Text Mining &   
TEXT PROCESSING

Notes compiled by –

Sandeep Kumar Kushwaha

University of Bologna, Italy

Year – 2022.

[Gmail, new, logo Icon in Google new logos](mailto:sandeepkumar998855@gmail.com)

[A picture containing text, clipart

Description automatically generated](https://www.linkedin.com/in/xandie985/)

[Shape

Description automatically generated with low confidence](https://github.com/xandie985)

[Icon

Description automatically generated](https://wa.me/+393517231221)

[](https://www.kaggle.com/sandeepkumarkushwaha)

# ­1.0 Text Processing

## 1.1 Structured Data

Pros:

* Highly organized – SQL is used to manage the structured data.
* Easily decipherable by machine learning algorithms.
* Easily used by business users – it does not require an in-depth understanding of different types of data and how they function. With a basic understanding of the topic relative to the data, users can easily access and interpret the data.
* Accessible by more tools – there are more tools that are available for using and analysing structured data.

Cons:

* Limited usage: Data with a predefined structure can only be used for its intended purpose, which limits its flexibility and usability.
* Limited storage options: Structured data is generally stored in data storage systems with rigid schemas (e.g., “data warehouses”). Therefore, changes in data requirements requires an update of all structured data, which leads to a massive expenditure of time and resources.

Tools:

* [**OLAP**](https://www.ibm.com/cloud/learn/olap)**:** Performs high-speed, multidimensional data analysis from unified, centralized data stores.
* [**SQLite**](https://sqlite.org/)**:** Implements a self-contained, serverless, zero-configuration, transactional relational database engine.
* [**MySQL**](https://cloud.ibm.com/catalog/content/mysql)**:** Embeds data into mass-deployed software, particularly mission-critical, heavy-load production system.
* [**PostgreSQL**](https://www.ibm.com/cloud/learn/postgresql)**:** Supports SQL and JSON querying as well as high-tier programming languages (C/C+, Java, Python, etc.).

Use cases for structured data:

* Customer relationship management (CRM) – CRM software runs structured data through analytical tools to create datasets that reveal customer behaviour patterns and trends.
* Online booking **–** Hotel and ticket reservation data (e.g., dates, prices, destinations, etc.) fits the “rows and columns” format indicative of the pre-defined data model.
* Accounting **–** Accounting firms or departments use structured data to process and record financial transactions.

## 1.2 Unstructured Data

The unstructured data does not have a predefined data model, so it is best managed in non-relational (NoSQL) databases. Another way to manage unstructured data is to use data lakes to preserve it in raw form. Around 80-85% of the data is in the form of unstructured.

Pros

* Native format: Unstructured data, stored in its native format, remains undefined until needed. Its adaptability increases file formats in the database, which widens the data pool and enables data scientists to prepare and analyze only the data they need.
* Fast accumulation rates: Since there is no need to predefine the data, it can be collected quickly and easily.
* Data lake storage:Allows for massive storage and pay-as-you-use pricing, which cuts costs and eases scalability.

*Cons*

* Requires expertise: Due to its undefined/non-formatted nature, data science expertise is required to prepare and analyze unstructured data. This is beneficial to data analysts but alienates unspecialized business users who may not fully understand specialized data topics or how to utilize their data.
* Specialized tools: Specialized tools are required to manipulate unstructured data, which limits product choices for data managers.

*Unstructured data tools*

* [MongoDB](https://www.ibm.com/cloud/learn/mongodb): Uses flexible documents to process data for cross-platform applications and services.
* [DynamoDB](https://aws.amazon.com/dynamodb/): Delivers single-digit millisecond performance at any scale via built-in security, in-memory caching and backup and restore.
* [Hadoop](https://www.ibm.com/cloud/blog/hadoop-vs-spark): Provides distributed processing of large data sets using simple programming models and no formatting requirements.
* [Azure](https://www.ibm.com/cloud/architecture/architectures/ibm-cloud-private-azure/): Enables agile cloud computing for creating and managing apps through Microsoft’s data centres.

*Use cases for unstructured data*

* [Data mining](https://www.ibm.com/cloud/learn/data-mining): Enables businesses to use unstructured data to identify consumer behavior, product sentiment, and purchasing patterns to better accommodate their customer base.
* [Predictive data analytics](https://www.ibm.com/analytics/predictive-analytics): Alert businesses of important activity ahead of time so they can properly plan and accordingly adjust to significant market shifts.
* [Chatbots](https://www.ibm.com/cloud/learn/chatbots-explained): Perform text analysis to route customer questions to the appropriate answer sources.

## 1.3 Semi Structured data

They are partially described by some model, such as hierarchical or graphs.

They don’t obey the tabular structure of data models associated with relational databases or other forms of data tables, but nonetheless contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data.

Examples: emails, XML, RDF, and other markup languages, binary executables.

*Dealing with SSD:*

There are methods and languages to partially deal witch SSD, ex: XPath, XQuery, etc.

## 1.4 The practical implications of Text Mining

* 1. Stock market prediction with 86% accuracy.
  2. Wavii – app for gathering, classifications and distribution of news – acq. By Google.
  3. Summly – news summarizing app of iOS – acquired by Yahoo.
  4. Crime Prediction Systems – CRUSH :is a large database of illegal activities. The goal is to predict the crimes. With the historic data of committed crimes, events, criminals, and behaviour 31% reduction of general crimes was achieved.

## 5 Text Mining

Process of extracting knowledge from unstructured textual data.

*Tasks:*

* Text classifications, clustering & topic modelling
* Text Extraction & Summarization
* Sentiment & opinion mining
* Question answering
* Descriptive Text mining: explaining phenomena from text corpora
* Information retrieval: methods & algorithms to search relevant docs. wrt user queries in the repositories of UD (Unstructured Data).
  + First step for Information retrieval is defining the data mining goals. Then,
  + data selection, where checks are performed to ensure the improved data quality.
  + IR offers efficient methods for representation and selection of UD which can be useful for TM (Text Mining).
  + TM offers techniques to improve complex IR searches.
    - Searching similar docs, within flat or hierarchical catalogues organized by topics.

## 1.6 Representation of Documents

In the matrix of documents and words – we place 1 if document contains the word else 0.

A picture containing shape

Description automatically generated

1.7 Binary Vectors

Vectors 0/1 for each term and document

To find presence of *Bruto*, *Cesare* and *Not Calpurnia* – we take the compliment of *Calpurnia*.

Bruto: 110100 (as it is)

Cesare: 110111 (as it is)

~Calpurnia: 101111 (compliment)

AND op: 100100 (performing AND operation on the three values above)

1.8 Boolean Queries: Exact match

In Boolean Retrieval model, the query expressive power is based on Boolean propositions.

These queries consist of AND, OR & NOT

* Each doc is considered a set of terms
* Matching is strict – Binary (Yes/No)

Simplest form of IR.

It’s predominant solution even in the modern software.

1.9 Limits of the Vector Representation

Let’s suppose, the number of documents is N = 1 million. Each of the document contains 1000 terms.

* If each term takes 6 bytes, we will have 6 GB of documents.

Let's suppose, the number of distinct terms M = 500k in million of documents.

500k X 1M is matrix with 500 billion 0 and 1 as values. (at most 1 billion of values as 1)

Sparse matrix of 1000 terms \* 1 M documents.

A better way of representation?

1.10 Inverted index

We store the terms and their occurrence in the form of dictionary, where each term t is stored as key, whereas fixed sized list of all the documents containing t is stored as value of the key. Each document is identified by docID.

*Bruto*  🡪 [1,2,4,11,21,33,45]

*Cesare* 🡪 [1,2,3,6,34,132,-]

*Calpurnia* 🡪 [2,32,55,121,-,-,-]

*keys* on RAM 🡪 *values* are stored on disk

1.10.1 Boolean queries with Inverted Index

We perform AND operation on lists included in the query. So, *Bruto* AND *Cesare* AND ~*Calpurnia.*

1.10.2 Query Optimization: Example

We can visit the lists in increasing order of frequency (no. of docs containing the term).

Starting from the shorter list for better efficiency.

So, we start from *Calpurnia 🡪 Cesare 🡪 Bruto*.

1.10.3 Arbitrary Boolean Queries

How to process more complex queries?

Like, (*Bruto* OR *Cesare*) AND NOT (*Antonio* OR *Cleopatra*)

**Naïve method:** Lista(*Bruto*) U Lista(*Cesare*) – Lista(*Antonio*) U Lista(*Cleopatra*)

(*Bruto* AND ~*Antonio* AND ~*Cleopatra*)OR(*Cleopatra* AND ~*Antonio* AND ~*Cleopatra*)

the union produces usually lists that are longer than intersection, therefore it is more efficient usually to process intersection before union.

1.10.4 Building the Inverted Index

1. Collect the documents to be indexed:

\framebox{\weestrut Friends, Romans, countrymen.} \framebox{\weestrut So let it be with Caesar}

1. **Tokenize the text**, turning each document into a list of tokens:

\framebox{\weestrut Friends} \framebox{\weestrut Romans} \framebox{\weestrut countrymen} \framebox{\weestrut So}….

1. **Linguistic pre-processing**, producing a list of normalized tokens,

\framebox{\weestrut friend} \framebox{\weestrut roman} \framebox{\weestrut countryman} \framebox{\weestrut so} ...

1. **Index the documents** so that each term occurs in by creating an inverted index, consisting of a dictionary and postings.

\begin{figure}
% latex2html id marker 1068
\begin{tabular}{p{2.3in}p{2.6in}}
\te...
...n each document) or the position(s) of the term in each
document.}
\end{figure}

X. Inverted Index: RDBMS vs NoSQL DataBase

**Generalized Inverted Index (GIN)**

* contains an index entry for each term, with compressed list of matching locations.
* Best for static data.
* Lookups are faster.

**Generalized Search Tree (GiST)**

* generalization of several traditional indexes, like B-trees, with no limitation in text size, moreover it allows using arbitrary predicates.
* Best for dynamic data.
* Data update under 100k terms is faster.

DBMS systems are equipped with such indexes for increasing the speed of full text searches.

* NoSQL is used for text manipulation, but it is less efficient than RDBMS.

X.Text Tokenization

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual smaller words/ terms, called tokens.

Challenges in tokenization:

* India’s 🡪 India/Indias/India’s?
* U.S. 🡪 U.+ S. breaking up?
* Co-author 🡪 shall we divide the terms?
* Date formats
* Language issues:
  1. Chinese & Japanese don’t use spaces to separate words.
  2. German has long composed nouns without separations.
  3. Arabic & Hebriw are written R to L, but not in case of numbers.

X. Stop Words

Frequent occurring words carrying very little information.

General stop word list, and domain specific stop words are also there.

X. Word Normalization

Normalization is **the process of converting a token into its base form**. In the normalization process, the inflectional form of a word is removed so that the base form can be obtained.

X. Language dependent Normalization and Tokenisation

Both of them depend of the language. But there are cases where multilingual usage is present, how to handle such cases?

Search Expansion: users give additional input on query words or phrases, possibly suggesting additional query terms.

**Uppercase and Lowercase:** FED/SAIL 🡪 fed/sail which is totally different from contrast.

**Synonyms and phonetic equivalence:** Soundex

* car = automobile color = colour
* Term equivalence by phonetic heuristics
* Developed by international police dept. to unify wanted criminal names differently registered in different countries. *Hermann* as *Herman* etc.
* Generate for each term phonetic hash so that terms with a similar sound have same hash code – *spandex algorithms*.

Lemmatization:

* Removing the inflectional endings only and,
* returns the *base/dictionary form* of the word (also called - *lemma*).
* The base form always has a meaning wrt the word.
* e.g.
* organizer, organizes, organizing 🡪 organize
* cars, car's, cars’ 🡪 car

Stemming:

* Here the terms are reduced to their “root”
* It is possible that after stemming the meaning of root can differ from original words.
* Eg. Automates, automatic, automation 🡪 automat
* Porter Algo:
* has 5 phases of word reductions applied sequentially.
* So each phase has set of rules which are selected and applied to longest suffix.
* It contains almost 60 suffixes \begin{example}
  \begin{tabular}[t]{lll@{\hspace{1in}}lll}
  \multicolumn{3}{l}{\te...
  ... & $\rightarrow$\ & & cats & $\rightarrow$\ & cat \\
  \end{tabular}\end{example}
* These rules are sensitive to the word length.
* Eg. If MEASURE>1 EMENT:
* Replacement → replac ;
* cement → cement; ~~cement → c~~ (this doesn’t happen)
* Lovins Stemmer:
* Longest suffix removal in single step
* Has 294 suffixes, each having their own exceptions
* If a word contains a suffix *s*, and if the word isn’t in the exception list, suffix *s* if removed from the word.
* Average benefit in Information Retrieval.
* Light weight computational load.

Performance in Different languages

Normalization, stemming help in IR?

* English – the results are inconsistent.
* Same word are used different context & due to preprocessing relevant information can be lost.
* Eg.
* Operate → oper (surgery context)
* Operational → oper (research context)
* Operating → oper (IT systems context)
* We saw more benefits in German, Spanish and Finnish language.
* Tools: WEKA, R, RapidMiner

Problems with Boolean Search Model

Pro:

* Very efficient processing algorithms for Boolean searches.

Cons:

* generate extreme results. (no results/too large results)
* results are not ordered/ranked.
* We only get exact matches, no proximity results.
* good for those expert users who know the text set and can formulate precise searches.
* It is challenging to express complex Boolean searches.

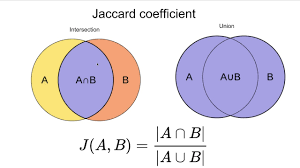
TEXT MINING 2

Scoring in Ranked Retrieval Models

Here the results are ranked on their relevancy wrt user query.

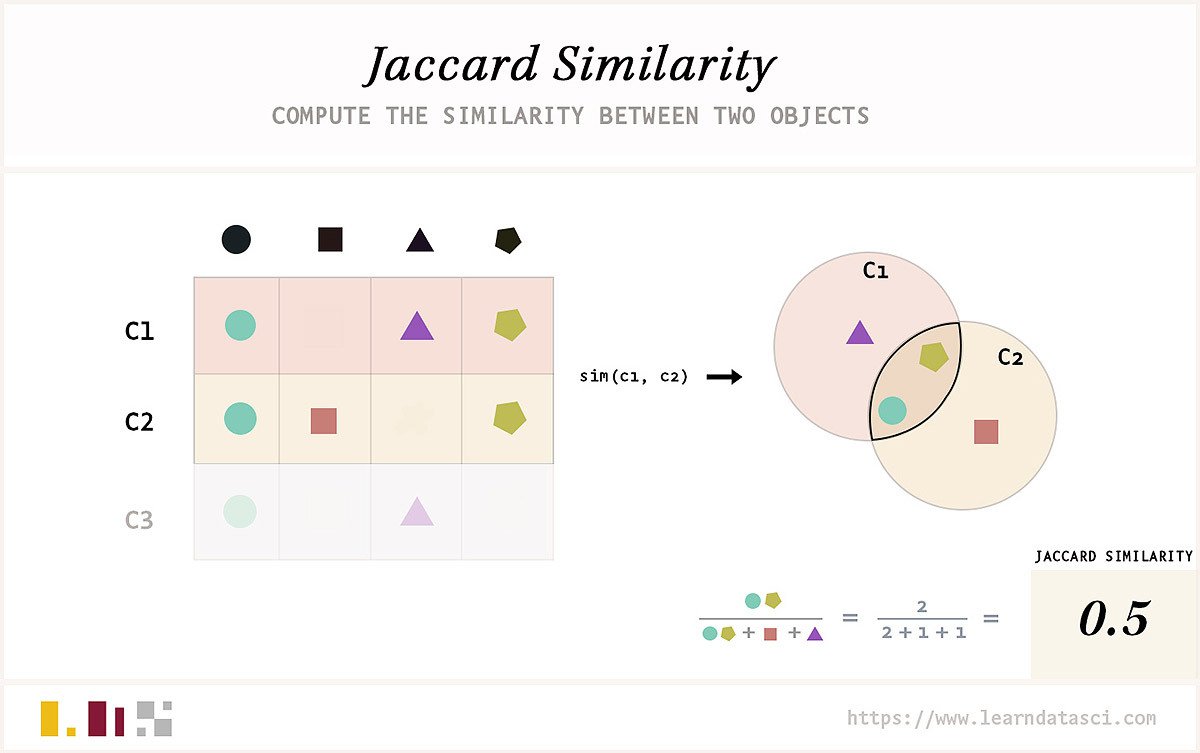
Range 0-1; 1→ max relevance → doc has more frequent terms.

Large number of results is no longer any problem because of ordering.

Jaccard Coeffcient:

Measures the grade of intersection of 2 sets.

Jaccard distance:

* which measures **dissimilarity** between sample sets
* JD(A,B) = 1 – J(A,B)

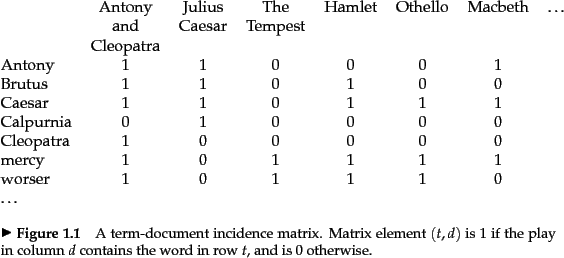
JD → doesn’t rely frequency of terms.

Rare terms are most informative than frequent terms, but Jaccard ignores this fact.

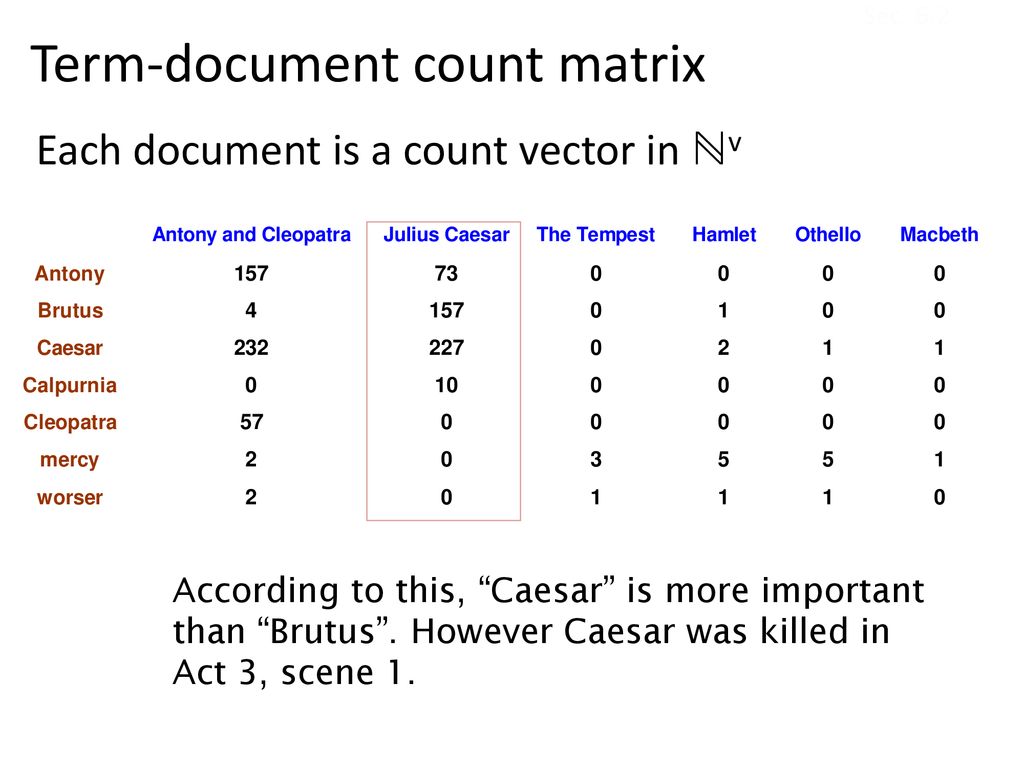
J & trigrams are suited for short texts.

Binary Matrix of terms-documents

Each doc is a binary vector



Term-doc Matrix with frequency



BOW – Bag of Words model

Previous models the vector representation ignores the word order in each doc.

Ie. Apple is sweeter than Orange & Orange is sweeter than Apple generate identical vectors.

There exist more complex versions of BOW model that includes the position of the words.

frequency of terms

*tf →* number of occurrence of terms in a document *d*.

used to determine which documents are more relevant for given query.

Doc with 10 occurrences of tf is more relevant, but not 10 times more relevant

The residency of go down when the document is represented using log.

Diagram

Description automatically generated with medium confidence

Relevance of terms

Rare terms are generally more important than frequent terms.

*tf* gives more relevance to frequent terms, which is not good.

The *tf* should be reweighted to give more importance to less freq. terms in doc.

*Logo

Description automatically generated with medium confidence*Inverse Document frequency (IDF)

*dft* is inv. Measure of information about the term *t* in doc *d*.

*dft*<= N (*number of docs in dataset*)

we use the log, as relevance is not inversely proportional to frequency.

*idf*,→ doesn’t take into account the repletion of term *t* in doc of corpus.

* Eg. *idf* with 1 million docs:

Table

Description automatically generated

**Solution:** Combining *tf* & *idf* – *tf-idf*

Text

Description automatically generated with low confidence

Conclusion

→ term relevance of any doc increases with number of occurrences *in doc*.

→ term relevance of any doc decreases with number of occurrences *of terms in all documents of corpus*.

**Relevance of doc *d* wrt query *q***:

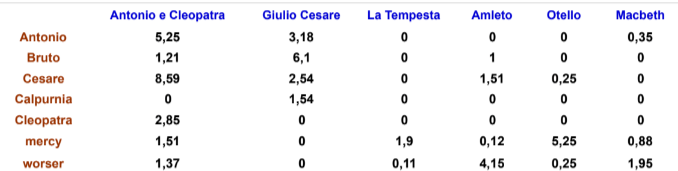
A picture containing icon

Description automatically generated

**tf-idf document of vectors of real:**

It’s high in multidimensionality space & sparsity.

* Each term → *axis*
* Each doc → *point*



**Query as vector in the doc space:**

Step 1: represent the query into vector.

Step 2: the relevance of doc wrt query is based on proximity (*vector similarity*) in space.

How to compute distances between the vectors?

* Euclidean distance among extremes,
* But this distance depends upon the length of vectors.

Chart, line chart

Description automatically generated**Problems with Euclidean distance.**

Here, *q* is query and *d* represent the documents.

A/c to ED: *q-d2*, and q*-d1, q-d3* are same. But in fact, *q-d2* should be prioritized.

Solution:

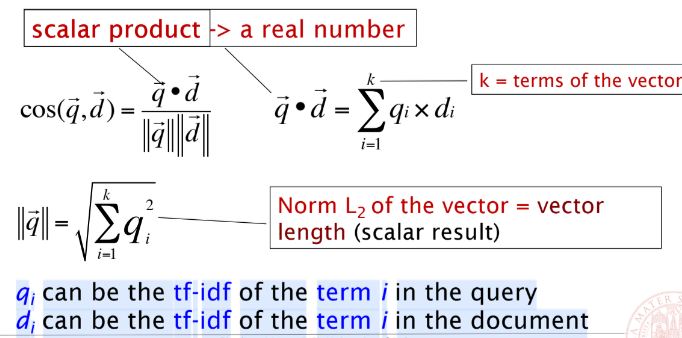
**Using Angles instead of distances.**

When we consider angles, 0o means max. similarity.

& angle 90\* is max. similarity.

**Using cosine(q,d):**

is the cosine angle between the vector *q* and *d*.

****

**Vector Normalization:**Vector can be normalized dividing each document by its norm L2.

We get a unit vector, this way docs of different sizes can be compared.

**Cosine with normalized vector:**

**Diagram

Description automatically generated with medium confidence**

Table, calendar

Description automatically generatedExample of similarity in 3 documents:

Table

Description automatically generatedTable

Description automatically generated

cos similarity b/w *SS and PP*:

789×0.832 + 0.515×0.555 +0.335×0.0 + 0.0×0.0 ≈ **0.94**

cos similarity b/w *SS and WH* ≈ **0.79**

cos similarity b/w *SS and WH* ≈ **0.69**

→ We can say SS ais more similar to PP.

Chart, background pattern, bubble chart

Description automatically generated

**Page rank: Relevance from links**

First algo used by Google.

– of the relevance of web pages from the *number* & *quality* of **entry links** rather than their content.

– by entry link, we assume that important websites have more links from other websites.

In the image we can see – C has higher PR, even if it has fewer connections, but its only connection is a very significant one (B).

It has 2 parts:  
1. Search process select the web page according to their content.

2. The web page results are ordered according to the pagerank algorithm.

**Lexical Matching:**

The similarity is computed on basis of LM.

Calculation: (counting the number of words) / (total number of words)

* 1. *Powerful car engine* vs *potent auto motor*
     + Meaning is equivalent.
     + But cosine similarity is zero.
  2. *Powerful math model* vs *powerful car model*
     + Meaning is different.
     + Cosine similarity is greater.

**Words:**

Represent ideas/objects/person/places.

Medium of communication.

Ferdnand de Saussure:

* Signifier: sound image
* Signified: concept

The meaning fo the words depend upon the relation of the words.

**Computational word meaning**

– eg. WordNet has 29 relations.

Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms called *'synsets'*, each expressing a distinct concept.

* Synsets are interlinked using conceptual-semantic and lexical relations such as hyponymy and antonymy.
* Hyponymy – type of relation with its hypernym.
* Bird (hyponymy) → crow, eagle, a seagull (hypernym)
* Antonymy – words with opposite meanings.
* Buy – sell.
* Synonym – words with similar meanings.
* Car – automobile, motorcar, etc.

**Drawbacks**:

Computationally expensive, limited dictionary.

Workaround: Word embeddings.

## Evaluation of results

There is need of information → converted to query → obtain search results.

Evaluate the search results → on basis of required information / not on the basis of search terms.

Relevance evaluation:  
Human experts evaluate for each query, whether the documents are relevant or not.

**Accuracy Evaluati**on

(Number of relevant doc retrieved + irrelevant not retrieved) / (all doc in corpus)

Table

Description automatically generated

* It is not sufficient for unbalanced datasets.

**Evaluation with precision and recall**

Precision:

→ Fraction of relevant docs retrieved

→ tp/(tp + fp)

Recall:

**→** Fraction ofrelevant docs that have been retrieved

**→** tp/(tp+fn)

F**-measure:** Combining *Precision* and *Recall*. The harmonic means of both.

Text

Description automatically generated

**→** here, *beta* can be changed accordingly.

→ standard case: *beta* = 1

→ when the *recall* is twice as important: *beta* = 2

It is possible to adjust F-score to give more importance to *precision* over *recall*, or vice versa.

Precision and recall go other ways, if one increases other decreases.

Chart, line chart

Description automatically generated**Breakeven point for *precision*-*recall*.**

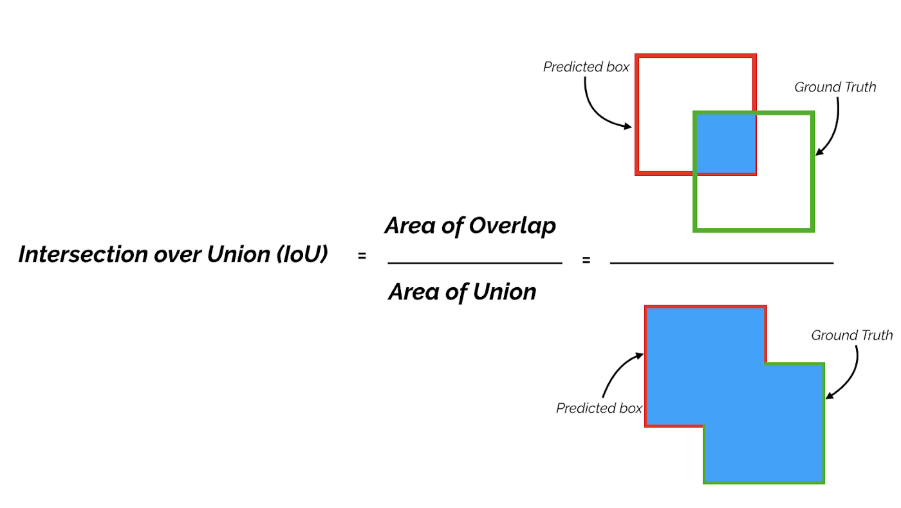
the point at which the precision-recall-curve intersects the bisecting line.

The area under the precision-recall curve (AUPRC) is another statistic to quantify the performance of a classifier.

**Further evaluation measure**

R-precision

→ The number of relevant docs within the first *r* documents divided by *r*.

Mean Average precision

→ The average of R-precision from 1 to mj relevant doc of qj

A picture containing text, clock

Description automatically generated→ taking the mean AP over all classes and/or overall IoU thresholds.

# DIMENSTIONALITY REDUCTION

Larger the numbers of docs → larger number of features.

So, we need to reduce the no of features, for ease of application to ML tools & algo.

Fewer features → less training time → less complex → more general.

Removing noisy features & features causing overfitting.

**Feature selection**

→ A process of reducing the number of input variables when developing a predictive model.

→ *tf* and *tf-idf* are one of the easiest way to select the features.

**Methods for FS:**

Method 1: mutual info

→ Dependency between variables.

→ Choose the terms with higher mutual information.

→ When mutual information is 0, terms are independent (*happens when the distribution of the term t is equivalent in the class and in the entire corpus of documents*).

Computing mutual information:

→ Based on max. likelihood estimation

→ N = N00 (docs not containing t, and not in class c)

+ N01 (docs not containing t, and are in class c)

+ N10  (docs containing t, and not in class c)

+ N11 (docs containing t, and are in class c)

Method 2: x2 test (chi-square)

→ Statistical test to examine diff. between categorical variables from a random sample to judge the goodness of fit between expected and observed results.

→ determine if a difference between observed data and expected data is due to chance, or if it is due to a relationship between the variables you are studying.

* In case of single categorical observation - χ²goodnessof fit test.

Allows testing whether the frequency distribution of the categorical variable is significantly different from the expectations. Often, but not always, the expectation is that the categories will have equal proportions.

Expectations in equal proportions:

*Null hypothesis:* bird species visiting the bird feeder are in = proportions.

*Alternative hypothesis:* bird species visit the bird feeder in ≠ proportions.

* In case of 2 measurements – X2 test of independence.

Feature selection by x2 test:

* \begin{equation*} X^2=\sum{\frac{(O-E)^2}{E}} \end{equation*}Χ2 is the chi-square test statistic
* Σ is the summation operator (it means “take the sum of”)
* *O* is the observed frequency
* *E* is the expected frequency

With ↑ difference between the O & E (*O* − *E*in the equation),↑ will be X2 will be.

To decide whether the difference is big enough to be significant, you compare the chi-square value to a critical value.

How to perform X2 test:

1. Create a table of O and E frequencies.
2. Calculate the chi-squared values
3. Find critical X2 value
4. Compare X2 value to critical value
5. Decide whether to reject the null hypothesis.

Chi-square test: accepting/rejecting the null hypothesis

A picture containing shape

Description automatically generatedConfidence value for a featute to represent a class c. confidence > 1 – *p*-value

Statistical Test X2 : Ranking of terms

X2 should be computed for each term, normalized, and then compared. In both cases given below, first k terms with larger values are selected.

|  |  |
| --- | --- |
| * Rank of each term is weighter avg of their X2 values wrt class. | Alternatively the rank of each term is its maximum value |
| Text  Description automatically generated | A picture containing text, watch  Description automatically generated |

Homogeneity statistical test based on X2

With X2, evaluate whether 2 sequences of values are homogeneous/not. (similar/not)

Statistical test X2: Limits

Test X2 is inadequate for small observed and expected values (data<30 or sum<200)

As no. of test performed on the problem ↑, the probability of total error ↑.

The ordering given by X2 to the terms is important.

**Impact of feature selection: Naïve Bayes with/without selection**

Diagram

Description automatically generated

Observation:

Binomial NB reaches its max with 10 selected features with mutual information.

* As no of features ↑, classification is effective ↓ because the doc representation with binary freq. becomes less distinguishable.

Multinomial NB takes more features (around 100) to reach max.

* Mutual info – 100 features.
* X2 reaches the same max value but with 300 features.
* X2 wrt MI gives more importance to rare terms. Hence needs more terms of MI in order to reach max.
* X2 selects better features.

A multinomial model with more frequent terms was selected for each category

Two modalities to compute the most frequent terms:

*Document frequency:* no of docs in class *c* that contain term *t*.

*Category frequency:* no of occurrences of term *t* in doc of class *c*.

Experiment with data set WebKB

Contains dataset containing web pages of 4 universities. 600 instances, 6 categories.

Chart, line chart

Description automatically generatedAs no. of selected terms with MI ↑, the binomial model, differently from multinomial its effectiveness ↓.

Conclusion on Naïve Bayes & Feature Selection

* Efficient in train/test phase.
* Linear cost wrt text quality.
* More effective in domains with numerous equally important features.
* Robust that other methods against irrelevant features.
* Resistant to concept drift (data that changes in classes over time)

=====

Feature extraction → a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.

**Counterproductive Features**

Suppose there is an irrelevant & unrelated word *w* with a particular word *W.* When trained *w* is wrongly associated with *W*.

# LATENT SEMANTIC ANALYSIS

Doc vector space model:

Pro

* Lexical matching among terms
* Ranking based on similarity measures ( cosine/ Jaccard )
* Several representations of terms and docs – binary frequencies/ term weighting with abs freq, log, *tf-idf*, etc.
* Using unsupervised ML algos.

Cons:

* Semantic problems in Natural Language. *Jaguar: car? animal?*
* *Polysemy*: a word with diff meanings depending on context.
* Vector space model of doc cannot recognize the diffrent meaning of the word itself.
* In case of *polysemy* the cosine similarity is > actual similarity. A picture containing icon

  Description automatically generated
* More irrelevant docs, reduction of precision
* *Synonymy*: terms lexically diff but with similar meaning.
* Vector space model of docs doesn’t represent the semantic association. i.e. wood is not related to the tree.

Diagram

Description automatically generated

**Latent Semantic Analysis/Indexing**

* an automated statistical method that determines the contextual meaning of any text by examining the relationships among word.
* words closer to their meaning are closer.

LSA: generate a new matrix using words present in docs in the corpus.

*rows* → represent the unique words present in each paragraph.

*col* → represents each paragraph.

Map matrix to new space, with lower dimensions, ignoring noises/ irrelevant details.

Usually, the content of the matrix is replaced by *tf-idf* score. Then reduction is performed.

LSA learns latent topics by performing matrix decomposition on doc-term matrix using SVD.

**SVD:**

Used for reducing no. of *rows* of the matrix. (*words*) while preserving the similarity structure among columns.

The rank r of matrix denotes no. of rows/cols vectors linearly independent.

Graphical user interface, text, application

Description automatically generatedSingular Value of Matrix

How many singular values ot select?

Chart, line chart

Description automatically generated

SVD applied to term-docs matrix: dimensionality reduction

Diagram

Description automatically generated with medium confidence

Low rank approximation

Text, letter

Description automatically generated

LSA: similarity among terms, docs, Query docs

Text

Description automatically generated

LSA = term and doc similarity

Text

Description automatically generated

How to calculate SVD of matrix A:

A picture containing text

Description automatically generated

* 1. find the eigen values of AAT

A picture containing text, clock, watch

Description automatically generated

* 1. Find singular value σi



* 1. Find the singular vectors by finding orthogonal set of eigen vectors of ATA.

*The eigenvalues of ATA are 25, 9, and 0*

* 1. We know ATA is symmetric and eigenvectors will be orthogonal.

A picture containing clock

Description automatically generatedLogo, company name

Description automatically generated

A picture containing diagram

Description automatically generated

For unit-length vector in kernel of *v1* =

Text

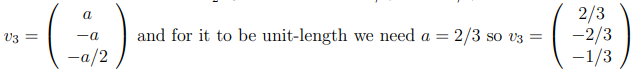
Description automatically generated

* 1. For the last eigenvector, we could compute the kernel of *ATA* or find a unit vector perpendicular to *v1* and *v2*.

A screenshot of a computer

Description automatically generated with medium confidence





* 1. So, at this point we know,

Diagram

Description automatically generated

Text

Description automatically generated

**Latent Semantic Index**:

* LSI uses - mathematical method that makes information retrieval more accurate.
* method working: identifying hidden contextual relationships b/w words.
* It may help you to break it down like this:
* Latent → Hidden
* Semantic → Relationships Between Words
* Indexing → Information Retrieval

LSA:

Left: original space → doc-term

Right: transformed new space → similar words are closer.

Diagram, timeline

Description automatically generated

LSA example:

Table

Description automatically generatedTable

Description automatically generatedTerm-doc Matrix U of terms

Table

Description automatically generatedTable

Description automatically generatedMatrix Σ Matrix V

Reducing dimension to 2:

Table

Description automatically generated

Original Matrix C and reduced C2:

Similarity among terms (outside LSA space)

A picture containing table

Description automatically generated

Similarity among Documents

A picture containing table

Description automatically generated

Example of Query without & with LSA

Calendar

Description automatically generated  
Chart

Description automatically generated

Diagram

Description automatically generatedLSA: effect of SVD factorization

LSA – Applications

Collaborative filtering and recommendation systems

* Understanding and predicting users’ interest from large dataset of behaviour/ purchase and choices.

Opinion mining and Sentiment Analysis

Speech to text & NLP

Powerful data clustering

LSA – Limitations

Cannot directly catch polysemy – multi-meaning words.

Each term - single vector in LSA space therefore one meaning.

“*chair of board*” & “*sit on the chair*” – chair is given same meaning by LSA.

Resulting term vector – average of diff. meanings of words in corpus.

Issue for words having more meanings. Their position is in middle of clouds.

Theoretical computation cost – SVD for matrix *m* x *n*

O(min(*mn*2, *m*2*n*)) = if n<<m → O(*mn*2), else O(*m*2*n*)

Limited expressivity due to BOW model.

# Text Classification: Rocchio, k-NN, Bayesian Algo, Evaluation methods

**IR to Text Classification:**

Standing queries: it is periodically executed on a collection to which new documents are incrementally added over time.

Text categorization/ Classification:

- the task of assigning predefined categories to free-text documents.

- provide conceptual views of doc collections & has important applications in the real world.

- supervised problem.

* we need pre-classified train docs
* methods to represent unstructured textual docs
* extract features (*curse of high dimensionality*)
* perform classification.

Diagram

Description automatically generated

**Multi label classification**

Binary classiciation – YES/NO

Multilabel-classification – FICTION/RELIGION/THRILLER/etc.

* chosen from a set of classes.
* Evaluation method is different for binary and multiclass.

Application: scientific docs/legal docs/musical tracks/movie genre/ genes/etc.

There are 2 solution groups:

**1.** Transforming the multilabel text cat. problem to single-label classification:

Chart

Description automatically generated with medium confidenceChart, table

Description automatically generatedA picture containing text, light

Description automatically generatedcan be handled using single-class classifiers.No. of possible classes to the power set 2|*C*|

**2.** Algorithm adaption methods: Instead of breaking into binary tasks, handle problem it full form.

Neural networks with multi-value output (*research paper*)

Generalization of AdaBoost, boosting algorithms (*research paper*)

Hybrid approaches which chain of classifiers (*research paper*)

* Incremental classification – for each instance determines sequentially the possible categories, decomposing the proble for each category in two sub problems.
* Text

  Description automatically generated
* Accuracy measures:
  + D – set of docs
  + C – set of categories
  + L(d) – set of actual categories of doc *d*.
  + L(d,c) = 1 if *c* is among actual categories of *d*, else 0
  + M(d) – set of predicted categories for *d*.
  + M(d,c) = 1 if model classifies *d* as class *c*, else 0.

for each doc no. of cat. correctly predicted, divided by no. of correct + wrong cat.

A picture containing text, watch, clock

Description automatically generated

No of doc correctly classified in each cat., divided by total no. docs -

Text

Description automatically generated

% of couples doc-category correctly predicted -

A picture containing text, watch

Description automatically generated

Precision

Text, letter

Description automatically generated

Recall

Text, letter

Description automatically generated

f-measure

Diagram, text, schematic

Description automatically generated

**Hierarchical Text Categorization:**classifying text documents into classes that are organized into a hierarchy.

- There is a arrangement of multiple binary, multi-class classifiers into a hierarchy

- where classification is executed from the top down.

- Each classification prediction results are sent to subsequent classifier, at the next

level down the tree, and so on until there are no more levels.

Diagram

Description automatically generated

* Each node of tree contains docs of same category
* Child node – docs dealing more deeply with same of father code.
* A doc may belong to multiple cat.

**Text categories and Search Engines**

Search eng. Use indexing for docs/webpages/images.