Real World Applications of Data Science

In partnership with:

Proscia Inc, Betamore, Spark B-more

Lecture 5: Natural Language Processing + Filtering

Natural Language Processing

Q: What is natural language processing?

NLP

Q: What is natural language processing?

A: Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding -- that is, enabling computers to derive meaning from human or natural language input.

NLP

Q: What are word boundaries?

NLP

Q: What are word boundaries?

A: The task of defining what constitutes a "word" involves determining where one word ends and another word begins. In other words, identifying word boundaries.

NLP

Q: What is Tokenization?

NLP

Q: What is Tokenization?

A: Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis.

NLP

Q: What is stemming?

NLP

Q: What is stemming?

A: In linguistic morphology and information retrieval, stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; *it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root.* Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation. [Example]

NLP

Q: What is tf-idf?

NLP

Q: What is tf-idf?

A: tf–idf, term frequency–inverse document frequency, is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

NLP

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification. One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

NLP

Q: What is cosine similarity?

NLP

Q: What is cosine similarity?

A: Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgement of orientation and not magnitude: two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].

NLP

Note that these bounds apply for any number of dimensions, and Cosine similarity is most commonly used in high-dimensional positive spaces. For example, in Information Retrieval and text mining, each term is notionally assigned a different dimension and a document is characterized by a vector where the value of each dimension corresponds to the number of times that term appears in the document. Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter. The technique is also used to measure cohesion within clusters in the field of data mining.

NLP

Given two vectors of attributes, A and B, the cosine similarity , cos(theta) is the dot product of A and B over the product of the magnitudes of A and B:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity.

For text matching, the attribute vectors *A* and *B* are usually the term frequency vectors of the documents. The cosine similarity can be seen as a method of normalizing document length during comparison.

In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1, since the term frequencies (tf-idf weights) cannot be negative. The angle between two term frequency vectors cannot be greater than 90°.

NLP

Q: What is latent semantic analysis?

NLP

Q: What is latent semantic analysis?

A: Latent semantic analysis (LSA) is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD) is used to reduce the number of columns while preserving the similarity structure among rows. Words are then compared by taking the cosine of the angle between the two vectors formed by any two rows. Values close to 1 represent very similar words while values close to 0 represent very dissimilar words.

NLP

Q: What is word sense disambiguation?

NLP

Q: What is word sense disambiguation?

A: In computational linguistics, word-sense disambiguation (WSD) is an open problem of natural language processing, which governs the process of identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings. The solution to this problem impacts other computer-related writing, such as discourse, improving relevance of search engines, anaphora resolution, coherence, inference *et cetera*.

Research has progressed steadily to the point where WSD systems achieve sufficiently high levels of accuracy on a variety of word types and ambiguities. A rich variety of techniques have been researched, from dictionary-based methods that use the knowledge encoded in lexical resources, to supervised machine learning methods in which a classifier is trained for each distinct word on a corpus of manually sense-annotated examples, to completely unsupervised methods that cluster occurrences of words, thereby inducing word senses. Among these, supervised learning approaches have been the most successful algorithms to date.

NLP

Q: What are topic models?

A: In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently: "dog" and "bone" will appear more often in documents about dogs, "cat" and "meow" will appear in documents about cats, and "the" and "is" will appear equally in both. A document typically concerns multiple topics in different proportions; thus, in a document that is 10% about cats and 90% about dogs, there would probably be about 9 times more dog words than cat words. *A topic model captures this intuition in a mathematical framework, which allows examining a set of documents and discovering, based on the statistics of the words in each, what the topics might be and what each document's balance of topics is*. [Example]

Although topic models were first described and implemented in the context of natural language processing, they have applications in other fields such as bioinformatics.

NLP

Q: What are parts of speech?

A: In grammar, a part of speech (also a word class, a lexical class, or a lexical category) is a linguistic category of words (or more precisely *lexical items*), which is generally defined by the syntactic or morphological behaviour of the lexical item in question. Common linguistic categories include *noun* and *verb*, among others. There are open word classes, which constantly acquire new members, and closed word classes, which acquire new members infrequently if at all.

Almost all languages have the lexical categories noun and verb, but beyond these there are significant variations in different languages. For example, Japanese has as many as three classes of adjectives where English has one; Chinese, Korean and Japanese have nominal classifiers whereas European languages do not; many languages do not have a distinction between adjectives and adverbs, adjectives and verbs (see stative verbs) or adjectives and nouns, etc. This variation in the number of categories and their identifying properties entails that analysis be done for each individual language. Nevertheless the labels for each category are assigned on the basis of universal criteria.

NLP

Q: What is the Flesch-Kincaid test?

A: The Flesch/Flesch–Kincaid readability tests are readability tests designed to indicate comprehension difficulty when reading a passage of contemporary academic English. There are two tests, the Flesch Reading Ease, and the Flesch–Kincaid Grade Level. Although they use the same core measures (word length and sentence length), they have different weighting factors. The results of the two tests correlate approximately inversely: a text with a comparatively high score on the Reading Ease test should have a lower score on the Grade Level test. Rudolf Flesch devised both systems while J. Peter Kincaid developed the latter for the United States Navy. Such readability tests suggest that many Wikipedia articles may be "too sophisticated" for their readers.

NLP

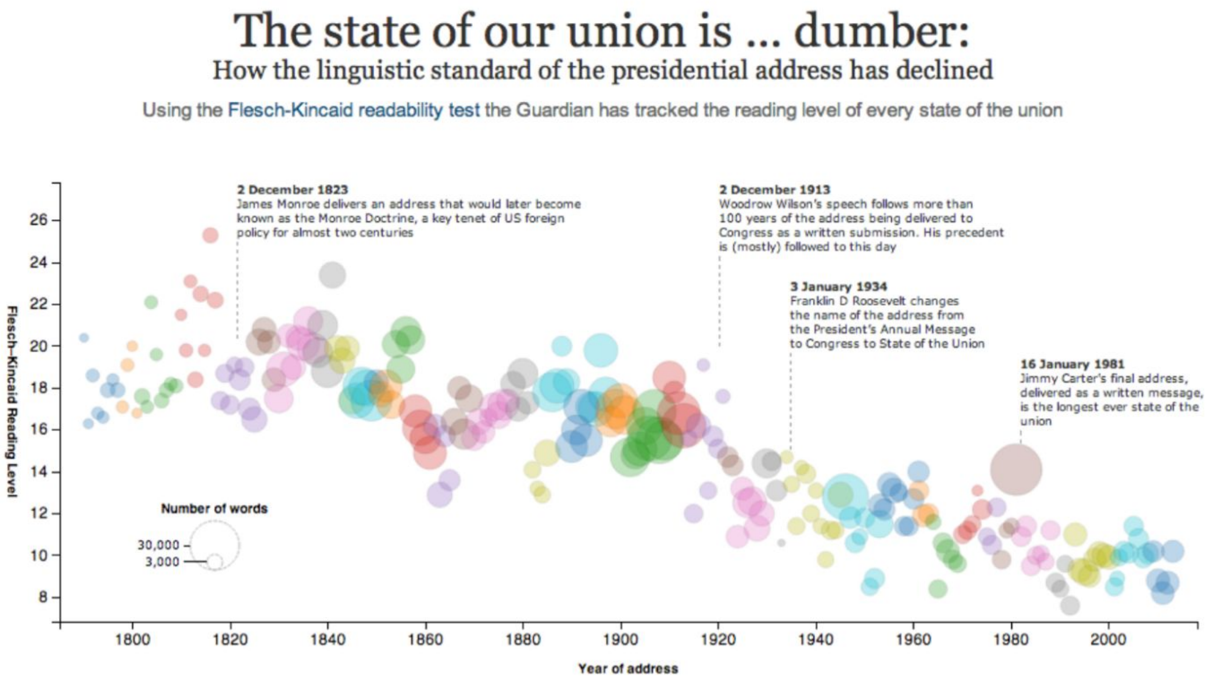
In the Flesch Reading Ease test, higher scores indicate material that is easier to read; lower numbers mark passages that are more difficult to read. The formula for the Flesch Reading Ease Score (FRES) test is

$$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$ [8]

Scores can be interpreted as shown in the table below.

Score	Notes
90.0–100.0	easily understood by an average 11-year-old student
60.0–70.0	easily understood by 13- to 15-year-old students
0.0–30.0	best understood by university graduates

NLP



The state of our union is ... dumber:
How the linguistic standard of the presidential address has declined
Using the Flesch-Kincaid readability test the Guardian has tracked the reading level of every state of the union
2 December 1823
James Monroe delivers an address that would later become known as the Monroe Doctrine, a key tenet of US foreign policy for almost two centuries
2 December 1913
Woodrow Wilson's speech follows more than 100 years of the address being delivered to Congress as a written submission. His precedent is (mostly) followed to this day
3 January 1934
Franklin D Roosevelt changes the name of the address from the President's Annual Message to Congress to State of the Union
16 January 1981
Jimmy Carter's final address, delivered as a written message, is the longest ever state of the union
Number of words
30,000
3,000
Flesch–Kincaid Reading Level
26
24
22
20
18
16
14
12
10
8
1800
1820
1840
1860
1880
1900
1920
1940
1960
1980
2000
Year of address

NLP

Q: Latent Dirichlet Allocation (LDA) ?

A: In natural language processing, Latent Dirichlet Allocation (LDA) is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. LDA is an example of a topic model and was first presented as a graphical model for topic discovery by David Blei, Andrew Ng, and Michael Jordan in 2003.

NLP

Q: Sentiment Analysis?

A: Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials.

Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation (see appraisal theory), affective state (that is to say, the emotional state of the author when writing), or the intended emotional communication (that is to say, the emotional effect the author wishes to have on the reader).

Filtering

Recommendations

A recommendation system aims to match users to products/items/brand/etc that they likely haven’t experienced yet.

This rating is produced by analyzing other user/item ratings (and sometimes item characteristics) to provide personalized recommendations to users.

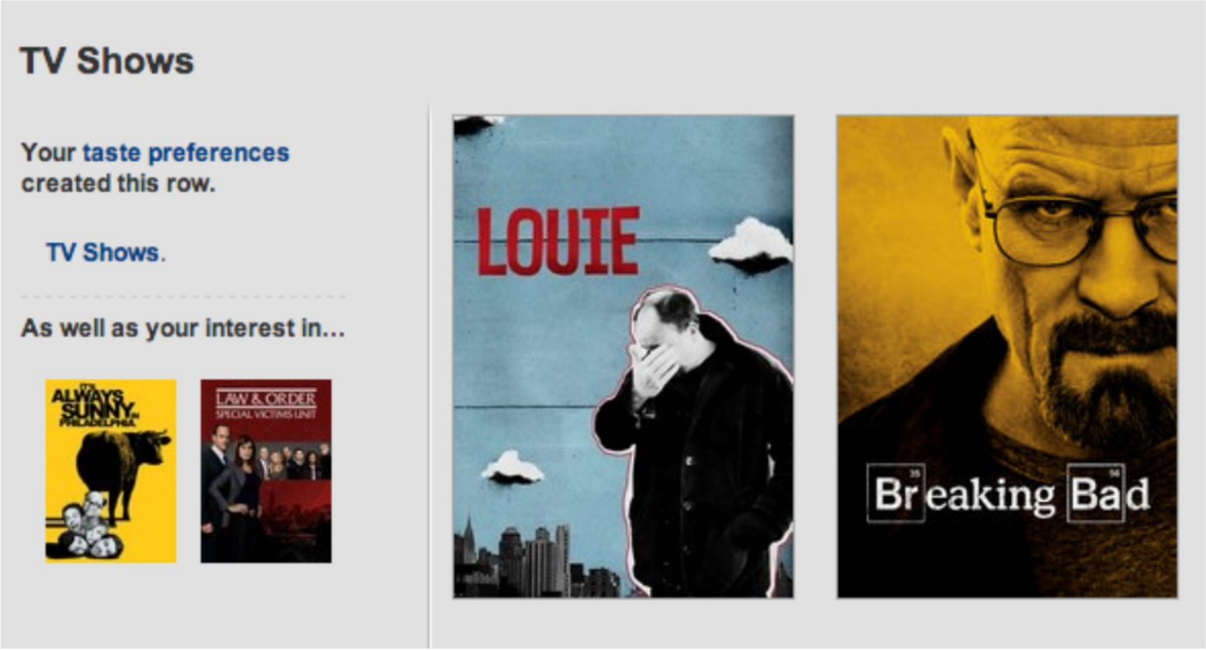
Filtering

There are two general approaches to the design:

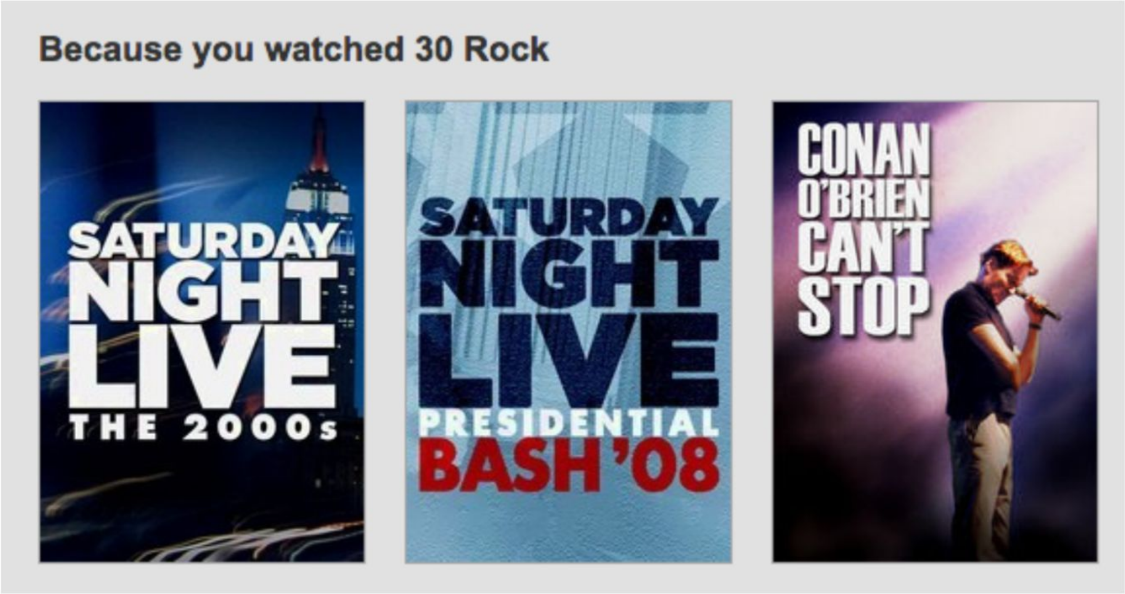
In **content-based filtering**, items are mapped into a feature space, and recommendations depend on *item characteristics*.

In contrast, the only data under consideration in **collaborative filtering** are user-item ratings, and recommendations depend on *user preferences*.

Examples


TV Shows
Your taste preferences created this row.
TV Shows.
As well as your interest in...
LOUIE
Breaking Bad

Examples



Because you watched 30 Rock
SATURDAY NIGHT LIVE
THE 2000s
SATURDAY NIGHT LIVE
PRESIDENTIAL BASH '08
CONAN O'BRIEN CAN'T STOP

Examples

- Manual Curation
- Manually Tag Attributes
- Audio Content, Metadata, Text Analysis
- Collaborative Filtering

content-based filtering

Content-based filtering

Content-based filtering begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.

Item vectors measure the degree to which the item is described by each feature, and ***user vectors*** measure a user's preferences for each feature.

Ratings are generated by taking **dot products** of user & item vectors.

Example: Content-based

features = (big box office, aimed at kids, famous actors)

Items (movies):	**Prediction (for Alice)**	
Finding Nemo = (5, 5, 2)	5*-3 + 5*2 + 2*-2	= -9
Mission Impossible = (3, -5, 5)	3*-3 + -5*2 + 5*-2	= -29
Jiro Dreams of Sushi = (-4, -5, -5)	-4*-3 + -5*2 + -5*-2	= +12

User:

Alice = (-3, 2, -2)

Example: Content-based

features = (big box office, aimed at kids, famous actors)

Items (movies):	**Prediction (for Bob)**	
Finding Nemo = (5, 5, 2)	5*4 + 5*-3 + 2*5	= +15
Mission Impossible = (3, -5, 5)	3*4 + -5*-3 + 5*5	= +52
Jiro Dreams of Sushi = (-4, -5, -5)	-4*4 + -5*-3 + -5*5	= -26

User:

Bob = (4, -3, 5)

Content-based

One notable example of content-based filtering is Pandora, which maps songs into a feature space using features (or “genes”) designed by the Music Genome Project.

Using song vectors that depend on these features, Pandora can create a station with music having similar properties to a song the user selects.

Artists


Aly And Aj
The Fray
Luke Bryan
Joe Brooks
Eric Church
Lady Antebellum
Zac Brown Band
Miranda Lambert
Josh Gracin
Selena Gomez
Avril Lavigne
Josh Turner
Sugarland
Dierks Bentley
Maroon 5
Big & Rich
David Archuleta
Taylor Swift
Colbie Caillat
Carrie Underwood
Trace Adkins
Justin Bieber
Blake Shelton
Sara Evans
Montgomery Gentry
The Band Perry
Reba Mcentire
Jack'S Mannequin
Lady Gaga
Martina Mcbride
The Wanted
Phil Vassar
Carly Rae Jepsen
Ariana Grande
Darius Rucker
Jason Reeves

Content-based

Content-based filtering has some difficulties:

- Must map items into a feature space (usually by hand!)
- Recommendations are limited in scope (items must be similar to each other)
- Hard to create cross-content recommendations (eg books/music films…this would require comparing elements from different feature spaces!)

Collaborative filtering

Collaborative filtering refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are *only* interested in the existing user-item ratings themselves.

In this case, our dataset is a *ratings matrix* whose columns correspond to items, and whose rows correspond to users.

NOTE

The idea here is that users get value from recommendations based on other users with similar *tastes*.

Ratings matrix

18,000 movies

480,000 users

x	1	1	x	...	x
x	x	x	5	...	x
x	x	3	x	...	x
x	4	3	x	...	2
...	x	x	x	...	x
x	5	x	1	...	x
x	x	3	3	...	x
x	1	x	x	...	2

NOTE

This matrix will always be *sparse*!

Content v. Collab

Main difference between content and collaborative filtering:

Content Based:
maps items and users into a feature space

Collaborative:
relies on previous user-item ratings

Collaborative filtering

We will look at collaborative filtering in a user-user sense.

We will take a given user, and find the K most similar users, and then recommend brands from the similar users!

NOTE

Sound familiar? It's similar to KNN!

Cold start problem

The cold start problem arises because we've been relying only on ratings data, or on explicit feedback from users.

Until users rate several items, we don't know anything about their preferences!

We can get around this by enhancing our recommendations using implicit feedback, which may include things like item browsing behavior, search patterns, purchase history, etc.

Cold start problem

While explicit feedback (ratings, likes, purchases) leads to high quality ratings, the data is sparse and cold starts are problematic.

Meanwhile implicit feedback (browsing behavior, etc.) leads to less accurate ratings, but the data is much more dense (and less invasive to collect).

Any questions?