www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Text Mining
In 2004 Ian Witten (Weka's "father") published a paper titled «Text Mining» in The Practical Handbook of Internet Computing.
In his paper Ian Witten reports that: "the first workshops [on text mining] were held at the International Machine Learning Conference in July 1999 and the International Joint Conference on Artificial Intelligence in August 1999".
"Text mining is a burgeoning new field that attempts to glean **meaningful information** from natural language text. It may be loosely characterized as the process of analyzing text to extract information that is useful for particular purposes."
With 'meaningful information' is **subjective**, because it depends by the reader, by the topics etc.
"the phrase "text mining" appears 17 times as often as "text data mining" on the Web, according to a popular search engine (and "data mining" occurs 500 times as often)."

Another definition:
"The phrase "text mining" is generally used to denote any system that analyzes large quantities of natural language text and *detects lexical or linguistic usage patterns in an attempt to extract probably useful* (although only probably correct) *information*" [Sebastiani, 2002].

Different Text Mining stuff:

- ⇨ Sentiment analysis:
  I may have the needed to analyze the sentiment about a post or a text in general.
- ⇨ Document summarization:
  Counting of words, obtaining a summary, so a synthesis, of the words in the text. We will have a short description of the text that is being analyzed.
  - Snippets: For example, in some search provider you can have a summary of the web page you are looking for.
- ⇨ Recommendation:
  - Content text filtering: based of the analyzes of the text the algorithm decides to present or not the text to the user;
  - Collaborative filtering algorithm: In a commercial website you will be recommended, for example films, or other websites. The system looks at the behavior of similar user to recommend something to the target users.
  - News recommendation: same, but for news.

  Of course, if we need to build a recommendation algorithm, we'll need to have a profiled user, so we will have a model of the user. Using cookies and also the historical usage of the web we do.
- ⇨ Text analytics in financial services
- ⇨ Text analytics in healthcare

Data mining can be more fully characterized as the extraction of implicit, previously unknown, and potentially useful information from data [Witten and Frank, 2000].

- The information is implicit in the input data: it is hidden, unknown, and could hardly be extracted without recourse to automatic techniques of data mining.

With **text mining**, the information to be extracted is clearly and explicitly stated in the text. It is not hidden at all.

- Text mining strives to bring it out of the text in a form that is suitable for consumption by computers directly, with no need for a human intermediary. The difficult part is to extract information automatically.

"Mining implies extracting precious nuggets of ore from otherwise worthless rock"
If data mining really followed this metaphor, it would mean that people were discovering new factoids within their inventory databases. However, in practice this is not really the case. Instead, data mining applications tend to be (semi)automated discovery of trends and patterns across very large datasets, usually for the purposes of decision making.

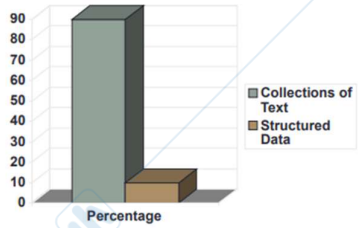
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

From: Marti A. Hearst. 1999. Untangling text data mining. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL '99).

	Finding Patterns	Finding Nuggets	
		Novel	Non-Novel
Non-textual data	standard data mining	?	database queries
Textual data	computational linguistics	real TDM	information retrieval

Table 1: A classification of data mining and text data mining applications.



Finding Nuggets => means for example finding something in particular (SQL) finding a score of a student.
Finding Patterns => statistical patterns that can be found in the texts.

In text mining we can have different tasks:

- *Text Summarization*: A text summarizer strives to produce a condensed representation of its input, intended for human consumption.
  In short words: is a summarization of a text; so a small part of the text.

- *Information Retrieval*: given a corpus of documents and a user's information need expressed by a query, IR is the task of identifying and returning the most relevant documents to the query. Web search engine also apply text summarization stage that focuses on the query posed by the user to provide a short synthesis of the retrieved documents.
  For example, the Search Engine are algorithm that find website based on words. In the case of news, we are looking to news. So, we can select the collection that we want to find.

- *Content Based Recommender Systems*: textual contents produced in a stream are pushed to a user to fulfill the user preferences as represented in a user model, also called user profile.

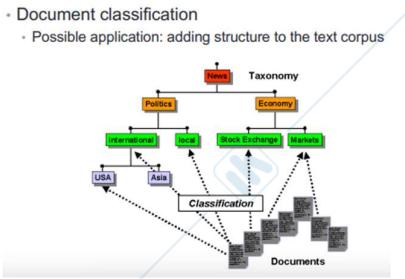
- *Text Classification:* Text classification (or text categorization) is the assignment of natural language documents to pre-defined categories according to their content [Sebastiani, 2002]. It has a variety of applications (e.g.sentiment analysis) hot topic in machine learning (supervised learning). Define based on predefined category, the document that belong to a specific category.

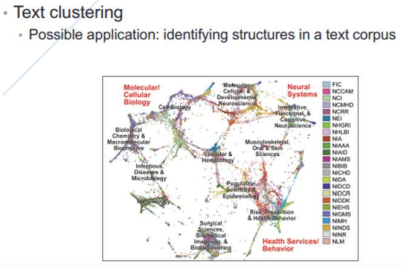
- *Document clustering*: document clustering is "unsupervised" learning in which there is no predefined category or "class," but groups of documents that share the similar topics are sought.
  Disclose group of documents that talk about same topic.

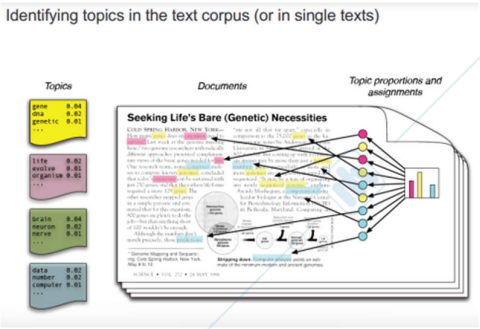
- Topic identification and tracking

Document: is a unit of information that can have other information (images, links, schemas).
If I have a huge amount of news, I want to organize the document in a topical structure, I may want to apply text classification. So, looking at the images at the right we can imagine classify it in that two categories. Classification and clustering are aimed to classify content by topic. In the case of clustering I don't need to classify the topics before using the algorithm, because it will associate the text looking for hidden structures. A problem can be associate the description of the context of text. How can I apply a short description? Maybe we can use text summarization.

Topic modeling =>



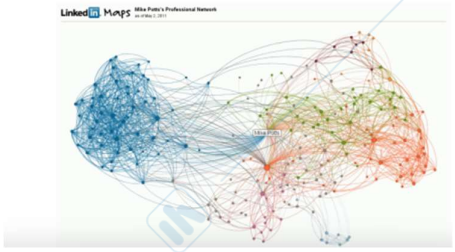




www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

- Exploring additional structure in the text corpus



Social Media Analytics =>
The web 2.0 give us the possibility to increase the view of the web thanks to the social media network and the user generated content; people are allowed to generate their own content without control.
Social media are textual description, and you can post your opinion. Huge amount of content generated by users to be mined. We can built predictive algorithm if a given product will be liked form a customer.

Mining structured data within texts.

- Entity extraction: many practical tasks involve identifying linguistic constructions that stand for objects or "entities" in the world (e.g. names of people, places, etc.).

- Information Extraction: the task of filling templates from natural language input.
  Filling templates, extract from document in natural languages, where there is not a clear structure of the text.
  I want to extract the name of the author, the date of publication and other information.

- Learning rules from texts: extracting rules that characterize the content of the text itself.
  Extract rules that the characterize the context of the text.
  - ⇨ Several Web resources have a structure: for example, Web pages are written is HTML. XML is another markup language that provides a "logical" structure to a text.
  - ⇨ Many software systems use external online resources by hand-coding simple parsing modules, commonly called "wrappers," to analyze the page structure and extract the requisite information.

TEXT PROCESSING

How can I formally represent a text in such a way that a machine is able to understand?

"Basic text processing is often the first step in any text mining application and consists of several layers of simple processing such as tokenization, lemmatization, normalization, or more advanced"
Tokenization: Isolate the words.
Lemmatizzation: reducing the words to the same route.
Normalization: reduce the entity in the same entity, different format of dates, names of cities etc.

"Great movie! It kept me on the edge of my seat the whole time. I IMAX-ed it and have no regrets." positive

"Waste of time! It sucked!" negative

"This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up." negative

"Trust me, this movie is a masterpiece .... after you've seen it 4+ times." ???

The purpose of analyzing texts can be either *predictive* or *exploratory:*

- *Predictive Analysis of Text:*
  - ⇨ developing computer programs that automatically recognize or detect a particular concept within a span of text. For example, polarity of text, so, if the text has a positive or negative meaning.
    - ➢ Opinion Mining:
      automatically detecting whether a span of opinionated text expresses a positive or negative opinion about the item being judged. For examples in a commercial site, you want to classify the review about a new product.
    - ➢ Sentiment/Affect Analysis
      automatically detecting the emotional state of the author of a span of text (usually from a set of pre-defined emotional states). Positive or negative thinking in a sentence.
    - ➢ Bias Detection
      automatically detecting whether the author of a span of text favors a particular viewpoint (usually from a set of pre-defined viewpoints).

- "[I] also found out that the radiologist is doing the biopsy, not a breast surgeon. I am more scared now than when I ..." fear
- "... My radiologist 'assured' me my scan was NOT going to be cancer...she was wrong." despair
- " ... My radiologist did my core biopsy. Not a problem and he did a super job of it." hope
- "It's pretty standard for the radiologist to do the biopsy so I wouldn't be concerned on that score." hope

- "Nationalizing businesses, nationalizing banks, is not a solution for the democratic party, it's the objective." -- Rush Limbaugh conservative (vs. liberal)

- "If you're keeping score at home, so far our war in Iraq has created a police state in that country and socialism in Spain. So, no democracies yet, but we're really getting close." -- Jon Stewart against war in iraq (vs. in favor of)

Relation Learning
CEO(<person>,<company>)

<person>, who was named CEO of <company>

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

- *Information Extraction*
  - automatically detecting that a short sequence of words belongs to (or is an instance of) a particular entity type, for example:
  - Person(X)
  - Location(X)
  - TennisPlayer(X)

- *Relation Learning*
  - automatically detecting pairs of entities that share a particular relation, for example:
  - CEO(<person>,<company>)
  - Capital(<city>,<country>)
  - Mother(<person>,<person>)
  - .....

➢ Text-driven Forecasting
monitoring incoming text (e.g., tweets) and making predictions about external, real- world events or trends
- a presidential candidate's poll rating
- a company's stock value change
- a movie's box office earnings
- side-effects for a particular drug

➢ Temporal Summarization
monitoring incoming text (e.g., tweets) about a news event and predicting whether a sentence should be included in an on-going summary of the event.
Extract the most important and salient aspect about a given topic or event.

- Exploratory Analysis of Text:
  ⇨ developing computer programs that automatically discover interesting and useful patterns or trends in text collections.
  ➢ Text clustering
  ➢ Topic modeling

TEXT AND DOCUMENTS

- Free-format text
  ⇨ EBCDIC coding, ASCII (8 bit), UNICODE (16 bit), etc.
  ➢ Language
  ⇨ Text consisting of strings of characters from an alphabet, etc.
  ➢ E.g., genome sequences, formulas of chemical compounds, words in natural language

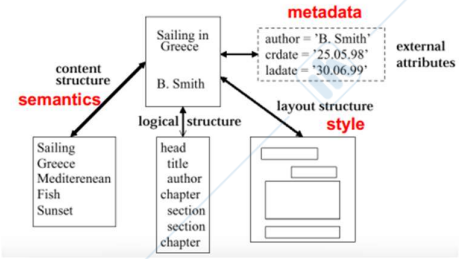
- Document
  - Text
  +
  - Structure
  +
  - Other media (images, sounds, …)
  +
  - Metadata

Examples:
Articles from newspapers, magazines, messages, letters, medical reports, Web pages, etc.

Metadata:

- Descriptive metadata:
  Relating to the creation of the document.
  E.g.,: title, authors, date, length (in pages, words, bytes, etc.), genre (book, article, memo, mail, etc.)

- Semantic metadata:
  Relating to the subject dealt with in the document.
  E.g.,: Library of Congress subject codes, controlled keywords extracted from an ontology



It's important to extract metadata for example for web search engines.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Metadata is information about information - labeling, cataloging and descriptive information structured in such a way that allows pages to be properly searched and processed by computer. In other words, what is now very much needed on the Web is metadata.

⇨ W3C's (World Wide Web Consortium) Metadata Activity is concerned with ways to model and encode metadata.

We need to transform our text in a such way that will be understandable from algorithms.

FORMATS FOR TEXT DOCUMENTS

In classical IR systems, documents had to be represented in an "internal" format in order to be indexed and managed.

⇨ 1st OPERATION: source file à generation or input file.
⇨ Documents written with common word-processors:
- Word
- TeX, RTF, HTML, XML (Rich Text Format) (ASCII format)

⇨ Formats for viewing and printing:
- PDF (Portable Document Format)
- MIME (Multipurpose Internet Mail Exchange) for e-mail, it supports various character encodings.

⇨ Compressed formats:
- ARJ, ZIP (Winzip, Gzip)

As we saw there are different file formats.

How to represent a document?

The machine should understand the text. We need to recognize words and analyze them to understand the meaning in the group of words.

- Doc1: Text mining is to identify useful information.
- Doc2: Useful information is mined from text.
- Doc3: Apple is delicious.

	text	information	identify	mining	mined	is	useful	to	from	apple	delicious
Doc1	1	1	1	1	0	1	1	1	0	0	0
Doc2	1	1	0	0	1	1	1	0	1	0	0
Doc3	0	0	0	0	0	1	0	0	0	1	1

Bags of words representation

(originally way)

We need to associate an identifier for each document. The table on the right is an example to understand how the algorithm classify the text.

The words in the columns are the words taken from the document, classified as binary 0,1.

STAGES OF TEXT PROCESSING

⇨ Tokenization

Cut character sequence into word tokens.
- Deal with "John' s " , a state-of-the-art solution

The input is a text, and the output is a set of tokens. Each token is a candidate to be a meaningful term. The second step is to understand wicha are the token that we really need for analysis.

The main problem is when we meet words that are difficult to separate. (Matteo's; lowercase, lower-case, lower case; San Francisco).

We can have similar problems using numbers, dates, telephone number etc.

There may be also language issues; for example in german you can find very long compund nouns, that can be splitted in different words.

The tokenization can depends, tokens can be phrases, words and also symbles. How can we do that? We can use regular rxpressions or statistical methods.

- Input: "***Friends, Romans and Countrymen***"
- Output: Tokens
  - ***Friends***
  - ***Romans***
  - ***Countrymen***
- A token is an instance of a sequence of characters
- Each token is now a candidate to be a meaningful term (index), after further processing
  - Described below
- But what are valid tokens to emit?

- German noun compounds are not segmented
  - ***Lebensversicherungsgesellschaftsangestellter***
  - 'life insurance company employee'
  - German retrieval systems benefit greatly from a **compound splitter** module
    - Can give a 15% performance boost for German
- Break a stream of text into meaningful units
  - Tokens: words, phrases, symbols
    - **Input:** It's not straight-forward to perform so-called "tokenization."
    - **Output(1):** 'It's', 'not', 'straight-forward', 'to', 'perform', 'so-called', ""tokenization.""
    - **Output(2):** 'It', "", 's', 'not', 'straight', '-', 'forward, 'to', 'perform', 'so', '-', 'called', "", 'tokenization', '.', ""
  - Definition depends on language, corpus, or even context

- We may need to "normalize" words :
  - We want to match ***U.S.A.*** and ***USA***
- Result is terms: a term is a (normalized) word type
- Most commonly, an implicit definition of equivalence classes of terms is obtained by, e.g.,
  - deleting periods to form a term
    - ***U.S.A., USA*** ⟨ ***USA***
  - deleting hyphens to form a term
    - ***anti-discriminatory, antidiscriminatory*** ⟨ ***antidiscriminatory***

⇨ Normalization

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

We need to normalize words. It means referring to a unique word. It can be used also for other languages, for example French, were we can find several words with accents.
We can also apply the lower case for all the letters during the normalization process.
Map text and query term to same form
- You want U.S.A. and USA to match.

⇨ Lemmatization
- Reduce inflection/variant forms to base form.

It modifies the words to keep all in the same way, for example 'am, are, is' to 'be'.

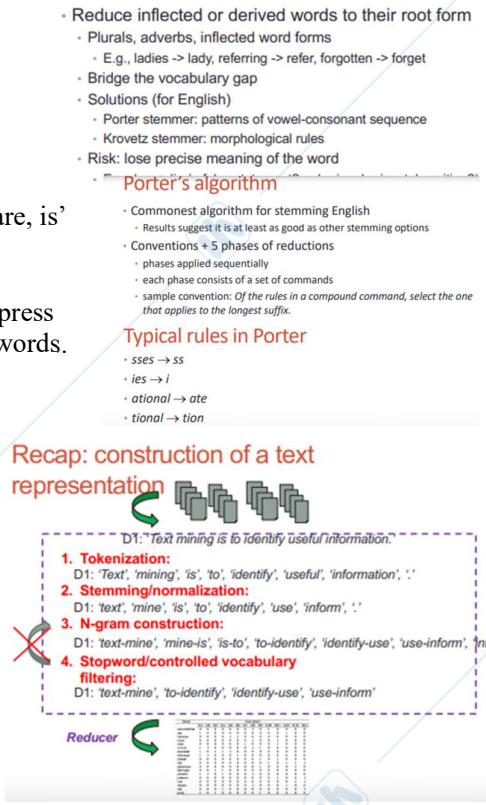
⇨ Stemming

Like lemmatization, it differs because it brings a word that can be express in different ways to the same. The risk is to lose precise meaning of words.
We may wish different forms of a root to match.
- authorize, authorization.

For the stemming process there are algorithms and rules usually applied that we could see on the right.
Does stemming help in retrieval?
Very useful for Spanish, German, Finnish and so on
In English, it's very variable, we can obtain mixed results.

⇨ Stop words removal.

Stop words are words that can be not useful for our model.
We may omit very common words (or not)
- the, a, to, of



POS, NER AND NLP

Now we will see how text representation can be enriched with simple *NLP* techniques.
Analyze the semantic and syntactic structure of the text. We need to find the way to process, applying a formal representation to data. Texts are objects with inherent complex structure. One of the simplest way to represent text is the BoW (Bag of Words) model that, unfortunately, is not good enough for text understanding.
We are going to analyze the semantic and shyntactic structure of a document, and we are going to use this techniques beacose the bow is not abale to capture important information from the text.

⇨ *Natural Language Processing* (NLP) provides models that go deeper to uncover the meaning.
- Part-of-speech tagging
- NER
- Syntactic analysis
- Semantic analysis
- Discourse structure

Phrases:

⇨ Phrases are more informative than single words. ('black sea', 'black' and 'sea') The first phrase make sense, the single words are two words that can stay together but without any sense if separate.

How are phrases recognized?
Three possible approaches:
- Use word n-grams (we have seen this).
- Identify the syntactic role of words within phrases by using a partof-speech (POS) tagger (we will see it "at high level" today).
- Store word positions of indexes in texts and use proximity operators in queries (we will see this when we will introduce search engines).

We will present POS and Named Entity Recognition as sequence labeling problems or tagging problems: given a sequence of words in input the aim is to define a model that produces in output a sequence of labels (tags). Either

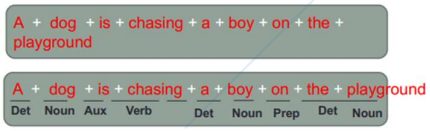
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

this model can be a rule-based model, or it can be a supervised learning problem. An important class of models for supervised learning problems is represented by generative models.

PART OF SPEECH TAGGING (POS TAGGING)

Once the preliminary text processing phases have been undertaken, POS tagging aims at marking up a word in a text (corpus) by a tag corresponding to a particular part of speech (POS tags can be of varying granularity).
Having a text and assigning a *tag* is a classification activity, we must assign one single lable to express the sense of the word. We can do this process very easily but, for a machine that's complicated, in particulare if we have huge amount of text.



N	noun	chair, bandwidth, pacing
V	verb	study, debate, read
ADJ	adjective	purple, tall, ridiculous
ADV	adverb	unfortunately, slowly
P	preposition	of, by, to
PRO	pronoun	I, me, mine
DET	determiner	the, a, that, those
CONG	conjunction	and, or

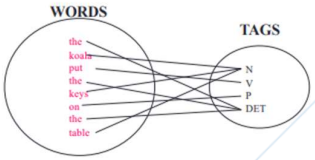
WORD CLASSES

Words that somehow 'behave' alike:

- Appear in similar contexts.
- Perform similar functions in sentences.
- Undergo similar transformations.

⇨ ~ 9 traditional word classes of parts of speech for IndoEuropean languages

- *Noun, verb, pronoun, adjective, preposition, adverb, article, conjunction, interjections*
- Called: parts-of-speech, lexical categories, word classes, morphological classes, lexical tags, POS.



Pos Tagging:

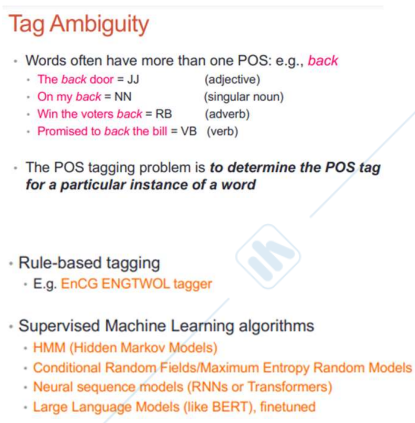
Original text:
Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Brill tagger:
Document/NN will/MD describe/VB marketing/NN strategies/NNS carried/VBD out/IN by/IN U.S./NNP companies/NNS for/IN their/PRP agricultural/JJ chemicals/NNS ,/, report/NN predictions/NNS for/IN market/NN share/NN of/IN such/JJ chemicals/NNS ,/, or/CC report/NN market/NN statistics/NNS for/IN agrochemicals/NNS ,/, pesticide/NN ,/, herbicide/NN ,/, fungicide/NN ,/, insecticide/NN ,/, fertilizer/NN ,/, predicted/VBN sales/NNS ,/, market/NN share/NN ,/, stimulate/VB demand/NN ,/, price/NN cut/NN ,/, volume/NN of/IN sales/NNS ./.

Eric Brill. 1992. A simple rule-based part of speech tagger. In Proceedings of the third conference on Applied natural language processing (ANLC '92). Association for Computational Linguistics, Stroudsburg, PA, USA, 152-155

⇨ To do POS tagging, first *need to choose a set of tags:*
⇨ Could pick very coarse (small) tagsets (we can choose to use a subset of the labels we saw before)
- N, V, Adj, Adv.

⇨ More commonly used: Brown Corpus (general corpus, Francis & Kucera '82), 1 million words, **87 tags** – more informative but more difficult to tag.
⇨ Most commonly used: Penn Treebank – **45 tags**: hand- annotated corpus of Wall Street Journal



We can incur in tag ambiguity, where a word can have different tags depending by the sentence.
Applying the POS tagging we will find a possible POS assignments. So different solutions to our problem. Some solutions should be more probable than the others.
Many words have only one POS tag (is, Mary), other have a *most likely* tag (dog), other ones have a tend to co-occur regularly with other tags.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Some ways to do POS tagging:

Rule based tagging:

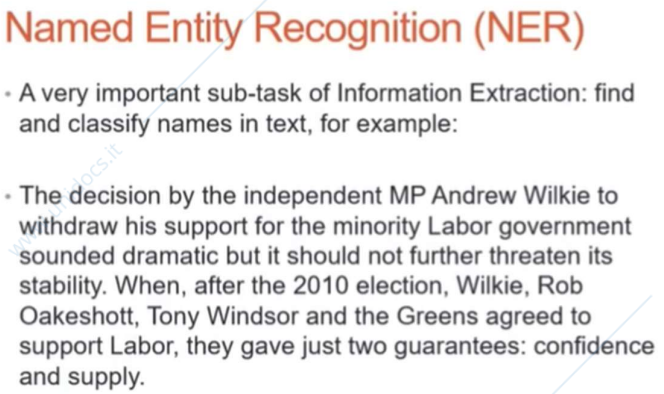
1. Dictionary: each word has a probability to appear as a noun, a verb etc.
2. Assign the probability of possible tags to the words.
3. Write rules to selectively remove tags.
4. Leaving the correct tag for each word.

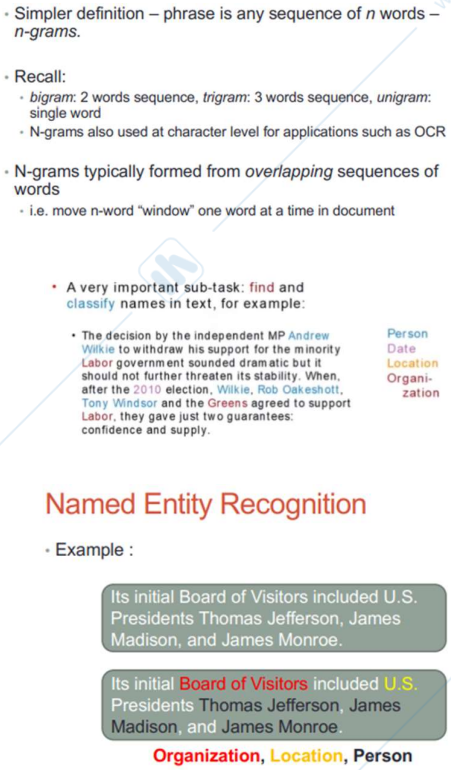
When do we use POS tagging?

Unfortunately, Pos tagging is too slow for large collections.

NAMED ENTITY RECOGNITION (NER)

Given a part of text we should find which are these entities.





Linguistically difficult problems:

- ⇨ NER involves identification of proper names in texts, and classification into a set of predefined categories of interest.
- ⇨ Three universally accepted categories: person, location and organization.
- ⇨ Other common tasks: recognition of date/time expressions, measures (percent, money, weight etc), email addresses etc.
- ⇨ Other domain-specific entities: names of drugs, medical conditions, names of ships, bibliographic references etc.

We have different applications for NER:

- ⇨ Yellow pages with local search capabilities: find for example an address from a text.
- ⇨ Monitoring trends and sentimental textual social media (specific type of persons 'what a student think about university Bicocca')
- ⇨ Interactions between genes and cells in biology and genetics (specific domain useful to understand a specific topic)

NLP

Natural language processing means the "usage" of natural language by computers.

Example of systems that require capabilities of language understanding are, for example: - Search Engines, Machine Translation, Question Answering, Intelligent personal assistant (siri, cortana), Text summarization.

Very important to keep in mind that for the machines is difficult to understand the context that's very important to give the right answer.

NLP is difficult…

Natural language is designed to make human communication efficient.

Therefore,

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

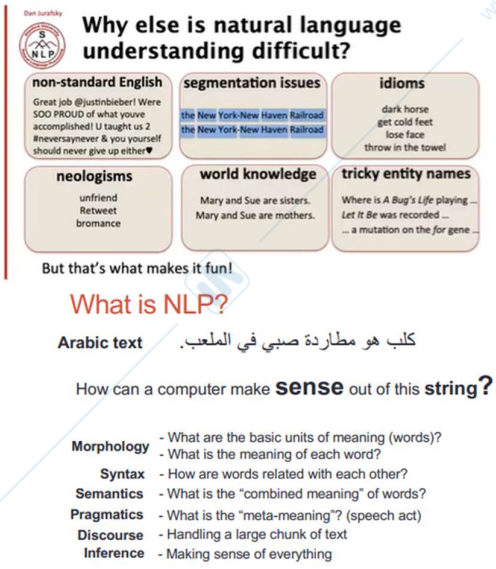
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

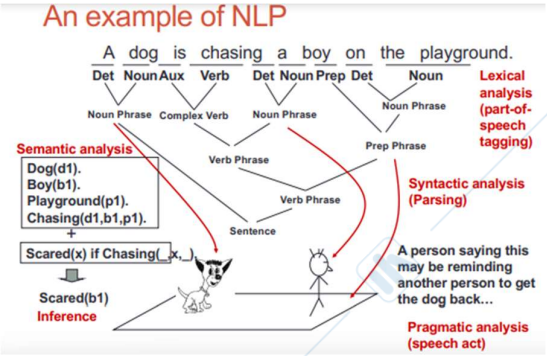
- We omit a lot of "common sense" knowledge, which we assume the hearer/reader possesses.
- We keep a lot of ambiguities, which we assume the hearer/reader knows how to resolve

⇨ This makes EVERY step in NLP hard.
Ambiguity is a "killer"!
Common sense reasoning is pre-required

The ambiguity can be contained in different levels:

- Lexical
  For example, 'peach' has multiple meanings.
- Syntactic
  A man saw a boy with a telescope -> there are two different meanings.
- Semantic
  La minestra brucia (brucia means different things)
- Pragmatic





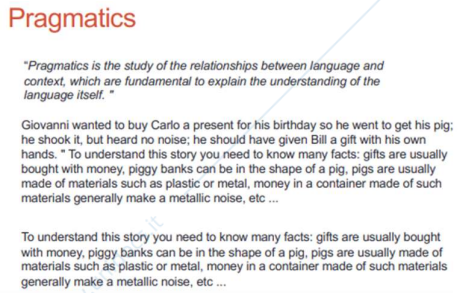
WORD EMBEDDING

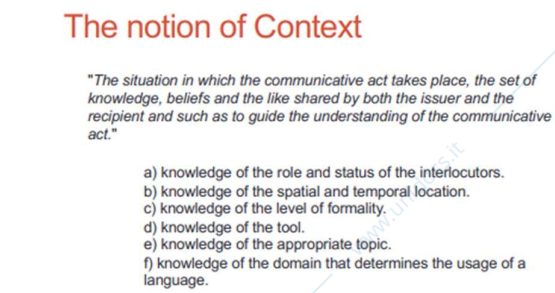
⇨ Vector semantics

The term word embedding indicates a set of techniques in Natural Language Processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers.
It involves a mathematical embedding from a space with many dimension per word to a vector space with a much lower dimension.
The models to generate this mapping include:

- Count-based models (distributed semantic models)
- Predictive models (Neural networks models)

Representing documents as vectors





- **Option 1**: Binary representation.

$|D| = N$

V	d_1	d_2	d_3
bear	1	0	0
cat	0	1	0
frog	0	0	1

$d_1 = [1, 0, 0]$ $d_2 = [0, 1, 0]$ $d_3 = [0, 0, 1]$

- **Option 2**: Raw frequency representation.

	d_1	d_2	d_3
bear	85	0	0
cat	0	10	0
frog	0	0	44

$d_1 = [85, 0, 0]$ $d_2 = [0, 10, 0]$ $d_3 = [0, 0, 44]$

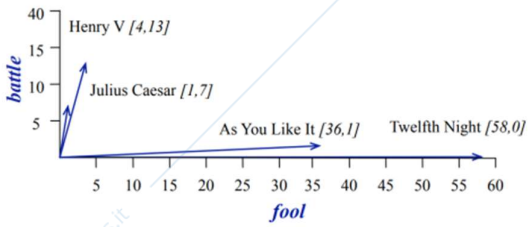
- **Option 3**: Weighted representation.
  - Weighted term frequency (different possibilities)
  - **tf-idf**

	d_1	d_2	d_3
bear	0.48	0	0
cat	0	0.48	0
frog	0	0	0.48

$d_1 = [0.48, 0, 0]$ $d_2 = [0, 0.48, 0]$ $d_3 = [0, 0, 0.48]$

similarity of documents:

Vectors of the two comedies are similar. They are different with respect to the history plays.
The vector representation of documents is at the basis of information retrieval.



www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Words can be represented as vectors too:
in the term-document matrix representation, a possible interpretation could be:

- *battle* is "the kind of word that occurs history plays, in Julius Caesar and Henry V especially".
- *fool* is "the kind of word that occurs in comedies, especially Twelfth Night".

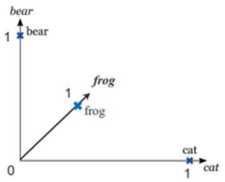
	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Usually, the similarity of words is not computed by using the term-document representation.
Two words are similar if their '*context vectors*' are similar.
The employed matrix representation, in this case, has words on both rows and columns.
Local representation:
Each word is represented by a vector of words. ***Option 1:***

- Each element represents a different word. Also known as '1-hot' or '1-0f-V' or local representation.
  This method tell us very little. We need a separate dimension for every word we want to represent.

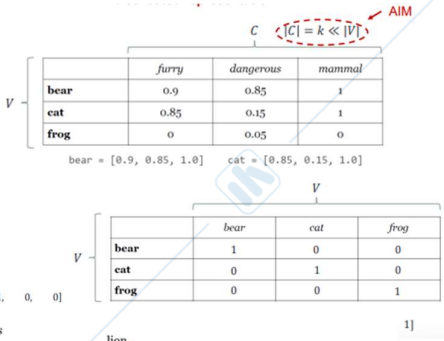
There are few problems:

- Number of dimensions increases linearly as we add words to the vocabulary.
- The matrix is very sparse, mainly made up of zeros.
- There is no shared information between words and no commonalities between similar words.



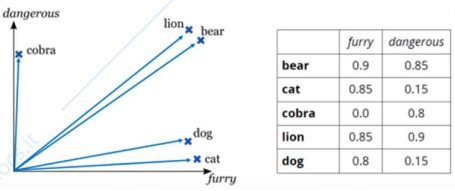
Distributed representation:
Option 2:
To each word of the vocabulary are associated k "context dimensions" that represent "properties" associated with the words of the vocabulary. It's also known as distributed representation.
In this case we can capture the similarity between words, and so we can compute the similarity between them.
This case is in six-dimensional space. (for simplicity in 2 dimesnion)

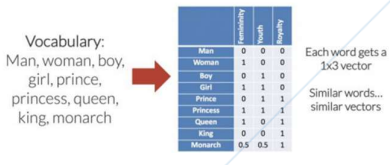
- For simple scenario, we can create k-dimensional mapping for a simple example vocabulary by manually choosing contextual dimensions that make sense.

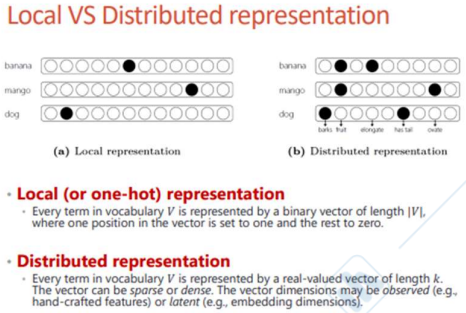


Some advantages:

- Each word is represented with a ***k***-dimensional vector.
  ⇨ Optimal representations are those with $\boldsymbol{k} \ll |\boldsymbol{V}|$.
- Similar words have similar vectors.
  ⇨ There's a smaller distance between vector representation for "girl" and "princess", than from "girl" to "prince".







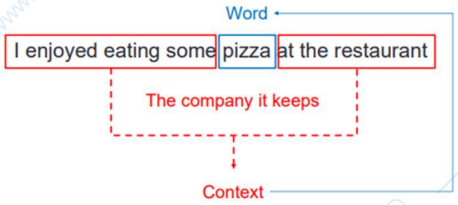
Distributional hypothesis

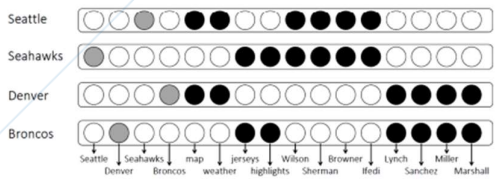
- Central idea: represent each word by some context, e.g., words co-occurring with the considered word.
  ⇨ We can use different granularities of contexts: documents, sentences, phrases, n-grams.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Example: *Neighboring terms*



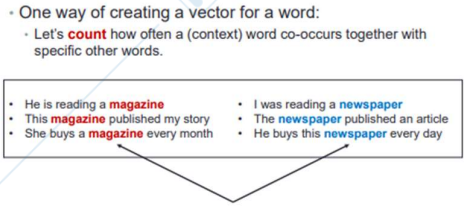


(b) "Neighbouring terms" features

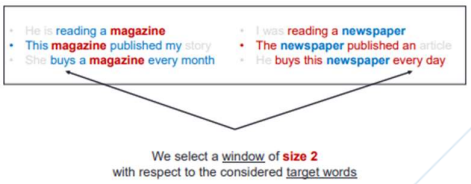
We can see which are the most similar. So representation based on the context.

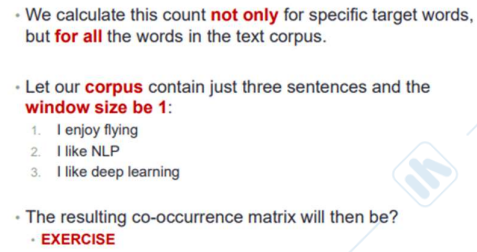
- In this method, given a text corpus, we count the number of times (context) word co-occurs:
  - ⇨ inside a window of a particular size,
  - ⇨ with the word of interest (i.e., target word).
- The resulting matrix is also known as (window-based).
  - ⇨ Word-word co-occurrence Matrix
  - ⇨ Term-context Matrix
  - ⇨ Count Matrix • Each word is represented by a so-called Count vector.

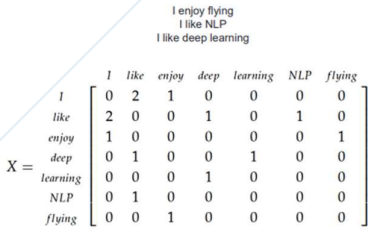
Window-based Co-occurrence Matrix



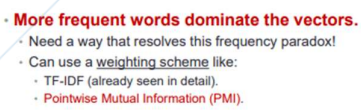
I have to decide which is the size the size of the window before and after the target words. So for example as we can see in the picture on the right. Let's take two words.
We are still not in the word embedding context.
How does it work in general?







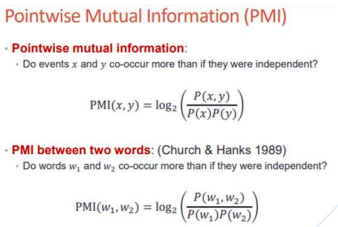
	reading	a	this	published	my	buys	the	an	every	month	day
magazine	1	2	1	1	1	1	0	0	1	1	0
newspaper	1	1	1	1	0	1	1	1	1	0	1



Raw frequency is a bad representation. Frequency is clearly useful; if sugar appears a lot near apricot, that's useful information.
PMI ranges + infinite and – infinite
Negative values are problematic:

- Things are co-occurring less than we expected by chance.
- Unreliable without enormous corpora.

We just replace negative PMI with 0 values. If you want to check a little bit better follow this link:



https://elearning.unimib.it/pluginfile.php/1411688/mod_resource/content/1/MV01.%20TMS%20%20-%20Word%20Embedding%20%28Vector%20semantics%29%20%282022-23%29.pdf and go to page 48. Going on there is also the '**weighting P(PMI)**'

Count-based and predictive models

- A Co-occurrence Matrix in reality is constituted by a very large number of words.
  - ⇨ For each word, TF-IDF or PPMI or other weighted vectors are:
    - long (length $|V| = 20{,}000$ to $50{,}000$)
    - sparse (most elements are equal to zero).
- There are techniques to learn lower-dimensional vectors for words, which are:
  - short (length = 50 to 1,000) (usually around 300)
  - dense (most elements are non-zero)

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

- These dense vectors in a latent space are called embeddings.

We are going to understand how this vector are computing.
We have two main family:

- Count-based models:

⇨ Distributed semantic models

- Predictive models

⇨ Neural networks models

Count-based models

Count-based models

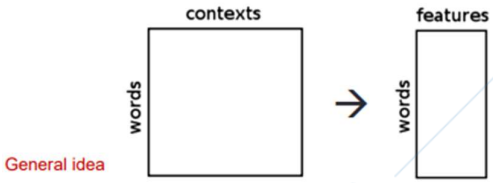
- Compute the statistics of how often each word co-occurs with its neighbor words in a large text corpus;
- Then map these count-statistics down to a small, dense vector for each word.

- Count-based models learn vectors by doing dimensionality reduction on a term-context matrix.
- The term-context matrix contains the information on how frequently each "word" (stored in rows), is seen in some "context" (the columns).

- They factorize this matrix to yield a lower-dimensional matrix of words and features, where each row yields a (dense) vector representation for each word.

Methods:

- Latent Dirichlet allocation (LDA)
- *Singular Value Decomposition (SVD)*
- *GloVe*



The Idea is that we start from the left position and then we would like to move to the right image so keeping just the features, depending on how I have computed the words.

	DOG	CAT	MOUSE
DOG	1		
CAT		1	
MOUSE			1

The Idea is to have the table we saw before so having an example like this:
It represents the context of the text with nearness, in the sense that if the two words appears a lot of time nearby it will be a significative way to classify it. The metrics will be computed form a sort of aggregation associated to a target word.

Predective models

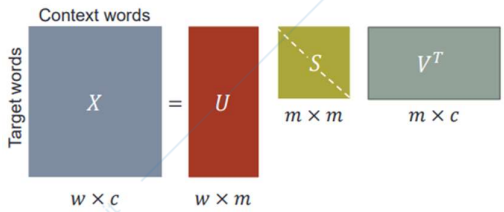
Try directly to predict a word from its neighnors in terms of learned small, dense embedding vectors.

- Word2vec => main algorithm used for this purpose

Singluar value decomposition (SVD)

Any rectangular $w \times c$ matrix X can be expressed as the product of 3 matrices:

- U: a $w \times m$ matrix where the w rows correspond to rows of the original matrix X, but the m columns represents a dimension (feature) in a new latent space.
- S: diagonal $m \times m$ matrix of singular values expressing the importance of each dimension (feature).
- V^t : transposed $m \times c$ matrix where the c columns correspond to the columns of the original matrix X, but the m rows correspond to singular values.



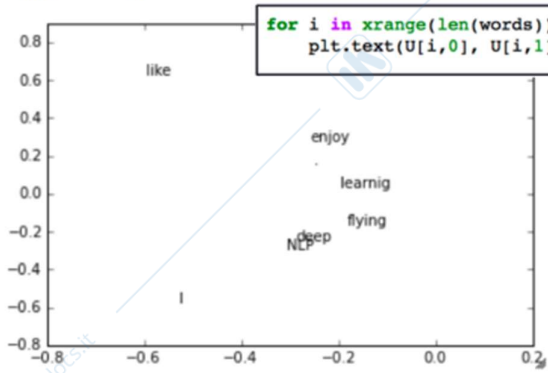
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

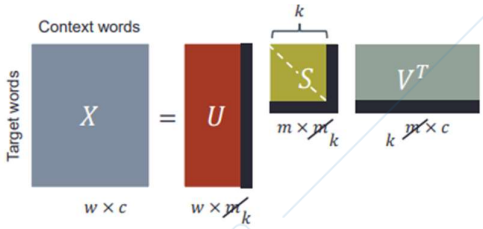
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

If, instead of keeping all m dimensions, we just keep the top-k singular values, we obtain a low-rank approximation of the original matrix X.
We are selecting the k important values in s.
Some example in Python for SVD

- Printing first two columns of U corresponding to the 2 biggest singular values

```
for i in xrange(len(words)):
    plt.text(U[i,0], U[i,1], words[i])
```






```
import numpy as np
la = np.linalg
words = ["I", "like", "enjoy",
         "deep","learnig","NLP","flying","."]
X = np.array([[0,2,1,0,0,0,0,0],
              [2,0,0,1,0,1,0,0],
              [1,0,0,0,0,0,1,0],
              [0,1,0,0,1,0,0,0],
              [0,0,0,1,0,0,0,1],
              [0,1,0,0,0,0,0,1],
              [0,0,1,0,0,0,0,1],
              [0,0,0,0,1,1,1,0]])

U, s, Vh = la.svd(X, full_matrices=False)
```

- The dimensions of the matrix change very often
- The matrix is extremely sparse since most words do not co.occur
- Quadratic cost to perform SVD
- Requires the incorporation of some 'hacks' on X to account the drastic imbalance in word frequency:
  - ⇨ Ignore words such as "the", "he", "she", "has", etc.
  - ⇨ Apply a ramp window - weight the co-occurrence count based on distance between the words in the document.
  - ⇨ Use Pearson correlation and set negative counts to 0 instead of using just raw count of PPMI

Basic notation

- X → the term-context matrix.
- X_{ij} → the frequency of word j occurring in context of word i.
- $X_i = \sum_k X_{ik}$ → the global frequency of any word appearing in the context of word i.
- $P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$ → probability that word j appears in the context of word i → **co-occurrence probability**

GloVe

- Global Vector

The model leverages statistical information by training only on the non-zero elements in a word-word co-occurrence matrix. Rather than:

- on the entire sparse matrix (e.g., SVD)
- on individual context windows in a large corpus (e.g., word2vec).

Global corpus statistics are captured directly by the model.

Example

- Can **certain aspects of meaning** be extracted directly from co-occurrence probabilities?
- Consider two words i and j that exhibit a particular aspect of interest; for concreteness, suppose we are interested in the concept of **thermodynamic phase**, for which we might take $i = ice$ and $j = steam$.
- The **relationship of these words** can be examined by studying the **ratio** of their co-occurrence probabilities with various "probe" words (i.e., context words), k.

Example

- For words k related to $i = ice$ but not $j = steam$, say $k = solid$, the ratio $\frac{P_{ik}}{P_{jk}}$ **should be large**.
- Similarly, for words k related to $j = steam$ but not $i = ice$, say $k = gas$, the ratio $\frac{P_{ik}}{P_{jk}}$ **should be small**.
- For words k like $water$ or $fashion$, that are either related to both $i = ice$ and $j = steam$, or to neither, the ratio $\frac{P_{ik}}{P_{jk}}$ **should be close to "1"**.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

- **Co-occurrence probabilities** for target words *ice* and *steam* with selected context words from a 6 billion token corpus.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

- Only in the ratio capture non-discriminative words like *water* and *fashion*, because:
  - **large values** (much greater than 1) correlate well with properties specific to *ice*.
  - **small values** (much less than 1) correlate well with properties specific of *steam*.

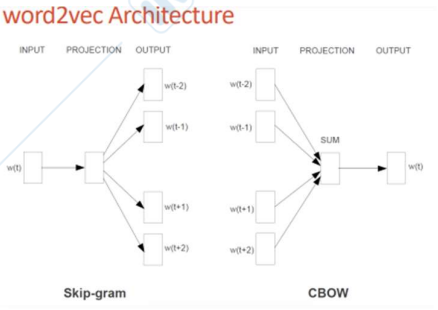
Lot of formulas in the slides. (Word embedding count-based…)

Word2vec

This technique provides a tool to create collections of similar concepts automatically, on raw texts and without advanced language skills on the part of the user.

- Raw texts are used as implicitly supervised training data. No need for hand-labeled supervision.

The largest the training set, the better performance.
The texts should include as many different words as possible.



Sip-gram model

Predict the surrondings rods based on the current words.
Explaination more exhaustive in the slides. (mainly visualization)

Cbow model

Predict the current word based on the surronding words

Statistical Language models

A text can be represented as a language model to represent its topics.

- Words that tend to occur often when discussing a topic will have high probabilities in the corresponding language model.

A text can be represented from a statistical point of view. The idea is that we are going to represent text form a context perspective, so not just with the text itself.
A model specifying probability distribution over word sequences:
It can be seen as a probabilistic mechanism for generating text. Also called as 'generative' model.

- P("*Today is Wednesday*") ≈ 0.001
- P("*Today Wednesday is*") ≈ 0.0000000000001
- P("*The eigenvalue is positive*") ≈ 0.00001

Why should we use this text reperesnetation?

- Machine translation:
  P(high winds tonite) > P(large winds tonite)
- Spell correction:
- Speech recognition:

We apply it also in Text categorization.

- Given that we observe 'baseball' three times and 'game' once in a new article, how likely is it about 'sports' v.s. 'politics'

Also applied in Information retrieval

- Given that a document is centered on the topic of sport, how likely would a query 'generate' by this document?

- **Goal**: compute the probability of a sentence or sequence of words:
  $P(W) = P(w_1,w_2,w_3,w_4,w_5...w_n)$
- **Related task**: probability of an upcoming word:
  $P(w_5|w_1,w_2,w_3,w_4)$
- So, a model that computes either of these:
  $P(W)$ or $P(w_n|w_1,w_2...w_{n-1})$ is called a **language model**.

We always use language model, for example when google suggests us the next word when we are asking something. (Query compilation)

Notation

- To represent the probability of a particular random variable X_i taking on the value "the", or $P(X_i = \text{"the"})$, we will use the simplified notation $P(the)$.
- a sequence of n words is denoted either as $w_1 \ldots w_n$ or as w_1^n
- the joint probability of each word in a sequence having a particular value: $P(X = w_1, Y = w_2, Z = w_3,\ldots,W = w_n)$ is denoted as $P(w_1,w_2,\ldots,w_n)$.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

How to compute $P(w_n|w_1,w_2\ldots w_{n-1})$?

- Let us start by computing $P(w_n|w_1,w_2\ldots w_{n-1})$, the probability of a word w_n given a sequence of words.
- For example: *P(the|its water is so transparent that)*
- Relative frequency counts: given a ***very large corpus***, count the number of times we see *its water is so transparent that*, and count the number of times it is followed by *the*.

$$P(the|its\ water\ is\ so\ transparent\ that) = \frac{C(its\ water\ is\ so\ transparent\ that\ the)}{C(its\ water\ is\ so\ transparent\ that)}$$

Too many possible sentences!

How to compute P(W) ?

- Similarly, if we aim to know the probability P(W) of a sentence W (i.e., the joint probability of an entire sequence of words like *its water is so transparent)*, we could do it by asking "out of all possible sequences of five words, how many of them are *its water is so transparent*?"
- To do so, we would have to get the count of *its water is so transparent* and divide by the sum of the counts of all possible five word sequences.
- *That seems rather a lot to estimate*!!!

p(B|A) = P(A,B)/P(A) Rewriting: **P(A,B) = P(A)P(B|A)**

- More variables:

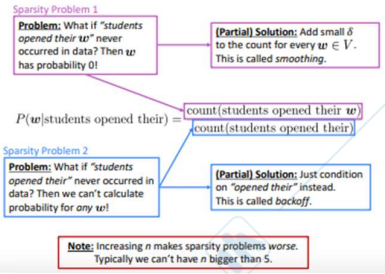
  $P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$
- The Chain Rule in General

  $P(x_1,x_2,x_3,\ldots,x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1,\ldots,x_{n-1})$

N-grams Language Models

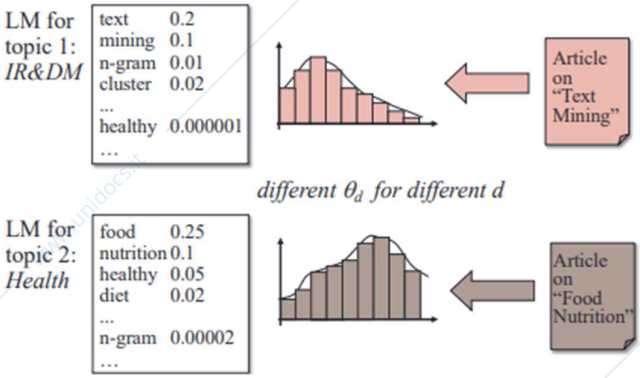
- *Unigram language model*
  - probability distribution over the words in a language
  - generation of text consists of pulling words out of a "bucket" according to the probability distribution and replacing them
  - *PROBABILITIES OF WORDS IN A SEQUENCE DO NOT DEPEND ON PREVIOUS WORDS*
- N-gram language model
  - some applications use bigram and trigram language models where probabilities depend on previous words
  - BIGRAM LM: the probability of a word in a sequence depend on the word that preceeds it
  - TRIGRAM LM: the probability of a word in a sequence depend on the two worda that preceed it

Sparsity Problems with n-gram Language Models



Chain rule
Conditional probabilities
In the slides we can find all the documentation about this process

Text representation with unigram LM



So, in the Unigram language model, doesn't care about the context. So use the words only according to the probability of words but not by the probability of the previous one.

Recap: Language Models

A language model is **well-formed** over alphabet Σ if $\sum_{s\in\Sigma^*} P(s) = 1$.

Generic Language Model

"Today is Tuesday"	0.01
"The Eigenvalue is positive"	0.001
"Today Wednesday is"	0.00001
...	

Unigram Language Model

"Today"	0.1
"is"	0.3
"Tuesday"	0.2
"Wednesday"	0.2
...	

Bigram Language Model

"Today"	0.1
"is" | "Today"	0.4
"Tuesday" | "is"	0.8
...	

How to handle sequences?

- Chain Rule (requires long chains of cond. prob.): $P(t_1t_2t_3t_4) = P(t_1)P(t_2|t_1)P(t_3|t_1t_2)P(t_4|t_1t_2t_3)$
- Bigram LM (pairwise cond. prob.): $P_{bi}(t_1t_2t_3t_4) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)$
- Unigram LM (no cond. prob.): $P_{uni}(t_1t_2t_3t_4) = P(t_1)P(t_2)P(t_3)P(t_4)$

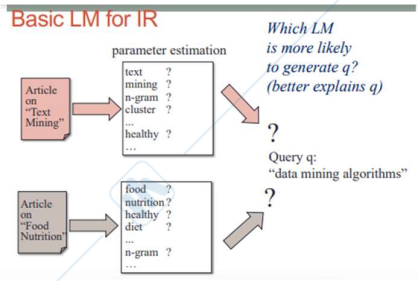
The N-gram models depends on the N number of previous or next words in the context.
In general the N-grams is an insufficient model of language. Because language has long-distance dependencies: 'The computer which I had just put into the machine room on the fifth floor crashed'.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

LMs for retrieval
3 opportunities:

- probability of generating the query text from a document language model.
- probability of generating the document text from a query language model.
- comparing the language models representing the query and document topics.



Estimating biagram probabilities: slides (TXT Mining and Search Language Models, more or less page 20-25)

Smoothing
I may want to determine non zero probability. So the probability of a term in a text.
Document texts are a sample from the language model.

- missing words should not have zero probability of occurring.

Smoothing is a technique for estimating probabilities for missing (or unseen) words.

- lower (or discount) the probability estimates for words that are seen in the document text
- assign that "left-over" probability to the estimates for the words that are not seen in the text

The probability is normalized by dividing by the total number of the occurrences, so if words are missing the score will be 0.
The problem is that I have to assign a probability for each text, but if some terms are not appearing of course it will be 0, so we will add a small probability to get a result. Otherwise all the probability would be 0.

Neural Language Models
To overcome some limitations of Statistical LM, neural LM have been defined.

- Fixed window neural LM
- RNN (recurrent NN) LM
- BERT (Bidirectional Encoder Representations from Transformers)
- BERT's variants

Evaluation
Prediction is not always the main goal of a project
What are we going to evaluate? (Computer science in general)

- system (search engine, or similar)

- Does our language model "prefer" good sentences to bad ones?
  - Assign higher probability to "real" or "frequently observed" sentences than those sentences that "rarely observed" or "ungrammatical" ?

There is a difference between effectiveness and efficiency:

- *effectiveness: behavior of the system, IR, I'm satisfied if I can find the resources I'm trying to find (example from a search engine). So is the level of satisfaction from the initial idea of the project.*
- *efficiency: is usually represented by time and space; Is it working with the right amount of data, for example*

We evaluate both of them, so effectiveness and efficiency. To measure the effectiveness we need metrics (*recall* and *precision for example*).
The language model is going to give high probability to the right words.

1. We train parameters of our model on a training set.
   - So we will need the right source. If I want to train a model about medicine, and I train in a Chemical dataset, it will not be accurate.
2. We test the model on data we have never seen.
   - an evaluation metric tells us how well our model does on the test set

Best evaluation
To compare two language models A and B, I'll observe the results i will obtain. I will check:

- spelling collector, speech recognizer, MT system

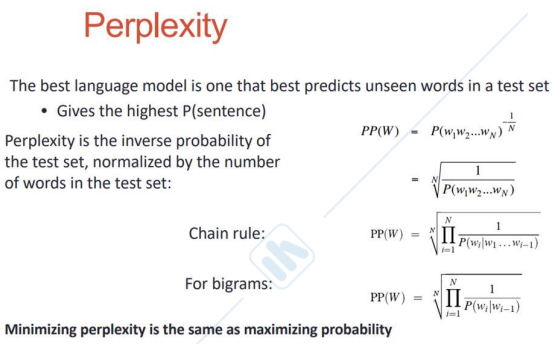
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Then run the task, and get an accuracy for A and B:
- How many misspelled words corrected properly
- how many words translate correctly
- COMPARE ACCURACY FOR A AND B

The *Extrinsic evaluation* works very well, but is Time consuming, can take days or weeks to complete.
Ao sometime we use **intrinsic** evaluation: **perplexity**
- bad approximation

So generally speaking the test data looks just like the training data.
So generally it is useful for pilot experiments.
So *Lower perplexity <-> better model*

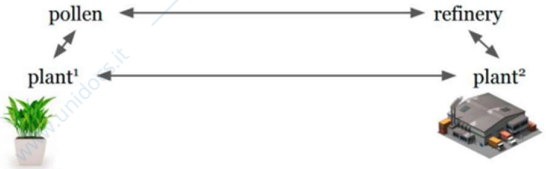


Language models for Retrieval
Search engine is a system. We formulate our information needs through keywords. A part of the search engine is structured for understanding what we are asking.
Text is complex and needs to be *represented*. And there exists also a comparison query, so to be aligned to check the text in the same ways. I cannot compare two different documents if one is represented from a mathematical point of view and the other one is represented with a statistical point of view.
Query-Likelihood Model:
The search engine gives a ranking for the similarity, so we will see first the most similar results and then the next ones.

tf.idf weight -> explanation on the slide and in the book

Contextualized Word Embeddings
The contextualized word embeddings is useful to understand the text, because, for example, we can have different meanings of a word in different context.
Words are semantically dynamic;
they change role depending on the context in wich they are used.
- One word can have more than one meaning, for example 'mouse'.

Word representation cannot capture polysemy



Which is the interpretation of the word 'Words':
- just unique words
- all the words in different sentences

I like the cat - the cat likes me

When we use word embeddings we are looking for word **types**, abstract and unique object.
What makes a word token different from a word type?
- we have the context of the word
- the context may inform the embeddings

So, word embeddings should:
Unify superficially different words
- bunny and rabbit are similar

Capture information about how words can be used
- go and went are similar, but slightly different from each other

Separate accidentally similar looking words
- Words are polysemous

=> The bank was robbed again
=> We walked along the river bank
- Sense embeddings

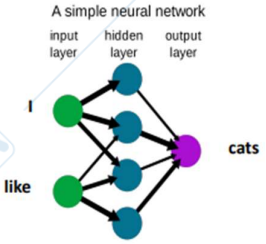
Type embeddings vs token embeddings
- Type embeddings can be thought of as a lookup table.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Map words to vectors independent of any context
A big matrix

- **Token embeddings** should be functions:
  Construct embeddings for a word on the fly
  There is no fixed 'bank' embedding, the usage decides what the word vector is.

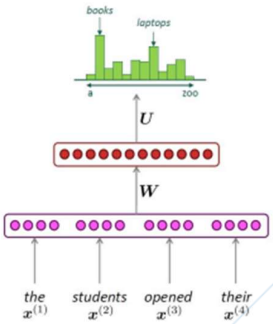


In the slide we can find the explanation about the sparsity problem and other statistical calculation, I'm not going to write all these stuff here 🙂 (slide pag 40 more or less)

How can we solve the problem about the Token embeddings?

- if we want to predict the next words we can use a neural network, these neural network models produce context-specific word representations at each position

It works with an input layer, an hidden layer and an output one.
The usage decides what the word vector is.



How does the model works?
Improvements over n-grams LM:

- don't need to store all observed n-gram;
- no sparsity problem

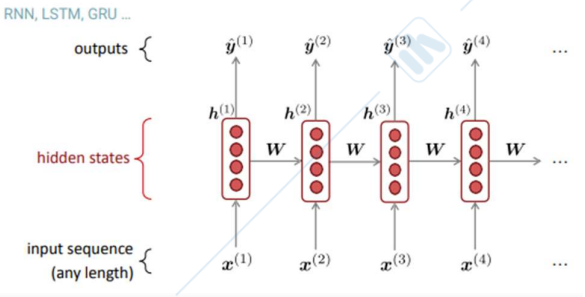
Remaining problems:

- Fixed window is too small;
- window can never be large enough.

So *we need a neural architecture that's able to process any length input*.

Recurrent Neural Network (RNN)
Different structure that the one we saw before.
basically the input can be much longer, so the length is not fixed.
We have some disadvantages:

- recurrent computation is slow;
- In practice, it is difficult to access information from many steps back.



There are several Github repo about the RNN.
We can train this model for several case. For example we can fit with recipe and get new recipe back.

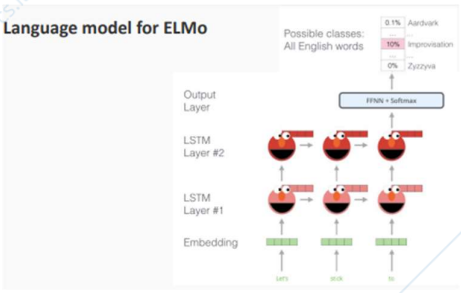
Contextual embeddings
given a sentence, we wnat word embeddings taht depends on the sentence itself.
two models:

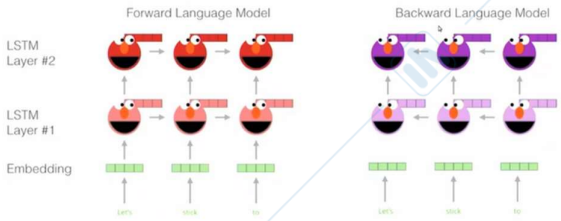
- ELMo
  Embeddings from Language Models:
  Use an RNN architecture (LSTM)
  Use the internal states of a language model as the word embeddings
  Main carhacteristich:
  - Contextual
  - deep
  - character-based

  Embed word types into a vector
  - can use pre-trained embeddings (GloVe)

  Deep bidirectional LSTM language model over the embeddings
  - two layers of BiLSTMs, but could be more

  loss = language model loss
  - cross-entropy over probability of seeing the word in a context



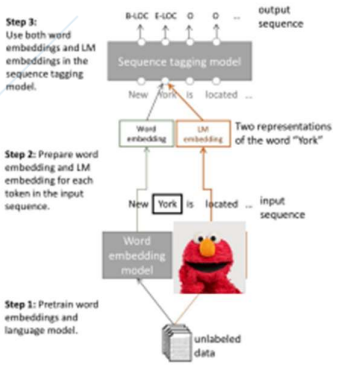


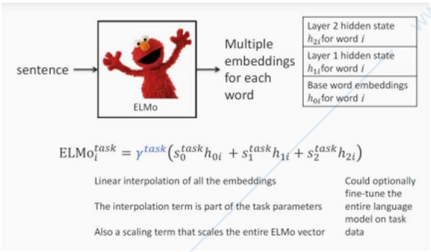
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

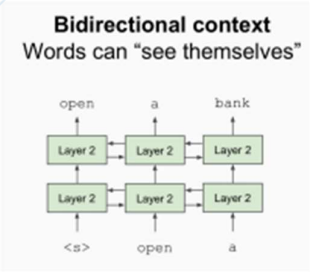
The limitation for this algorithm is that it is unidirectional, so we are wasting context. it reads one word a time from left to right.
So maybe the solution could also be to also read from right to left.

How does the ELMo works?





- BERT
  Bidirectional Encoder Representations from Transformers (provided by Google)
  The problem is that language models only use left context or right context, but language understanding is bidirectional.
  - Directionality is needed to generate a well-formed probability distribution.

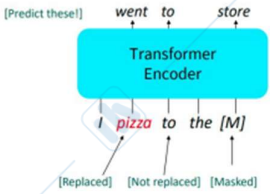


to learn the relationship between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence.

Devlin et al., 2018 proposed the "Masked LM" objective and **released the weights of a pretrained Transformer**, a model they labeled BERT.

Some more details about Masked LM for BERT:

- Predict a random 15% of (sub)word tokens.
  - Replace input word with [MASK] 80% of the time
  - Replace input word with a random token 10% of the time
  - Leave input word unchanged 10% of the time (but still predict it!)



Some other slides skipped, about models, with some links for datasets and so on.

TEXT CLASSIFICATION

- *What is classification?*

is a task with the aim to assign to some object a class. In our case it is a text document, so we will try to classify based on the text.
The class must not be with certainty. In text classification, data items are textual:

- emails, new articles, tweets, sentences, questions, ecc.

We have different types of classification:

- *binary classification*
  assigning an email to spam or not spam (two categories)
- *single-label multi class classification*
  assigning new articles to one of HomeNews, international, entertainment ecc.
- *multi-label multi class classification*
  assigning computer science articles to classes in the ACM classification system. (may belong to more than one class)
- *ordinal classification*
  assigning product reviews to one of: excellent, good, ao and so, poor, disastrous.

We can have two different classification methods, *hard classification* or *soft classification*. The first one happens when we have for example 50% and 50% probability for a classification (spam not spam), the soft classification is when we have a probabilistic method. so for example classifying by topic (98% topic about house, 2% topic about something different)
Usually the HC is used for fully autonomous classifiers, while SC is used for interactive classifiers.

DIMENSION OF TEXT CLASSIFICATION

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Text classification may be performed according to several dimensions:
- by topic: most frequent case, its applications are ubiquitous;
- by sentiment: useful in market research, online reputation management, customer relationship management, social science, political science.
- by language; useful, in query processing within query engines;
- by type: automotiveNews or automotiveBlogs, is useful in website classification and others;
- by author
- by native language
- by gender: useful in forensics and cybersecurity.

Application of text classification
- knowledge organization

- Long tradition in both science and the humanities.
  - The goal was organizing knowledge, i.e., **conferring structure** to an otherwise unstructured body of knowledge.
- The rationale is that using a structured body of knowledge is easier / more effective than if this knowledge is unstructured.
- Automated classification tries to automate the tedious task of assigning data items based on their content, a task otherwise performed by human annotators (or "assessors", or "coders").

- **Scores of applications** (examples):
  - Classifying news articles for selective dissemination
  - Classifying scientific papers into specialized taxonomies
  - Classifying patents
  - Classifying "classified" ads
  - Classifying answers to open-ended questions
  - Classifying topic-related tweets by sentiment
  - ...
- **Retrieval** (as in search engines) could also be viewed as (binary + soft) classification into `Relevant` vs. `NonRelevant`.

Filtering
Performing a binary classification, so filter by relevance in what I would like to know.
It belongs to the hard classification.
Some example could be:
- spam filtering
- detecting unsuitanble content (racism, violent content)

Empowering other IR tasks
functional to improving the effectiveness of other tasks in IR or NLP
Many of these tasks involve classifying very small texts

- Some examples:
  - Classifying queries by intent within search engines
  - Classifying questions by type in QA systems
  - Classifying named entities
  - Word sense disambiguation in NLP systems

SUPERVISED LEARNING AND CLASSIFICATION
- A generic learning algorithm is used to train a classifier from a set of manually classified examples.
- The classifier learns, from these training examples, the characteristics a new text should have on order to be assigned to class c.

TEXT REPRESENTATION
- documents are usually converted in vector in a common vector space.
- the dimensions of vectors space are called features, and the number k of features used is called the dimensionality of the vector space.

In order to generate a vector-based representation for a set of documents D, the following steps need to be taken:
- feature extraction
  has the goal of identifying the most discriminative features, so that the others may be discarded.
- feature selection or feature synthesis
  matrix decomposition techniques (PCS, SVD, LSA) can be used to synthesize new features that replace the features discussed above.
  These methods are based on the principles of semantics, which states that the semantics of a word is the words it co-occurs with in corpora of language use.
- feature weighting

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Feature weighting means attributing a value w_{ki} to feature t_k in the vector $\vec{x}_i$ that represents document d_i: this value may be:

- Binary (representing presence/absence of t_k in d_i).
- Numeric (representing the importance of t_k for d_i). It can be obtained via feature weighting functions in the following two classes:
  - Unsupervised: e.g, $tf - idf$.
  - Supervised: e.g., $tf * MI$, $tf * \chi^2$.

Unigrams - Bi-grams - n-grams

SUPERVISED LEARNING
for binary classification any supervised learning algorithm can be used for training a classifier; popular choice include:

- Support vector machine
- boosted decision stumps
- logistic regression
- naive bayesian methods
- lazy learning methods

for non binary classification we have the opportunity to use:

- decision trees
- boosted decision stumps
- logistic regression
- naive bayesian methods
- lazy learning methods

EVALUATING A CLASSIFIER
two important aspects in the evaluation of a classifier are *efficiency* and *effectiveness*

TEXT CLUSTERING
Clustering is the process of grouping a set of objects (documents) into classes of similar objects.

- documents within a cluster should be similar.
- documents from different clusters should be dissimilar.

Is the commonest form of unsupervised learning.

Classification	Clustering
Supervised learning	Unsupervised learning
Classes are human-defined and part of the input to the learning algorithm	Clusters are inferred from the data without human input
Output = membership in class only	Output = membership in class + distance from centroid ("degree of cluster membership")

The cluster hypothesis
documents in the same cluster behave similarly with respect to relevance to information need.
If I focus on clustering and a document tends to be clustered with a group of documents also that one should belong to the same/similar topic or something else similar.

- Issues for clustering

representation for clustering:
document representation: vector? normalization?
need a notion of similarity/distance

How many clusters?
fixed a priori?
completely data-driven?
avoid trivial clusters - too large or small.

Notion of similarity/distance

- ideal: semantic similarity.
- practical: term-statistical similarity
  docs as vectors
  for many algorithms, easier to think in terms of a distance (rather than similarity) between docs.
  we will mostly speak of Euclidean distance (real implementation use cosine similarity)

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Algorithms:

- Flat algorithms
  creates flat clusters without any explicit structure that would relate clusters to each other.
  - k-means clustering
  - model-based clustering
- hierarchical algorithms

We can also have two different kind of clustering: hard and soft

- hard: each document belongs to exactly one cluster (common and easier to do)
- soft: the document can belong to more than one cluster (fractional membership in several clusters)

Flat clustering Problem statement 1 and 2:

We can define the goal in hard flat clustering as follows:

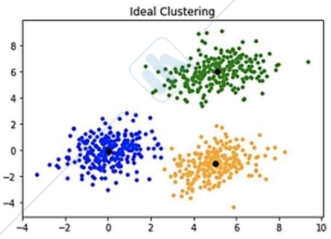
- Given:
  - (i) a set of documents $D = \{d_1, d_2, \ldots, d_N\}$,
  - (ii) a desired number of clusters k,
  - (iii) an *objective function* that evaluates the quality of a clustering.
- We want to compute an assignement

$$\gamma: D \rightarrow \{\omega_1, \omega_2, \ldots, \omega_k\}$$

that minimizes (or, in other cases, maximizes) the objective function.

- In most cases, we also demant that γ is surjective, i.e., that none of the k clusters is empty.
- The objective function is often defined in terms of similarity or distance between documents.
- For (textual) documents, the type of similarity we want is usually topic similarity or high values on the same dimensions in the Vector Space Model.
  - E.g., documents about China have high values on dimensions like Chinese, Beijing, and Mao, whereas documents about the UK tend to have high values for London, Britain, and Queen.

K-means



is the most important flat clustering algorithm.

- documents are represented as length-normalized vectors in a real-valued space in the familiar way
- its objective is to minimize the average squared Euclidean distance of documents from their cluster centers.

the first step is to select as initial cluster centers, k randomly selected documents.
the algorithm then moves the cluster centers around in space in order to minimize distance

Termination conditions

several possibilities:

- a fixed number of iterations -> insufficient number of iteration
- doc partition unchanged -> good clustering, runtime could be too long.
- centroid positions do not change.
- the distance between documents centroids falls below a certain threshold.

How many clusters?

- finding the 'right' number of clusters is part of the problem.

Hierarchical Clustering

Hierarchical outputs a hierarchy, a structure that is more informative than the unstructured set of clusters returned by flat clustering.
Hierarchical clustering does not require to prespecify the number of clusters.
Advantages of hierarchical clustering come at the cost of lower efficiency.

Bottom-up and top-down HC

Hierarchical clustering algorithms are either bottom-up or top-down.
Bottom-up clustering treats each document as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all clusters have been merged into a single cluster that contains all documents.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Top-down clustering requires a method for splitting a cluster. It proceeds by splitting clusters recursively until individual documents are reached.

Hierarchical agglomerative clustering

- starts with each doc in a separate cluster
  - then repeatedly joins the closest pair of clusters, until there is only one cluster.
  - Hierarchical clustering employs a measure of distance/similarity to create new clusters.
- the history of merging forms a binary tree or hierarchy

Evaluation

internal criteria:

- The Silhouette analysis measures how well an observation is clustered and it estimates the average distance between clusters.
- The Silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.
- The Silhouette Coefficient is calculated using the mean intra-cluster distance $a(i)$ and the mean nearest-cluster distance $b(i)$ for each sample i.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Range of Silhouette values

- $S(i)$ will lie between $[-1,1]$.
  - If Silhouette value is close to 1, sample is well-clustered and already assigned to a very appropriate cluster.
  - If Silhouette value is about to 0, sample could be assigned to another cluster closest to it and the sample lies equally far away from both the clusters. That means it indicates overlapping clusters.
  - If Silhouette value is close to –1, sample is misclassified and is merely placed somewhere in between the clusters.

External criteria:

External criteria

- Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data.
- Assesses a clustering with respect to ground truth… requires labeled data.
- Assume documents with C gold standard classes, while our clustering algorithms produce k clusters, $\omega_1, \omega_2, \ldots, \omega_k$ with n_i members.

TOPIC MODELING

Topic modeling is an unsupervised machine learning technique aimed at:

- scanning a set of documents, detecting word and phrase patterns within them;
- automatically clustering word groups and similar expressions that best characterize a set of documents.

Topic modeling provides topics, which are interpreted as collections of words that make sense together.

So, what is the difference between text clustering and topic modeling?

Example: Five topics from a twenty-five topic model fit on Enron e-mails. Example topics concern financial transactions, natural gas, the California utilities, federal regulation, and planning meetings. We provide the five most probable words from each topic (each topic is a distribution over all words).

Topic	Terms
3	trading financial trade product price
6	gas capacity deal pipeline contract
9	state california davis power utilities
14	ferc issue order party case
22	group meeting team process plan

- in text clustering, the basic idea is to group documents into different clusters based on a suitable similarity measure (or distance)
- In Topic Modeling, the basic idea is to group words into different clusters, where:
  - Each word in the cluster is likely to occur "more" (have a probability of occurrence) for the given topic;
  - Different topics have their respective clusters of words along with corresponding probabilities;
  - Different topics may share some words and a document can have more than one topic associated with it.

In a text is also possible to have multiple topic, but we will be able to find the most dominant topic in the multitude of them.

Main techniques for topic modeling

- Latent Semantic Analysis (LSA)

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

The core idea is to take the Document-Term matrix and decompose it into a separate Document-Topic matrix and a Topic-Term matrix

- Latent Dirichlet Allocation (LDA)
  • Each document is considered a mixture of topics and each word in a document is considered randomly drawn from document's topics;
  • The topics are considered hidden (latent) and must be uncovered via analyzing joint distribution to compute the conditional distribution of hidden variables (topics) given the observed variables, words in documents.

The output of a topic modeling algorithm is a list of topic with associated clusters of words.

Latent semantic analysis

Is one of the simplest topic modeling methods. It is based on the distributional hypothesis:

- the semantics of words can be grasped by looking at the contexts the words appear in;
- Under this hypothesis, the semantics of two words will be similar if they tend to occur in similar contexts.

It computes how frequently words occur in the documents – and the whole corpus – and assumes that similar documents will contain approximately the same distribution of word frequencies for certain words.
In this case, syntactic information (e.g., word order) and semantic information (e.g., the multiplicity of meanings of a given word) are ignored and each document is treated as a bag of words.

The standard method for computing word frequencies is TF-IDF
Once TF-IDF frequencies have been computed, we can create a Document-Term matrix that contains the TF-IDF values for each term in a given document.

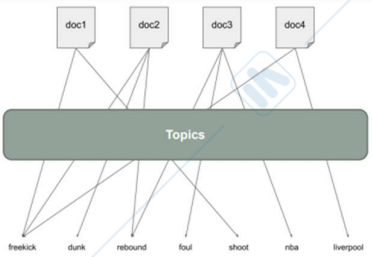
The Document-Term matrix can be decomposed into the product of 3 matrices ($USV\ T$) by using Singular Value Decomposition (SVD).
The U matrix is known as the Document-Topic matrix and the V^T matrix is known as the Topic-Term matrix.
Linear algebra guarantees that the S matrix will be diagonal, and LSA will consider each singular value, i.e., each of the numbers in the main diagonal of matrix S, as a potential topic found in the documents.
Probabilistic LSA
The main goal of probabilistic LSA (pLSA) is identifying and distinguishing between different contexts of word usage without recourse to a dictionary or thesaurus.
• It allows to disambiguate polysemy, i.e., words with multiple meanings;
• It discloses topical similarities by grouping together words that share a common context
The pLSA uses a probabilistic method instead of SVD to tackle the problem.
The core idea is to find a probabilistic model with latent topics that can generate the data we observe in our Document-Term matrix.

- We want a model $P(D, W)$ such that for any document $d \in D$ and word $w \in W$, $P(d, w)$ corresponds to that entry in the Document-Term matrix.
- pLSA considers that our data can be expressed in terms of 3 sets of variables:
  • Documents: $d \in D = \{d1, \dots, dN\}$ observed variables. Let N be their number, defined by the size of our given corpus.
  • Words: $w \in W = \{w1, \dots, wM\}$ observed variables. Let M be the number of distinct words from the corpus.
  • Topics: $z \in Z = \{z1, \dots, zK\}$ latent (or hidden) variables. Their number, K, has to be specified a priori.

LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation (LDA) is a Bayesian version of pLSA.

- LDA treats documents as bags of words;
- LDA assumes documents are produced from a mixture of topics;
- LDA categorizes documents by topic via a generative probabilistic model;
- Distribution of topics in a document and the distribution of words in topics are Dirichlet distributions.

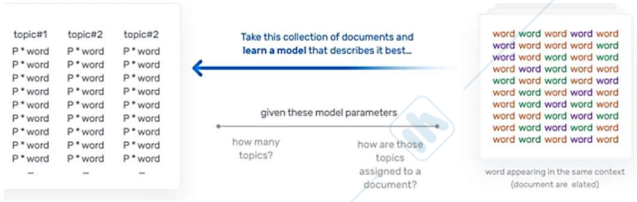


The Idea is that: - words are generated from topics; each document has a particular probability of using particular topics to generate a given word; - we seek to find which topics given documents are likely to use to generate words.
So, some topics are generated starting from documents and others are generated starting from words.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

How does it works?
INPUT: We start with a corpus of M documents and choose how many k topics we want to discover out of this corpus.
OUTPUT: the topic model, and the M documents expressed as a combination of the k topics.
OPERATION: the algorithm finds the weight of connections between documents and topics and between topics and words.



The algorithm created an intermediate layer with topics and figured out the weights between documents and topics and between topics and words.
Documents are no longer connected to words but to topics.
In the previous example, each topic was named for clarity, but in real life, we would not know exactly what they represent.

- We would have topics 1, 2, …, up to k, that's all

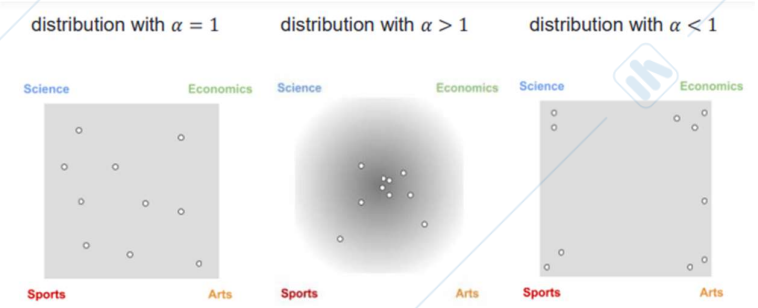
Dirichlet distributions encode the intuition that documents are related to a few topics.
In practical terms, this results in better disambiguation of words and a more precise assignment of documents to topics.

- in a random distribution, documents would be distributed across the four topics. but, in real life, they are distributed not random

So, we are going to use:

- A Dirichlet distribution $Dir(p)$ is a way to model a probability function (PF) which gives probabilities for discrete random variables.

The result are not predictable, we don't know which will be the result, but we have a selection of results (dime example only 6 numbers)
In the example with documents, topics, and words, we have two PFs:
θd: the probability of topic k occurring in document d;
φk: the probability of word w occurring in topic k.
The p parameter in $Dir(p)$ is named concentration parameter, and rules the trend of the distribution to be:



• uniform ($p = 1$)
• concentrated ($p > 1$)
• sparse ($p < 1$)
When we consider topics and words, we denote $p = \beta$.
By using concentration parameters $\alpha, \beta < 1$, these probabilities will be closer to the real world.

Using the same concentration parameter we obtain many different distributions of documents over topics. they get adjusted during the training process to make the model better.

Parameters of LDA
Alpha and Beta Hyperparameters

- Alpha represents document-topic density.
  => Higher the value of alpha, documents are composed of more topics and lower the value of alpha, documents contain fewer topics.
- Beta represents topic-word density.
  => Higher the beta, topics are composed of a large number of words in the corpus, and with the lower value of beta, they are composed of fewer words.

Number of Topics

- Selected randomly.
- By using the Kullback-Leibler (KL) Divergence Score.

Number of Topic Terms

- Generally decided according to the requirement.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

=> If the problem statement talks about extracting themes or concepts, it is recommended to choose a higher number;
=> If problem statement talks about extracting features or terms, a low number is recommended.

Number of Iterations

- Maximum number of iterations allowed to LDA algorithm for convergence.

Evaluating topic models

Eye Balling Models
• Top-n words
• Topics/Documents
Is a sort of human inspection.

Intrinsic Evaluation Metrics
• Capturing model semantics
• Topics interpretability
Two Intrinsic Evaluation metrics that best describe the performance of a topic model:
• Perplexity
measure of uncertainty, meaning lower the perplexity better the model.
It's a statistical measure of how well a probability model predicts a sample.
It aims to capture how "surprised" a model is of new data it has not seen before.
measuring "how probable" some new unseen data is given the model that was learned earlier
• Coherence
measure of semantic similarity between top words in our topic. Higher the coherence better the model performance.
score a single topic by measuring the degree of semantic similarity between high scoring words in the topic.
These measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference.
There are different coherence measures, that are specified in the slide (important for the python script)

Human Judgements
• Quantitative methods for evaluating human judgment

Extrinsic Evaluation Metrics/Evaluation at task
• Is the model good at performing predefined tasks, such as classification?

TOPIC CLASSIFICATION

Topic Modeling is an unsupervised machine learning technique (i.e., it does not require training).

- If there is not the possibility to priorly analyze texts (to label it), or if the aim is not looking for a fine-grained analysis, topic modeling algorithms are indicated.

Topic Classification is a supervised machine learning technique, i.e., it needs training before being able to automatically analyze texts.

- If there is a list of predefined topics for a set of texts, and the aim is to gain accurate insights, topic classification is more suitable.

A topic classification machine learning model needs to be fed examples of text and a list of predefined tags, known as training data.
Once the text is transformed into vectors and the training data is tagged with the expected tags, this information is fed to an algorithm to create the classification model.
Different systems:
Naive Bayes
Family of that deliver good results even when dealing with small amounts of data, say between 1,000 and 10,000 texts;
It works by correlating the probability of words appearing in a text with the probability of that text being about a certain topic.

Support Vector Machines (SVM)
Slightly more complex than Naive Bayes;

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

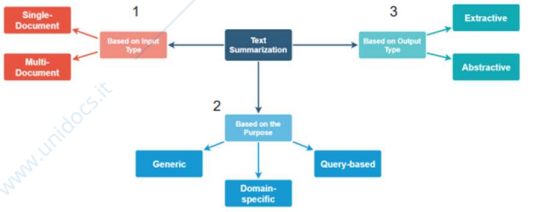
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

They often deliver better results than NB for topic classification;
Downside: they require complex programming and require more computing resources.
It is possible to speed up the training process of an SVM by optimizing the algorithm by feature selection, in addition to running an optimized linear kernel such as scikit-learn's Linear SVC.

Deep Learning
Topic Classification benefit from Deep Learning;
It employs two main deep learning architectures:
• Convolutional Neural Networks (CNN);
• Recurrent Neural Networks (RNN).
• Downside: They require much more training data than traditional machine learning algorithms.
• Instead of, for example, 1,000 training samples, it is necessary to have millions of samples.
There are also hybrid systems, these are simply combinations of machine learning classifiers and rule-based systems, which improve results as you fine-tune rules.

For the evaluation of Machine learning algorithms we will use the classic metrics:
- accuracy
- precision
- recall
- F1 score

TEXT SUMMARIZATION
"Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)"
The goal is to produce an abridged version of a text that contains information that is important or relevant to a user.
There are severa summarization applications:
- headlines
- summaries
- minutes
- previews
- reviews
- digests
- biography



The advantages are:
reduction of reading time when searching for documents , summaries make the selection process easier
automatic summarization improves the effectiveness of indexing
We can produce also the personalized summaries, that are useful in question-answering system as thy provide personalized information.
Using automatic or semi-automatic summarization system enables commercial abstract to increase the number of text documents they can process.

1) Based on input type
   Single-document summarization.
   Given a single document, produce:
   - Abstract
   - Outline
   - Headline

   The input length is short. Many of the early summarization systems dealt with single document summarization.
   Multiple-document summarization
   Given a group of documents, produce a gist of the content, e.g.:
- A series of news stories on the same event.
- A set of Web pages about some topic or question.

The input can be arbitrarily long.

2) Based on the purpose

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Generic
- The model makes no assumptions about the domain or content of the text to be summarized and treats all inputs as homogeneous.
- The majority of the work that has been done revolves around generic summarization.

Domain-specific
The model uses domain-specific knowledge to form a more accurate summary.
- For example, summarizing research papers of a specific domain, biomedical documents, etc

Query-based
- The summary only contains information which answers natural language questions about the input text.
  Summurize a document with respect to an information need expressed in a user query.
  A kind of complex-answering

3) Based on output type
Extractive summarization
- Important phrases or sentences are selected from the input text.
- The summary is created from these phrases or sentences in the source document(s).
  Several summarization approaches today are extractive in nature.
  Abstractive summarization
- Expresses the ideas in the source document(s) using (at least in part) different words.
- The model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate.

This approach is definitely more appealing, but much more difficult than extractive summarization.

Creating an intermediate representation of the input which captures only the key aspect of the text.
Scoring sentences based on the representation
Selecting a summary consisting of several (scored) sentences.

Topic representation approaches:
- First derive an intermediate representation of the text that captures the topics discussed in the input.
- Based on these representations of topics, sentences in the input document are scored for importance.

Indicator representation approaches:
- The text is represented by a diverse set of possible indicators of importance which do not aim at discovering topicality.
- These indicators are combined, very often using machine learning techniques, to score the importance of each sentence.

A summary is produced choosing the sentences that will go in the summary one by one, or globally optimizing the selection, choosing the best set of sentences to form a summary.

Intermediate representation
Topic representation approaches:
- Topic words approaches in which the topic representation consists of a simple table of words and their corresponding weights, with more highly weighted words being more indicative of the topic.
  • Lexical chain approaches in which a thesaurus such as WordNet is used to find topics or concepts of semantically related words and then give weight to the concepts.
- Latent semantic analysis (LSA) in which patterns of word co-occurrence are identified and roughly construed as topics, as well as weights for each pattern.
- Bayesian topic models (LDA) in which the input is represented as a mixture of topics and each topic is given as a table of word probabilities (weights) for that topic.

Indicator representation approaches:
- Represent each sentence in the input as a list of indicators of importance such as:
  • Sentence length;
  • Location in the document;
  • Presence of certain words;
  • Etc.
- In graph models, such as LexRank and TextRank, the entire document is represented as a network of inter-related sentences.

Scoring sentences

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Once an intermediate representation has been derived, each sentence is assigned a score which indicates its importance.
For topic representation approaches, the score is commonly related to:

- how well a sentence expresses some of the most important topics in the document.
- to what extent it combines information about different topics.

For the majority of indicator representation methods, the weight of each sentence is determined by combining the evidence from the different indicators, most commonly by using machine learning techniques to discover indicator weight

Selecting the summary

The summarizer must select the best combination of important sentences to form a paragraph length summary.
Best n approaches

- The top n most important sentences which combined have the desired summary length are selected to form the summary

Maximal marginal relevance approaches

- Sentences are selected in an iterative greedy procedure:
  • At each step of the procedure the sentence importance score is recomputed as a linear combination between the original importance weight of the sentence and its similarity with already chosen sentences.
  • Sentences that are similar to already chosen sentences are dispreferred.

Global selection approaches

- The optimal collection of sentences is selected subject to constraints that try to maximize overall importance, minimize redundancy, and, for some approaches, maximize coherence.

TOPIC REPRESENTATION APPROACHES

Intuition dating back to Luhn (1958):*

• Choose sentences that have salient or descriptive words (topic words/signatures).
• Frequent content words would be indicative of the topic of the article (stopwords are not considered).
• Frequency thresholds to identify descriptive words in a document to be summarized.

The importance of a sentence is computed as:

1. The number of topic signatures it contains.
2. The proportion of topic signatures in the sentence.

Both sentence scoring functions are based on the same topic representation; despite this, the scores they assign to sentences may be rather different.
• The first approach is likely to score longer sentences higher, simply because they contain more words.
• The second approach favors density of topic words.

Weighting words

When assigning weights of words in topic representations, we can think of binary (0 or 1) or real-value (continuous) weights and decide which words are more correlated to the topic.
The two most common techniques in this category are:
• Word probability
• TF-IDF

- This weighting technique assesses the importance of words and identifies very common words (that should be omitted from consideration) in the document(s) by giving low weights to words appearing in most documents.
- First step:
  • TF-IDF vector representations of the documents are created.
  • A clustering algorithm is run over the TF-IDF vectors, adding documents to clusters and recomputing centroids.
  => Centroids can be considered as pseudo-documents that consist of those words whose TF-IDF scores are higher than a certain threshold and form the cluster.
  - Second step:
    • Using centroids to identify sentences in each cluster that are central to the topic of the entire cluster.
    => The sentences which have more words from centroid of cluster are considered as central sentences.

There are several slides about probability and statistics 🙂

The sumBasic system

The algorithm then selects the best scoring sentence that contains the highest probability word.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

This step ensures that the highest probability word (which should represent the topic of the document) is included in the summary.

- After the best sentence is selected, the probability of each word that appears in the chosen sentence is adjusted (set to a smaller value, e.g., $Pnew\ wi = Pold\ wi * Pold\ wi$)

LSA

- Latent semantic analysis (LSA) is an unsupervised technique for deriving an implicit representation of text semantics based on observed co-occurrence of words.
- LSA has been initially proposed for single and multi-document generic summarization of news, as a way of identifying important topics in documents without the use of lexical resources such as WordNet.
- Building the topic representation starts by filling in a n by m matrix A: each row corresponds to a word from the input (n words) and each column corresponds to a sentence in the input (m sentences).
- Entry aij of the matrix corresponds to the weight of word i in sentence j
  • If the sentence does not contain the word, the weight is zero, otherwise the weight is equal to the TF-IDF weight of the word.
- The matrix A can be represented as the product of three matrices: $A = U\Sigma V\ T$
  • Matrix U is an n (words) by m (topics) matrix of real numbers. Each column can be interpreted as a topic, i.e., a specific combination of words from the input with the weight of each word in the topic given by the real number.
  • Matrix Σ is diagonal m (topics) by m (topics) matrix. The single entry in row i of the matrix corresponds to the weight of the "topic", which is the ith column of U.
  • Matrix $V\ T$ is an m (sentences) by m (topics) matrix, a new representation of the sentences, one sentence per row, each of which is expressed not in terms of words that occur in the sentence but rather in terms of the topics given in U.
  • The matrix $D = \Sigma V\ T$ combines the topic weights and the sentence representation to indicate to what extent the sentence conveys the topic, with dij indicating the weight for topic i in sentence j.

Bayesian topic models

- Bayesian topic models are probabilistic models that uncover and represent the topics of documents.
- Latent Dirichlet Allocation (LDA) model is the state-ofthe-art unsupervised technique for extracting thematic information (topics) of a collection of documents.
  • Documents are represented as a random mixture of latent topics, where each topic is a probability distribution over words.

Indicator Representation Approaches

Indicator representation approaches aim to model the representation of the text based on a set of features and use them to directly rank the sentences rather than representing the topics of the input text.
Graph-based methods and machine learning techniques for summarization are often employed to determine the important sentences to be included in the summary.

- Represented as node in a graph, and the link represent the weight of similarity.

They represent the documents as a connected graph.
• Influenced by the PageRank algorithm.
Sentences form the vertices of the graph and edges between the sentences indicate how similar the two sentences are.
A common technique employed to connect two vertices is to measure the similarity of two sentences and if it is greater then a threshold, they are connected.
The most often used method for similarity measure is cosine similarity with TF-IDF weights for words.
The graph representation results in two outcomes.
• First, the partitions (sub-graphs) included in the graph, create discrete topics covered in the documents.
• The second outcome is the identification of the important sentences in the document.

- Sentences that are connected to many other sentences in the partition are possibly the center of the graph and more likely to be included in the summary

Graph-based methods can be used for single as well as multi-document summarization.
Since they do not need language-specific linguistic processing other than sentence and word boundary detection, they can also be applied to various languages.
Nonetheless, using TF-IDF weighting scheme for similarity measure has limitations, because it only preserves frequency of words and does not take the syntactic and semantic information into account.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Machine learning approaches
(Supervised) machine learning approaches model the summarization as a (binary) classification problem.
Sentences are classified as summary sentences and non summary sentences based on their features.
• Given a training set of documents and their extractive summaries.
The likelihood of a sentence to belong to the summary class, or the confidence of the classifier that the sentence should be in the summary, is the score of the sentence.
The chosen classifier plays the role of a sentence scoring function.
• Input <- intermediate representation;
• Output → score of the sentence.

Training classifiers
A problem inherent in the supervised learning paradigm is the necessity of labeled data on which classifiers can be trained.
• Asking annotators to select summary-worthy sentences.
  - Time consuming.
  - Annotator agreement is low.
• Abstracts written by people (often professional writers).
  - Used also in abstractive methods.
  - One could compute similarity between sentences in human abstracts and those in the input in order to find very similar sentences, not necessarily doing full alignment.

Selecting summary sentences
One of the early summarization approaches for both generic and query focused summarization that has been widely adopted is Maximal Marginal Relevance (MMR).
At each selection step, the greedy algorithm is constrained to select the sentence that is:
• maximally relevant to the user query (or has highest importance score when a query is not available).
• minimally redundant with sentences already included in the summary.

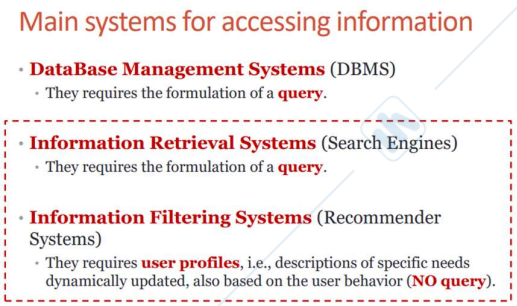
Evaluation
Evaluation of a summary is a difficult task because there is no ideal summary for a document, or a collection of documents and the definition of a good summary is an open question to large extent.
It has been found that human summarizers have low agreement for evaluating and producing summaries.
Additionally, prevalent use of various metrics and the lack of a standard evaluation metric has also caused summary evaluation to be difficult and challenging.

ROUGE is the most widely used metric for automatic evaluation.
Lin* introduced a set of metrics called Recall-Oriented Understudy for Gisting Evaluation (ROUGE) to automatically determine the quality of a summary by comparing it to human (reference) summaries.

There are several variations of ROUGE. The most broadly used are: • ROUGE-*n* • ROUGE-*L* • ROUGE-S • Variants of the above-mentioned measures
There is the explanation of these methods on the slides

INFORMATION RETRIEVAL

Development of systems that help the user to identify information relevant to her/his needs (to inform, i.e., reduce ignorance).
The definition of such systems is based on the solution of a decision problem: how to identify and define the importance of information that satisfies the user's preferences? It is necessary to:
- interpret the content of texts, images, video, audio.
- interpret the user's needs.
- Central role of the notion of relevance: relevance is a subjective property: difficult to define and measure!



www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Help users in performing the decision process, I don't know anything about something, and through the IR I can understand. We have IR system based also in users profile, so if we will search for something it will depend based on our profile.

Pull technology

- The user explicitly requests information in an interactive mode – **Information Retrieval** (IR)
- 3 modes:
  - Browsing (hypertext).
  - Retrieval (IR systems).
  - Browsing and retrieval (digital libraries and Web searches).

Push technology

- The user is automatically updated with information of possible interest – **Information Filtering** (IF)
- Systems for the recommendation of information (goods/ services) provide the user with relevant information for deferred use.

IR

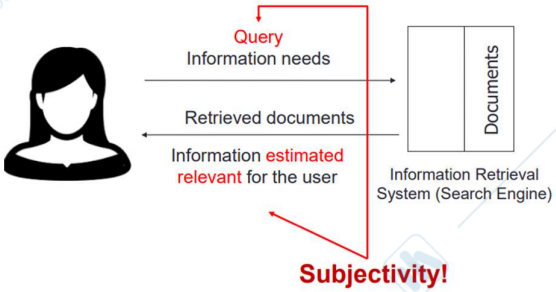
• IR is the computer science discipline that deals with the storage and retrieval of documents.
Its goal is the creation of software systems that allow the storage of large quantities of documents in an archive, to allow an efficient and effective retrieval of documents relevant to users' information needs.

"IR is the name for the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him (…). IR embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, techniques, and machines that are employed to carry out the operation." [Mooers, 1951]

The primary objective of IR Systems is to find the correct and relevant document to the user by neglecting non-relevant documents.
We have subjectivity in different points:

- writing the query
- when the search engine is going to select the documents, we have an estimation of relevance.

Retrieval status value -> is the level of relevance of the document we are receiving

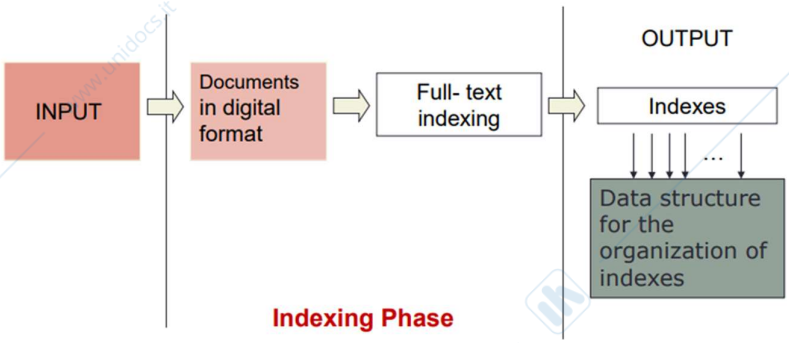


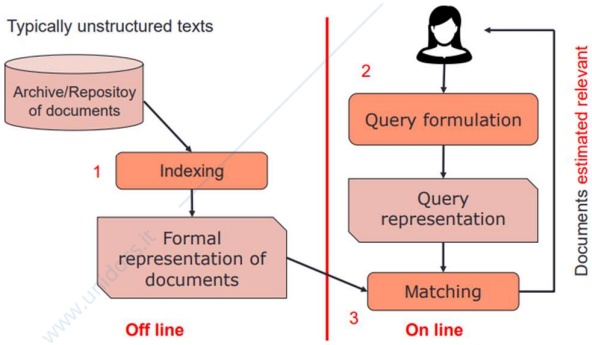
Indexing

We are going to extract the words that are more suitable to represent our document.

- useful tu describe the semantic of a document. The document are represented as weighted sets of words.

We can do automatic indexing and we can create a dictionary.





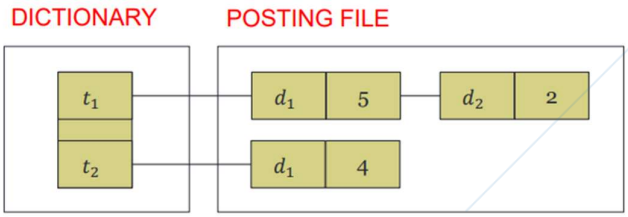
An IRS is based on a mathematical model

We will use the inverted structure, because we want to find the document that contain a certain word. so we will have a table with documents as coulnn and words as rows and the count of each of the words.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

The index terms are organized in a dictionary.
Each term points to a list containing the references to documents of which the term is an index.
Use of two files: dictionary and posting file.



Information retrieval models
An IR system is based on a mathematical model that provides a formal description:

- of the document,
- of the query,
- of how to compare the query and the document representations to estimate the relevance of documents to the query.

It should be noted that the use of the same formal framework to represent both documents and queries guarantees a correct matching.

IMPORTANT: An IR system simplifies the complexity of the retrieval activity → the results produced are not «perfect» (estimate of relevance)

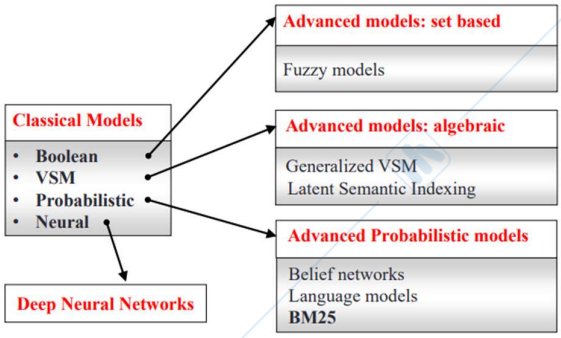
We can have:

- Exact comparison: “binary” notion of relevance
  Relevant / Not relevant.
- Partial comparison: “gradual” notion of relevance:
  Idea: comparison between the document and the query that tolerates mismatches (e.g., similarity of the query to the document)

The boolean model is based on the Set Tehory

- a document is represented as a set of index terms, binary weights associated with index terms.
- a query is formally represented as a boolean expression on terms

Different examples in the slides
The limitations are that it is a binary decision criterion.



VSM Vector Space Model
It is based on linear algebra:
Both documents and queries are represented in an n-dimensional vector space, where n is the number of index terms in the considered collection.

- Inverted file (with simple posting lists).

Dictionary	Posting file
term	*docID*
abacus	3 19 22
actor	2 19 29
aspen	5
atoll	11 34

Posting list of abacus

Words are index terms selected in the indexing phase

Relevance is modeled as a gradual notion and it is proportional to the proximity of the vector that identifies a document to the vector that identifies the query: • proximity = similarity of vectors;
proximity ≈ inverse of distance.

Term independence: terms that appear simultaneously in a document are assumed not correlated.

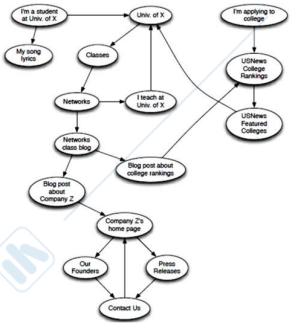
The retrieval status value also needs to be implemented for example with relevance, so if the document is recent it will be probably more important.
For example, also for the restaurant, if I'm searching for a chinese one, I would like to find the nearest ones.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

ACCESSING INFORMATION

Web search



It could be seen as a huge distributed collection of web pages that may contain different types of data. The most popular web search engines are Google and Bing.

We use the Web Graph to represent the web. The nodes are the URLs and there is an edge between node x and node y when the URL x contain a link to the URL y. It is a dynamic graph, since it can change several time.

There are different search methods:

- Direct search given URL.
- Search by link (browsing).
- Use of Web services for search:
  - Search engines that index a portion of Web documents and allow the user to formulate queries and retrieve relevant webpage addresses (Google, Bing, ...).
  - Web portals that in addition to providing a search engine, classify Web documents by directory and provide an interface for browsing the document catalog.
  - Systems supporting electronic commerce (Recommender Systems)

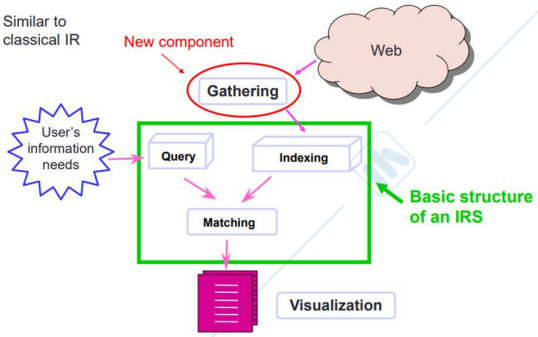
The first search engine appeared in 1994.

In 2000, google was the first search engine to index more than a billion web pages. We are using the second generation:

- link analysis (connectivity)
- click through data (which results are most clicked)
- anchor text (how people get to the page)

Now we are moving also to the third generation:

- semantic analysis
- focus on the needs of the user
- determine of the context
- interaction with the user
- integration of search and text analysis



Web search:

Terminology and definitions:

- A Web page corresponds to a document in traditional IR.
- Web pages differ in size, structure, type of files that are contained (text, graphics, sounds, images, video, format (HTML, GIF, JPEG, ASCII, PDF, etc.).
- Web search considers as a collection of documents the part of the Web that is publicly indexable and excludes pages that require authorizations, dynamic pages, etc.
  - The part that is not publicly indexable is called Hidden Web or Deep Web.

So the web is bigger than the image we have through the web search.

Deep Web (or Hidden Web) indicates the parts of the World Wide Web whose contents are not indexed by standard Web search engines.

- Some examples:
  - dynamic pages, which are dynamically generated by specific applications, then discarded. E.g., flight reservation pages. (for example the recap order in amazon)
  - limited access content pages, where the access is limited in a technical way (e.g., using CAPTCHAs).
  - private Web pages, which require registration and login (passwordprotected resources).
  - unlinked content, i.e., pages which are not linked to by other pages.

Document Gathering

There are two ways:

- web pages are provided directly to the search engine by the owners.
- The search engine is equipped with a software agent (information agent), namely

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

crawler, (spider, worm, robot, Web search agent)
that browses the Web to send new or updated pages to a server that indexes them.

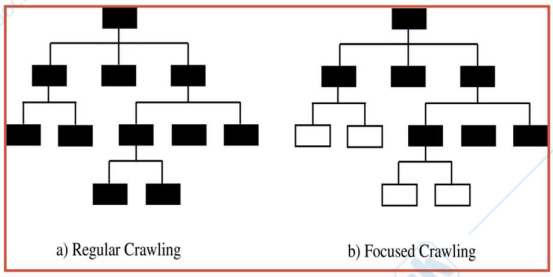
- The crawler surfs the Web using known URLs as starting points (wellknown interesting points of access) and then visits other Web pages through links that go from one page to another.

A crawler must collect pages, essentially by visiting the web graph. They run on local servers and send requests to remote servers. The collection starts from one pages, called seed of collection (seed set).
Crawling process:

1. Initialize a page queue with some known URLs (popular or sent by users)
2. Select a URL address from the queue
3. Select the page
4. Search for other URLs in the Web page
   - For example, publications
5. Discard the URLs that:
   - Can not be analyzed, e.g., .exe
   - Have already been visited
6. Add URLs to the queue
7. If the time has not expired, go back to step 2

```
S = 0              // Already visited pages (store)
F = {seed(…)}      // Border (to be visited)
While (F!=0) {
      choose p in F
      download p from the network
      F = F\{p}
      S = S ∪ {p}
      ∀ link x in p
      if (x ∉ S and x ∉ F)
      F = F ∪ {x}
}
```

Is also possible to have a focused crawling, that only visit pages that are considered relevant to the topic.



Link Analysis
The main difference between Web search and traditional search is the fact that links pointing to a page provide a measure of its popularity. Links between pages indicate the existence of relationships between the pages.
The 2 main criteria for the estimation of relevance are:
• Topicality (thematic relevance): content of the pages.
  • The most common models are the Boolean model and the Vector Space Model.
• Popularity: link analysis.
  • PageRank, HITS, which determine the popularity of Web pages (such as the citations analysis for Impact factor calculation of scientific publications).
Popular pages are considered more reliable than not popular ones.
The popularity of a page grows as the number of its inlinks increases (and may depend from the number of its out-links).
Popular pages are more likely to contain relevant information with respect to non-popular ones.

- Popularity independent from the query - global analysis
  - PageRank (Google): it simulates a random walk through the Web and computes the "degree" of probability of reaching page *p (the page rank of p)*

- Query-dependent popularity - local analysis
  - HITS (Hyperlink Induced Topic Search): it identifies authoritative pages based on those retrieved from a query or those connected to them (to the retrieved pages)

Page Rank:
Basic idea: PageRank simulates a user walk on the Web.
• A page with high PageRank value has:
  • either several in-links,
  • or a few in-links with a high PageRank value.

Retrieval (originally) is a combination of:
• Topicality estimate.
• PageRank value.

Contextual search
Search Engines require users to synthesize/translate their information needs into an expression of a formal query language (generally keyword-based, with Boolean operators)
→ query-centered approach, where the system strongly relies on user queries as explicit signals of the user's interests.

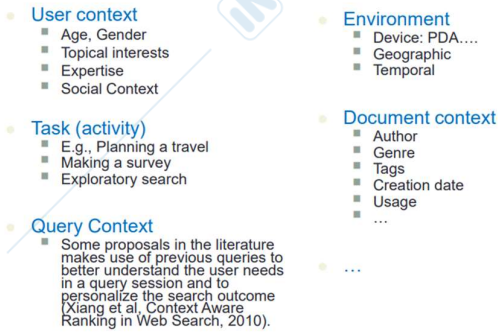
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Poor representation of the information needs as:
• Users may find it difficult to express by a few keywords a complex information need,
• Users are encouraged to enter short queries (both by the search engine interface, and by the fact that long queries do not produce good results),
• Users sometimes have no clear idea of what they are looking for.
tHIS PROCESS BASED ONLY IN A QUERY IS NOT CONSISTENT. For example if we only write Java, we will find results for the programming language but also for the island. Which one is correct?
So We need to think about the context, and so contextual search. There are a lot of contextual aspect:

- location
- type of document
- etc.

So we change from normal search to contextual search. Is easy because the search engine have all the information and can improve the results of the research.



Personalized Search
Produce a search outcome fitting th user context. So we are considering the context of the users.

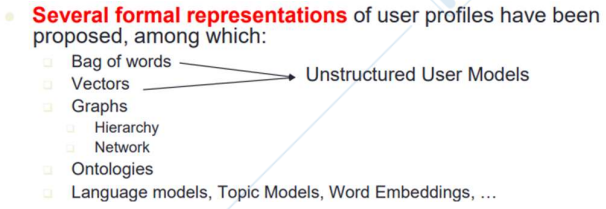
- based on the query and filtered according by the user preference

The user context are:

- demographic data
- content related information

There is also a difference based on long or short term user profiles. Depending on the period of time.

User Profile
Different ways:
Words



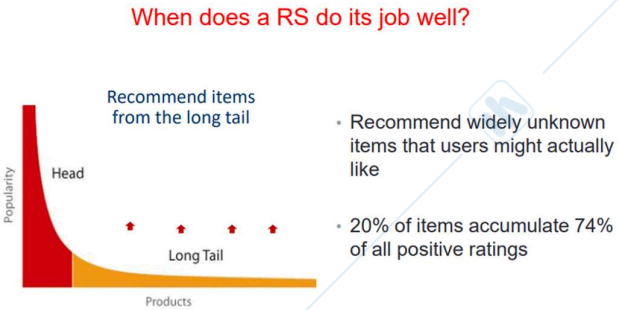
Concepts

- use of external knowledge resources to identify concepts in the user related information (existing taxonomies or ontologies, e.g. OPD –taxonomy-, Wordnet, Yago –Yet another Great Ontology)
- Need a collection of information that represent the user interests
- Need a mapping of the user information to the external resource

Personalization process

- pre-processing approaches: query modification (I search for sport, and find the football ones)
- post processing approaches: result re-ranking
- in-process approaches: personalization integrated in the ranking process.

Recommender system
Without submitting any query we get recommendations based on our usage profile.
it helps to mach users items.



The aim is also to push the products that are in the long tail, otherwise the user will get always the same products.
So, given user model and Items, it find a relevance score that can be used for ranking and finally recommend the items that is assumed to be relevant.

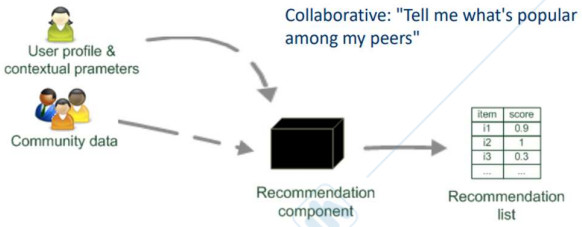
- Recommender system are based on Information Filtering

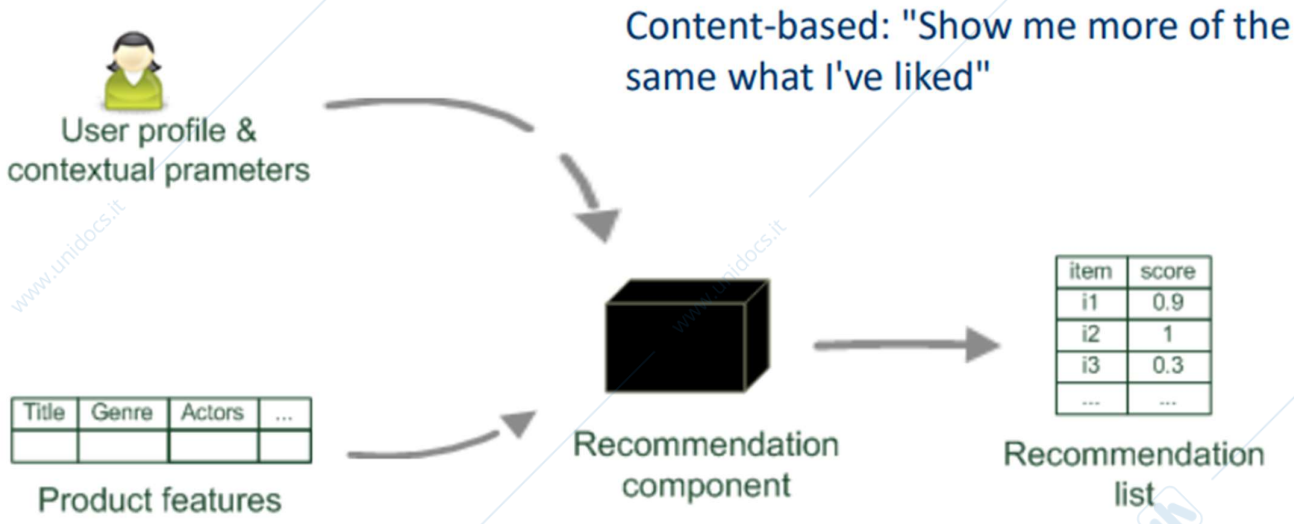
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Information filter: it represents the user's interests and as such has the goal of identifying only those pieces of information that a user would find interesting
how to acquire and possibly dynamically adapt such an information filter to the user's needs?

- collaborative
  the community appreciate some items in the past, I will need to find the users that are similar to me such a way I can recommend that Items to the users.
- content based filtering





User based collaborative filtering:

- input: a matrix of given user-item ratings
- Oustput: numerical prediction.

The basi assumption is the nearest-neighbor, so, if a user had similar tastes in the past, they would have similar tastes in the future.
Given an "active user" (Alice) and an item i not yet seen by Alice
• find a set of users (peers/nearest neighbors) who liked the same items as Alice in the past and who have rated item i
• use the average of their ratings (for example) to predict if Alice will like item i
• do this for all items Alice has not seen and recommend the best rated

- Example
  - A database of ratings of the current user, Alice, and some other users is given:

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

  - Determine whether Alice will like or dislike *Item5*, which Alice has not yet rated or seen

- Some first questions
  - How do we measure similarity?
  - How many neighbors should we consider?
  - How do we generate a prediction from the neighbors' ratings?

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Popular similarity measure is the Pearson correlation coefficient.

In user-based CF

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

• the rating matrix is directly used to find neighbors/make predictions
• does not scale for most real-world scenarios
• large e-commerce sites have tens of millions of customers and millions of items

- Example:
  - Look for items that are similar to *Item5*
  - Take Alice's ratings for these items to predict the rating for *Item5*

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Item-based collaborative filtering.
Basic ideas:

- use the similarity between items to make predictions.

Explicit ratings
Pure CF-based systems only rely on the rating matrix
• Probably the most precise ratings. Most commonly used (1 to 5, 1 to 7 Likert response scales)

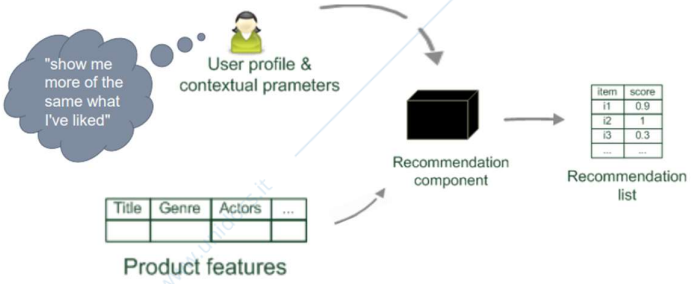
Research topics
• Optimal granularity of scale; indication that 10-point scale is better accepted in movie domain
• Multidimensional ratings (multiple ratings per movie such as ratings for actors and sound)

Main problems
• Users not always willing to rate many items
• number of available ratings could be too small → sparse rating matrices → poor recommendation quality
  • How to stimulate users to rate more items?

So we usually use implicit ratings.
The user checked some items, maybe commented on some of them and so on. The problem is that if I start looking for a present, for example, I will modify my user interest.

Content based filtering
It is based by the user, so it doesn't care about the items

The The content of items can be represented as a textual document structured or unstructured.



www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari