choco**late**, also called bitter choco**late**, contains cocoa solids and cocoa butter in varying proportions, without any added sugar. Much of the choco**late** consumed today is in the form of sweet choco**late**, a combination of cocoa solids, cocoa butter or added vegetable oils, and sugar. Milk choco**late** is sweet choco**late** that additionally contains milk powder or condensed milk. White chocolate contains cocoa butter, sugar, and milk, but no cocoa solids.
- Although cocoa originated in the Americas, recent years have seen African nations assuming a leading role in producing cocoa. Since the 2000s, Western Africa produces almost two-thirds of the world's cocoa, with Ivory Coast growing almost half of that amount.

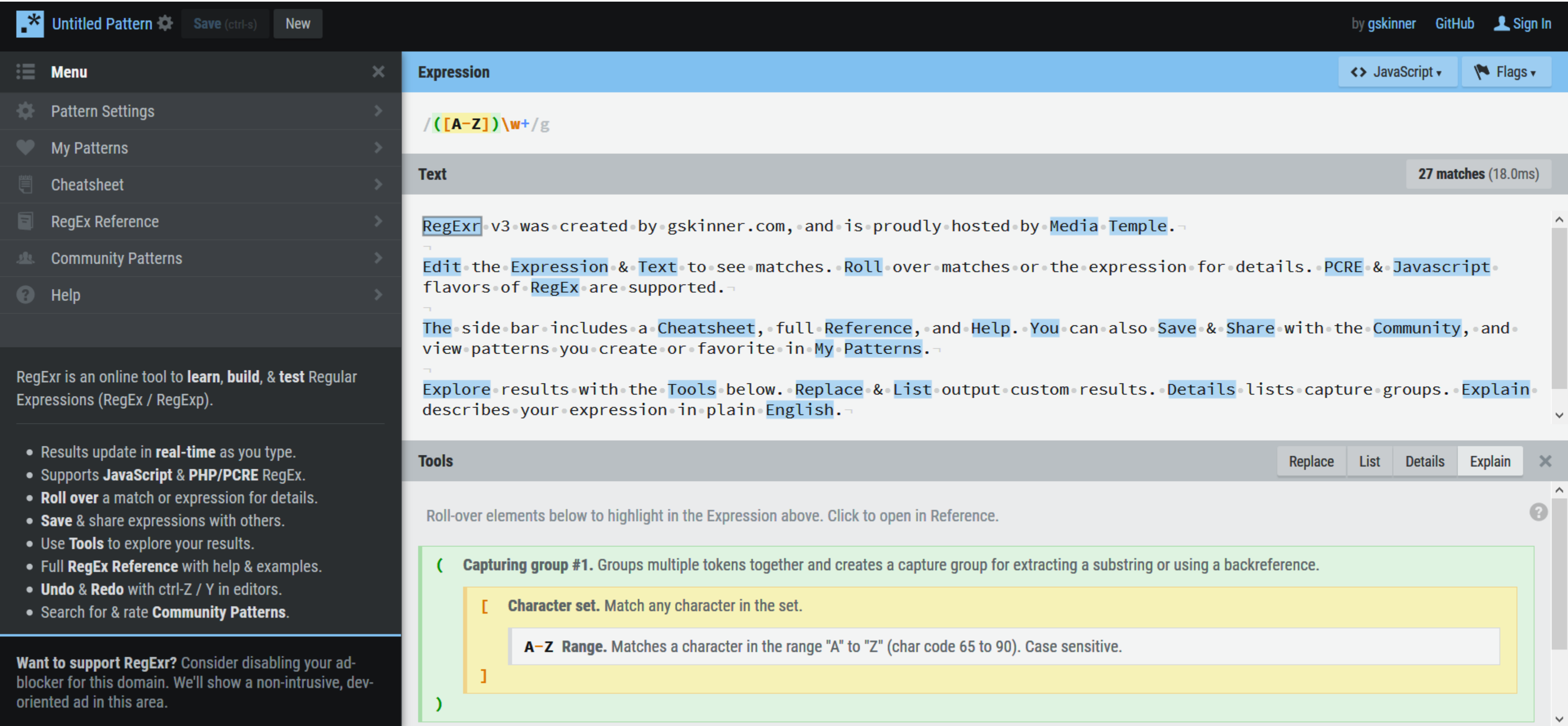
https://en.wikipedia.org/wiki/Chocolate

Regular Expressions (Regex)

- Can represent highly complex deterministic queries.
- See eg http://www.regexr.com/
- Is very difficult to learn/interpret (easiest if you can see results in real time)
- Simple regex searching for "happy" preceded by a space (to avoid "unhappy")

```
'.*[ ]happy.*'
```

http://www.regexr.com/



```
/([A-Z])\w+/g
```

```
RegExr v3 was created by gskinner.com, and is proudly hosted by Media Temple.

Edit the Expression & Text to see matches. Roll over matches or the expression for details. PCRE & Javascript flavors of RegEx are supported.

The side bar includes a Cheatsheet, full Reference, and Help. You can also Save & Share with the Community, and view patterns you create or favorite in My Patterns.

Explore results with the Tools below. Replace & List output custom results. Details lists capture groups. Explain describes your expression in plain English.
```

RegExr is an online tool to **learn**, **build**, & **test** Regular Expressions (RegEx / RegExp).

- Results update in **real-time** as you type.
- Supports **JavaScript** & **PHP/PCRE** RegEx.
- **Roll over** a match or expression for details.
- **Save** & share expressions with others.
- Use **Tools** to explore your results.
- Full **RegEx Reference** with help & examples.
- **Undo** & **Redo** with ctrl-Z / Y in editors.
- Search for & rate **Community Patterns**.

Want to support RegExr? Consider disabling your ad-blocker for this domain. We'll show a non-intrusive, dev-oriented ad in this area.

Roll-over elements below to highlight in the Expression above. Click to open in Reference.

(**Capturing group #1.** Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.

[**Character set.** Match any character in the set.

A-Z **Range.** Matches a character in the range "A" to "Z" (char code 65 to 90). Case sensitive.

]

)

Sophisticated Search for Keywords

Sentiment Analysis on Twitter

VADER (Valence Aware Dictionary and sEntiment Reasoner)

- A lexicon and rule-based sentiment analysis tool that is specifically to social media (Twitter).
- Typically assigns positive/negative points for each word, then adds them up.
- VADER is specifically tuned for Twitter (which is a very distinctive data set), so we're pretty safe here...
- VADER has approx. 600 lines of code, and a "lexicon" (dictionary) of over 7,000 terms.
- https://github.com/cjhutto/vaderSentiment

- Be very careful: If a meal in a restaurant is "to die for", that's a good thing!

Typical Use Cases for VADER Sentiment Analysis

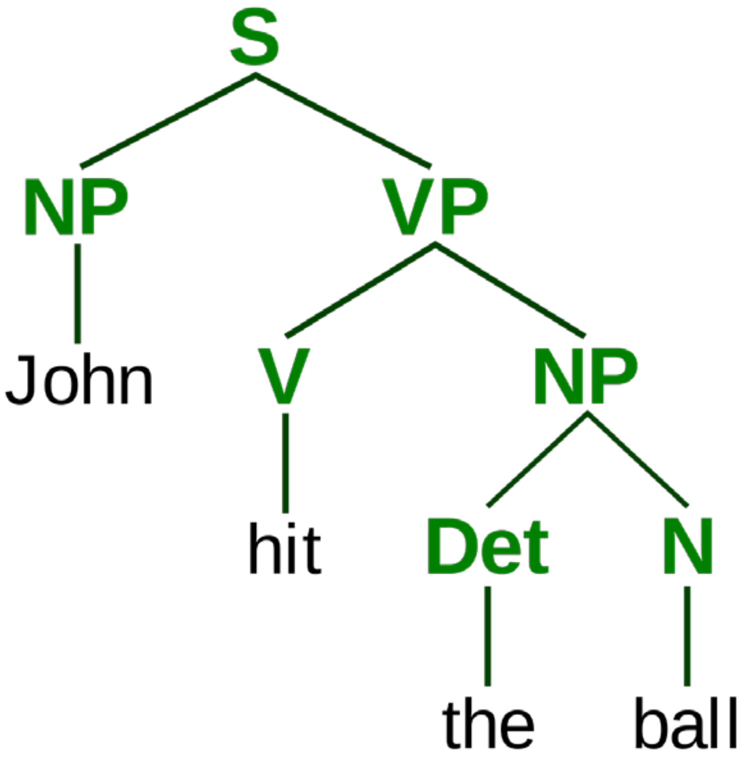
- Typical negations (e.g., "**not** good")
- Contractions as negations (e.g., "was**n't** very good")
- Conventional use of punctuation to signal increased sentiment intensity (e.g., "Good**!!!**")
- Conventional use of word-shape to signal emphasis (e.g., using **ALL CAPS** for words/phrases)
- Degree modifiers to alter sentiment intensity (e.g., intensity boosters such as "**very**" and intensity dampeners such as "**kind of**")
- Sentiment-laden slang words (e.g., **'sux**’)
- Sentiment-laden slang words as modifiers such as **'uber**' or **'friggin**' or **'kinda**’
- Sentiment-laden emoticons such as :) and **:D**
- Translating utf-8 encoded **emojis** such as 💘 and 💋 and 😁
- Sentiment-laden initialisms and acronyms (for example: **'lol**')

Linguistic Approaches

Parts of Speech and Semantics

- Parts of Speech
  - Nouns, verbs, adjectives, adverbs, pronouns…
  - Lots of words with different uses and meanings, so a dictionary alone will not help, e.g. “present”
- Phrase Structure Rules
  - E.g. “noun phrase” is often followed by “verb phrase”
- Parse tree:
  Identify parts of speech that form
  phrase structures that form
  sentences.

A simple parse tree



Combinations helps relieve ambiguity

Use a later step to solve ambiguity of an earlier step

- He gives him the change.
  - change as verb does not work for parsing.
- He changes the place.
  - change as noun does not work for parsing.

Good enough for simple text and interactions. Don’t get too fancy.

Some success story: Chatbot ELIZA (1966)

young woman: Men are all alike.
eliza: In what way?
young woman: They're always bugging us about something specific or other.
eliza: Can you think of a specific example?
young woman: Well, my boyfriend made me come here.
eliza: Your boyfriend made you come here?
young woman: He says I'm depressed much of the time.
eliza: I am sorry to hear you are depressed.
young woman: It's true. I'm unhappy.
eliza: Do you think coming here will help you not to be unhappy?
young woman: I need some help, that much seems certain.
eliza: What would it mean to you if you got some help?
young woman: Perhaps I could learn to get along with my mother.
eliza: Tell me more about your family.
young woman: My mother takes care of me.

Weizenbaum, Joseph (1966). "ELIZA—a computer program for the study of natural language communication between man and machine". Communications of the ACM. ***9****: 36–45. doi:10.1145/365153.365168.*

ELIZA – Cont'd

eliza: Who else in your family takes care of you?

young woman: My father.

eliza: Your father.

young woman: You are like my father in some ways.

eliza: What resemblance do you see?

young woman: You are not very aggressive but I think you don't want me to notice that.

eliza: What makes you think I am not very aggressive?

young woman: You don't argue with me.

eliza: Why do you think I don't argue with you?

young woman: You are afraid of me.

eliza: Does it please you to believe I am afraid of you?

young woman: My father is afraid of everybody.

eliza: What else comes to mind when you think of your father?

young woman: Bullies.

eliza: Does that have anything to do with the fact that your boyfriend made you come here?

*Weizenbaum, Joseph (1966). "ELIZA—a computer program for the study of natural language communication between man and machine". Communications of the ACM. **9**: 36–45. doi:10.1145/365153.365168.*

Achievements in ELIZA

- Identification of critical words
- Discovery of a minimal context
- Choice of appropriate transformations
- Generation of responses appropriate to the transformation or in the absence of critical words
- Provision of an ending capacity for ELIZA scripts

Methods used in Eliza

- Find a trigger in the input of the user (e.g. father)
- Evoke a possible candidate pattern (e.g. family or mother) (~limited parsing)
- Compose a sentence by filling in the slots of the pattern (picking some elements from the user input)
- If no appropriate pattern is found, ask a general question, possibly related to the user input

- **But:** Not flexible/generalisable, requires lots of manual rules

Model Based Text Analytics

The biggest weaknesses of the rules based approaches are:

- While easier for simple cases, increasingly convoluted logic is required to improve performance, or adapt to a new data set.
- Language changes over time, and in different contexts. Rules based approaches must be hand-tuned to account for this.
- The internet age leaves us with enormous amounts of text data. Rules based approaches cannot fully harness the embedded information.

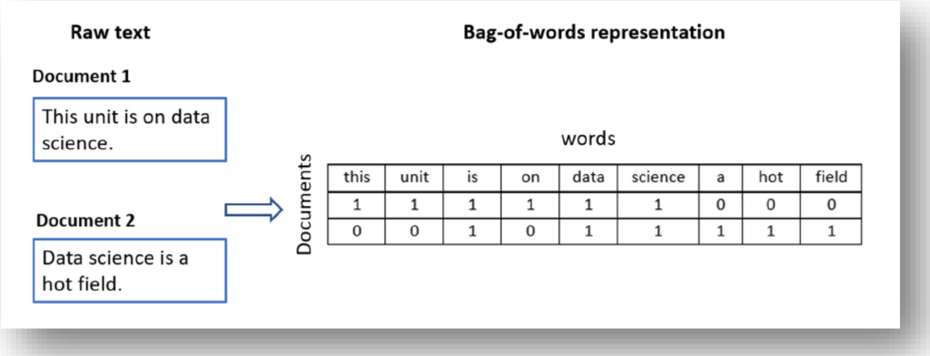
Structuring Unstructured Data

(for Machine Learning)

THE UNIVERSITY OF SYDNEY

From Week 4: Bag-of-Words

- Bag-of-words, or BoW is a popular way of extracting features from text that describes the occurrence of words within a document.
  - A vocabulary of known words.
  - A measure of the presence of known words.
- It is called a “bag” of words, because it does not capture the order of words in a document.



Raw text

Document 1

This unit is on data science.

Document 2

Data science is a hot field.

Bag-of-words representation

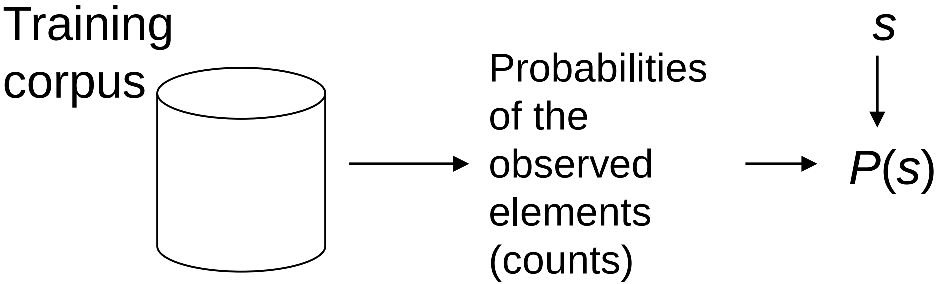
words

Documents

this	unit	is	on	data	science	a	hot	field
1	1	1	1	1	1	0	0	0
0	0	1	0	1	1	1	1	1

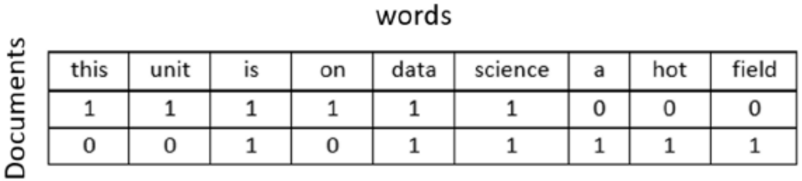
Goal: Language modeling

- Goal: create a statistical/machine learning model so that one can calculate the probability of a sequence of words $s = w_1, w_2, \ldots, w_n$ in a language.
- General approach:



Describe a Text as a Matrix of “Keyword” Frequencies

- Keywords ~ “Tokens”
- A *token* is an instance of a sequence of characters in some particular document that are grouped together as a *useful semantic unit* for processing.
- In English a token is most often identified by whitespace and/or punctuation around it.
  - But consider e.g. O’Neill, Abercrombie Street, San Francisco, faux pas



words

Documents

this	unit	is	on	data	science	a	hot	field
1	1	1	1	1	1	0	0	0
0	0	1	0	1	1	1	1	1

Using probability in language modelling

Probabilities computed in the context of corpora

1. P(“The sun rises in the East”.)
2. P(“The sun rise in the East”.)
   - Less probable because of grammatical error.
3. P(“The svn rises in the East”.)
   - Less probable because of lexical error.
4. P(“The sun rises in the West”.)
   - Less probable because of semantic error.

n-Gram representation

- An n-gram is a contiguous sequence of n tokens from a given sample of text.
  - An n-gram of size 1 is referred to as a "unigram";
  - Size 2 is a "bigram" (or, less commonly, a "digram");
  - Size 3 is a "trigram".
  - English cardinal numbers are sometimes used, e.g., "four-gram", "five-gram", and so on.
- This maintains some of the structure and possibly meaning of the original text.
  - More so than bag-of-words

Calculate probability (4-gram)

- P(“sun rises in the East”.) =
  P(sun) * P(rises | sun) * P(in | sun, rises) *
  P(the | sun, rises, in) * P(east | rises, in, the)

- #(rises | sun) > #(rise | sun)
- #(sun) > #(svn)
- #(east | rises, in, the) > #(west | rises, in, the)

Built on a huge corpus it can look like a system can learn grammar, semantics and spelling, simply because it can count frequencies.

Supervised NLP Models

Idea: Train supervised models on frequencies of tokens on n-grams

Challenges:

– The models will have a **very high dimensionality** (lots of features), depending on the number of unique tokens.

– The data sets are often highly unbalanced (e.g. positive tweets might be very rare!)

Getting rid of some of the Complexity

- ***Stemming*** usually refers to a heuristic process that removes the ends of words with the goal maintaining and unifying the core meaning of tokens.
- ***Lemmatization*** refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*.
  - Words in third person are changed to first person and verbs in past and future tenses are changed into present.

NLTK (Natural Language Toolkit)

Stemming is the common processing step in rules-based text analysis.

NLTK algorithms are quite complex, and have varying degrees of "aggressiveness".

- These are not statistical/learned algorithms...
- Porter 2 (a.k.a. Snowball English Stemmer) is generally the best choice for English.

Stop words

- Filter out some words that tend to add little to the meaning of a text.
- "stop words" usually refers to the most common words in a language
- No single universal list of stop words used by all natural language processing tools, and indeed not all tools even use such a list.
- Examples: https://www.ranks.nl/stopwords

TF-iDF (term frequency–inverse document frequency)

- Count frequency of n-gram per “document”
- Calculate average n-gram count across corpus
- Divide count / average count
- Translate counts into a measure of how “important” (i.e. unique) an n-gram is to a document in a collection or corpus.
- TF–iDF values increase proportionally to the number of times a word appears in the document and is offset by the number of times it appears across documents.

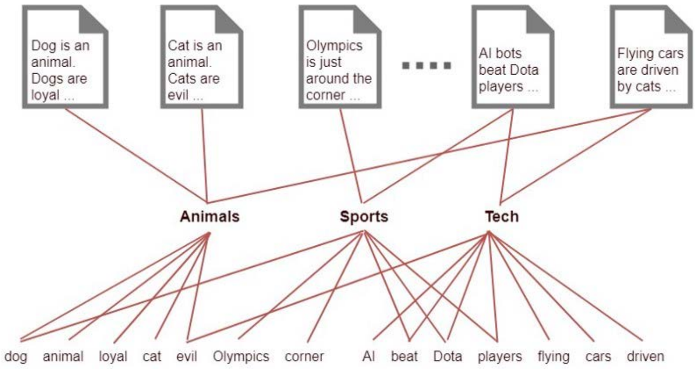
State of the Art

- Sufficiently large training data
  - The longer is n (n-gram), the lower is perplexity (harder to estimate probabilities)
- In many NLP researchers, one uses 5-grams or 6-grams
- Google books n-gram (up to 5-grams) https://books.google.com/ngrams

Topic Modelling

Latent Dirichlet Allocation

- Unsupervised machine learning
- Define k, the number of topics to identify
- Like k-means, algorithm groups documents
- Unlike k-means, a document can have multiple topics (mixtures)


Dog is an animal. Dogs are loyal ...
Cat is an animal. Cats are evil ...
Olympics is just around the corner ...
....
AI bots beat Dota players ...
Flying cars are driven by cats ...
Animals
Sports
Tech
dog
animal
loyal
cat
evil
Olympics
corner
AI
beat
Dota
players
flying
cars
driven

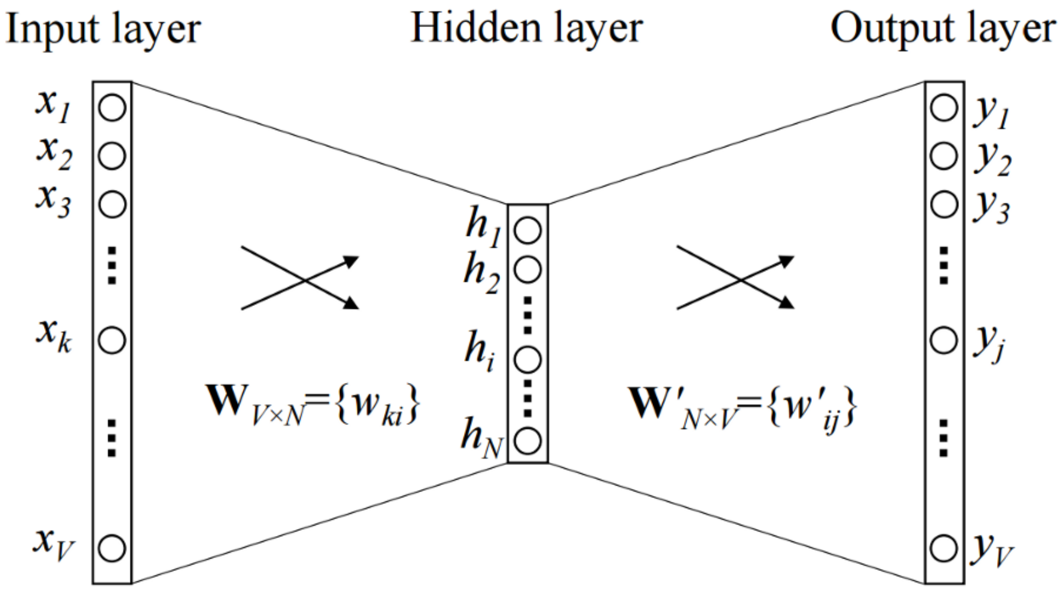
https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158

Word Embedding

Word embedding

- **Problem:** 0-1 vector representation of words looses a lot of information
  - Dog vs poodle
  - Dog vs cat
  - Dog vs "micro-chip-manufacturer"
- **Idea:** Represent words as vectors in multidimensional space
  - "similar" words should be closer
- **Solution:** Word2Vec uses shallow neural networks to calculate these vectors. It was developed by Tomas Mikolov in 2013 at Google.
- Word2Vec captures the context of a word in a document, semantic and syntactic similarity, relation with other words, etc.

Word2Vec



https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

Model Based Text Analytics:
What if we could do maths with those vectors?

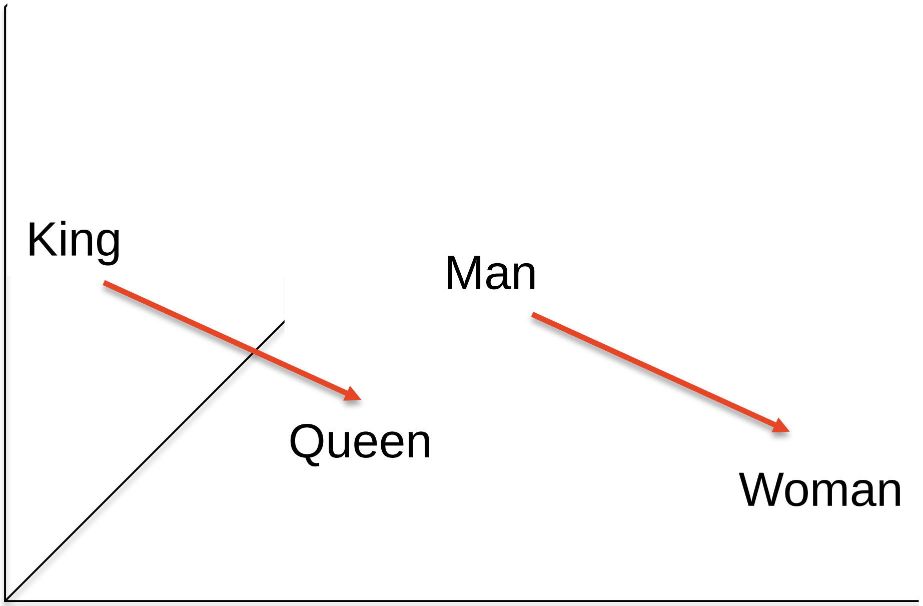
The relation between the concepts

"King" and "Queen"

is the same as between

"Man" and "Woman"

In the embedded space this means that the vectors between both pairs are (almost) parallel.



Summary

- Rule based systems rely on human decision making and manual input
- Traditional machine learning relies on human engineered features
- No human intervention necessary for Deep Learning and accuracy may improve also.

Resources

THE UNIVERSITY OF SYDNEY

https://www.youtube.com/watch?v=fOvTtapxa9c