Document Classification

Statistical Analysis and Document Mining

Spring 2019

Vasilii Feofanov, Massih-Reza Amini

Université Grenoble Alpes

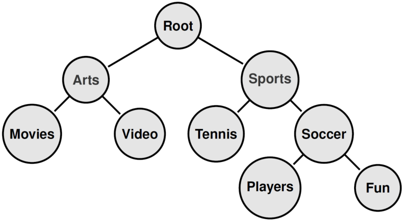
vasilii.feofanov@univ-grenoble-alpes.fr

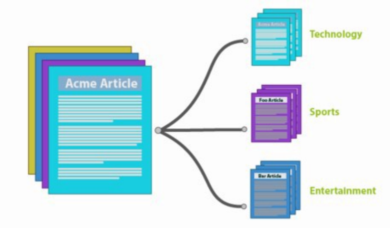
Outline

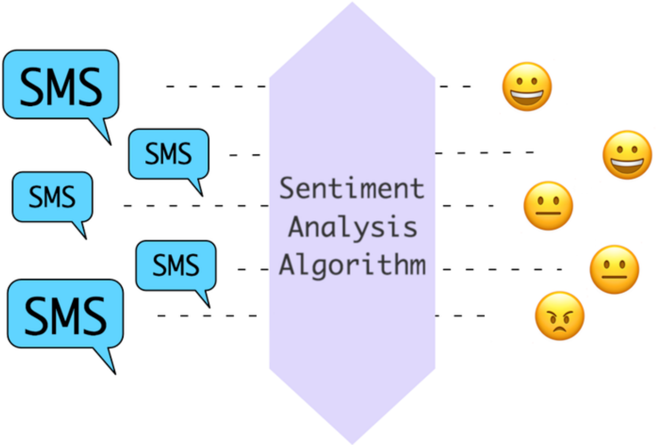
Automatic Document Classification

Application 1: Spam Detection

Application 2: Genre Classification



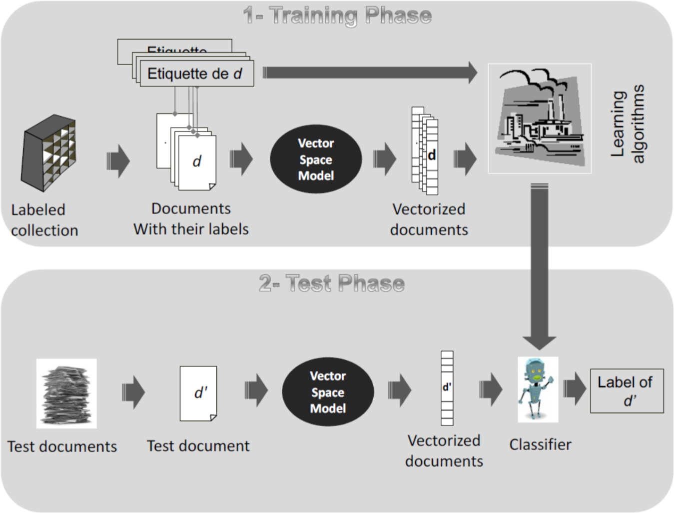



SMS
SMS
SMS
SMS
SMS
Sentiment Analysis Algorithm

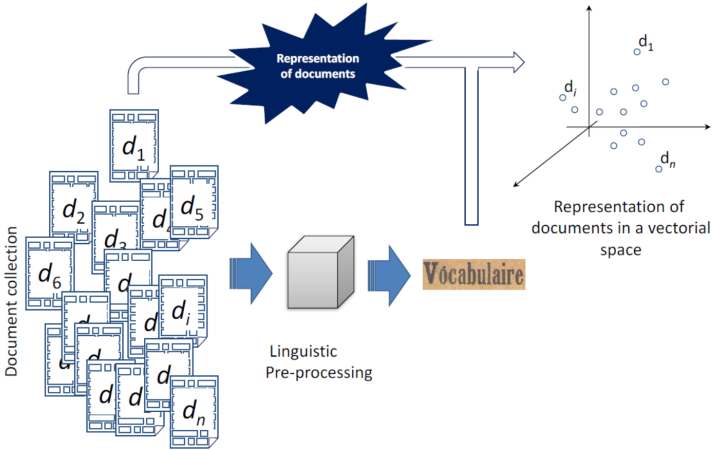

meamh
야옹
meong
mjav
miaou
mjau
мяу
ņau
miauw
mnau
meow
miyav
mjá
miao
mjay
ニャー
喵


?

Classification Scheme



From Document to Vector



Segmentation

- *Segmentation* (tokenization): separate a sequence of characters into semantic elements, or *words*.
- *Term* (type of words): class of all words having the same sequence of characters.
- *Example*:
  "The cat sat on the mat."
  Words: The, cat, sat, on, the, mat
  Terms: the, cat, sat, on, mat
- *Dificulty*: Tokenization is language specific.

Example: Segmentation In French

In French, the following issues may arise during the segmentation process:

- Lexical components with hyphens:
  chassé-croisé, peut-être, rendez-vous
- Lexical components with an apostrophe:
  jusqu'où, aujourd'hui, prud'homme
- Idiomatic expressions:
  au fait, poser un lapin, tomber dans les pommes
- Contracted forms:
  j', M'sieur, Gad'zarts (les gars des Arts et Métiers)
- Acronyms:
  K7, A.R., CV, càd, P.-V.

Normalization

1. *Textual normalization*: consists in reducing the words of a same family to their canonical forms.
   - Punctuation: suppression of points and hyphens;
   - Lower-upper case: transform all upper cases to lower cases;
   - Accents: suppression of accents.
2. *Linguistic normalization* consists in
   - Rooting: replace each word by its root;
   - Stemming: replace each word by its canonical form.

Examples of Preprocessing

Non-spam message before prepocessing

Subject: Re: 5.1344 Native speaker intuitions The discussion on native speaker

intuitions has been extremely interesting, but I worry that my brief intervention may have muddied the waters. I take it that there are a number of separable issues. The first is the extent to which a native speaker is likely to judge a lexical string as grammatical or ungrammatical per se. The second is concerned with the relationships between syntax and interpretation (although even here the distinction may not be entirely clear cut).

Examples of Preprocessing

Non-spam message before prepocessing

Subject: Re: 5.1344 Native speaker intuitions The discussion on native speaker

intuitions has been extremely interesting, but I worry that my brief intervention may have muddied the waters. I take it that there are a number of separable issues. The first is the extent to which a native speaker is likely to judge a lexical string as grammatical or ungrammatical per se. The second is concerned with the relationships between syntax and interpretation (although even here the distinction may not be entirely clear cut).

Non-spam message after prepocessing

re native speaker intuition discussion native speaker intuition extremely interest worry brief intervention muddy waters number separable issue first extent native speaker likely judge lexical string grammatical ungrammatical per se second concern relationship between syntax interpretation although even here distinction entirely clear cut

Preprocessed Documents from Different Classes

Non-spam message after prepocessing

re native speaker intuition discussion native speaker intuition extremely interest worry brief intervention muddy waters number separable issue first extent native speaker likely judge lexical string grammatical ungrammatical per se second concern relationship between syntax interpretation although even here distinction entirely clear cut

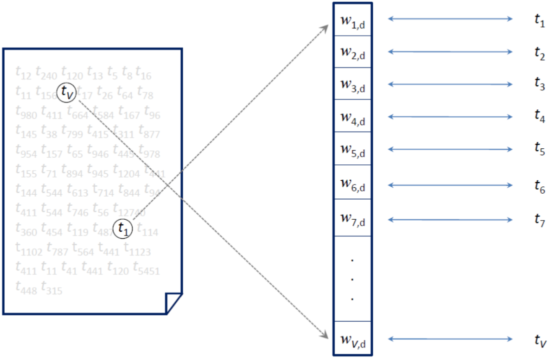
Spam message after prepocessing

financial freedom follow financial freedom work ethic extraordinary desire earn least per month work home special skills experience required train personal support need ensure success legitimate homebased income opportunity put back control finance life ve try opportunity past fail live promise

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

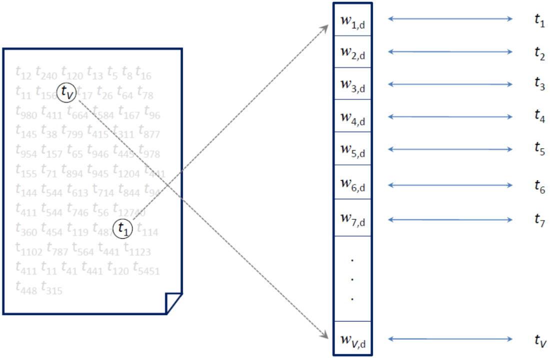
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

In the bag-of-words the word order in a sentence is ignored. Given a document, to each term t_j a specific value $w_{j,d}$ is assigned.



In the bag-of-words the word order in a sentence is ignored. Given a document, to each term t_j a specific value $w_{j,d}$ is assigned.

$\Rightarrow$ *A document can be represented as a vector:* $\mathbf{d} = (w_{j,d})_{j=1}^{V}$.



- Observation is a document $\mathbf{d} = (w_{j,d})_{i=1}^{V}$, where
  - V is a vocabulary size;
  - $w_{j,d}$ is a value (importance) corresponding to the term t_j.

- Observation is a document $\mathbf{d} = (w_{j,d})_{i=1}^{V}$, where
  - V is a vocabulary size;
  - $w_{j,d}$ is a value (importance) corresponding to the term t_j.
- Training set: $S = \{\mathbf{d}_i, y_i\}_{i=1}^{n}$, where
  - $\mathbf{d}_i \in \mathbb{R}^V$;
  - $y_i \in \{1, \ldots, K\}$.

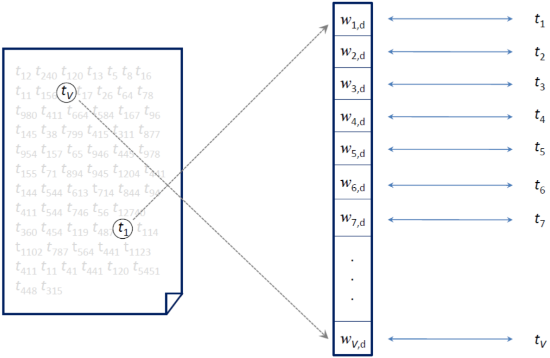
Document Classification Task

- Observation is a document $\mathbf{d} = (w_{j,d})_{i=1}^{V}$, where
  - V is a vocabulary size;
  - $w_{j,d}$ is a value (importance) corresponding to the term t_j.
- Training set: $S = \{\mathbf{d}_i, y_i\}_{i=1}^{n}$, where
  - $\mathbf{d}_i \in \mathbb{R}^V$;
  - $y_i \in \{1, \dots, K\}$.
- *Target*: minimise the misclassification error:

$$P(h(\mathbf{D}) \neq Y) = \sum_{c \in \{1,\dots,K\}} P(Y = c) P(h(\mathbf{D}) \neq c | Y = c).$$

A document is represented as a vector: $\mathbf{d} = (w_{j,d})_{j=1}^{V}$.

How we define importance $w_{j,d},\ j \in \{1, \dots, V\}$*?*



- If term t_j is present in the document, then $w_{j,d} = 1$. Otherwise, $w_{j,d} = 0$.

First Approach: Binary Values

- If term t_j is present in the document, then $w_{j,d} = 1$. Otherwise, $w_{j,d} = 0$.
- Suppose conditional independence of terms given a class.

First Approach: Binary Values

- If term t_j is present in the document, then $w_{j,d} = 1$. Otherwise, $w_{j,d} = 0$.
- Suppose conditional independence of terms given a class.
- *Question:* How then the term t_j contributes into classification?

First Approach: Binary Values

- If term t_j is present in the document, then $w_{j,d} = 1$. Otherwise, $w_{j,d} = 0$.
- Suppose conditional independence of terms given a class.
- *Question:* How then the term t_j contributes into classification?
  - For each class, we can compute the probability that the term is present in a document from this class.
    $\Rightarrow$ Bernoulli distribution.

Bernoulli Naive Bayes for Documents

- Let $W_{j,d}|Y = c$ be distributed acc. to *Bernoulli*. Then, probability that t_j is present in a document $\mathbf{d}$ is:

$$P(W_{j,d}|Y = c) = p_{t_j|c}.$$

Bernoulli Naive Bayes for Documents

- Let $W_{j,d}|Y = c$ be distributed acc. to *Bernoulli*. Then, probability that t_j is present in a document $\mathbf{d}$ is:

$$P(W_{j,d}|Y = c) = p_{t_j|c}.$$

- Naive assumption for documents:

$$P(\mathbf{D} = \mathbf{d}|Y = c) = \prod_{j=1}^{V} P(W_{j,d} = w_{j,d}|Y = c).$$

- Let $W_{j,d}|Y = c$ be distributed acc. to *Bernoulli*. Then, probability that t_j is present in a document $\mathbf{d}$ is:

$$P(W_{j,d}|Y = c) = p_{t_j|c}.$$

- Naive assumption for documents:

$$P(\mathbf{D} = \mathbf{d}|Y = c) = \prod_{j=1}^{V} P(W_{j,d} = w_{j,d}|Y = c).$$

- Bernoulli naive Bayes decision rule for the document classification task:

$$\begin{aligned} h_{NB}(\mathbf{d}) &:= \underset{c \in \mathcal{Y}}{\operatorname{argmax}} \, P(Y = c) \prod_{j=1}^{V} P(W_{j,d} = w_{j,d}|Y = c) \\ &= \underset{c \in \mathcal{Y}}{\operatorname{argmax}} \, P(Y = c) \prod_{j=1}^{V} p_{t_j|c}^{w_{j,d}} (1 - p_{t_j|c})^{1 - w_{j,d}}. \end{aligned}$$

Bernoulli NB: Train/Prediction

- Estimation of priors:

$$P(Y=c) \leftarrow \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_i=c) =: \frac{n_c}{n}.$$

Bernoulli NB: Train/Prediction

- Estimation of priors:

$$P(Y = c) \leftarrow \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i = c) =: \frac{n_c}{n}.$$

- Estimation of the Bernoulli parameters:

$$\hat{p}_{t_j|c} = \frac{\sum_{i=1}^{n} \mathbb{I}(w_{j,d_i} = 1 \wedge y_i = c)}{\sum_{i=1}^{n} \mathbb{I}(y_i = c)} =: \frac{\mathrm{df}_{t_j}(c)}{n_c}.$$

  - What happens if $\hat{p}_{t_j|c} = 0$?

Bernoulli NB: Train/Prediction

- Estimation of priors:
$$P(Y = c) \leftarrow \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i = c) =: \frac{n_c}{n}.$$
- Estimation of the Bernoulli parameters:
$$\hat{p}_{t_j|c} = \frac{\sum_{i=1}^{n} \mathbb{I}(w_{j,d_i} = 1 \wedge y_i = c)}{\sum_{i=1}^{n} \mathbb{I}(y_i = c)} =: \frac{\mathrm{df}_{t_j}(c)}{n_c}.$$
  - What happens if $\hat{p}_{t_j|c} = 0$?
- It would be better to estimate $\hat{p}_{t_j|c}$ by:
$$\hat{p}_{t_j|c} = \frac{\mathrm{df}_{t_j}(c) + 1}{n_c + 2}.$$

Bernoulli NB: Train/Prediction

- Estimation of priors:

$$P(Y=c) \leftarrow \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_i=c) =: \frac{n_c}{n}.$$

- Estimation of the Bernoulli parameters:

$$\hat{p}_{t_j|c} = \frac{\sum_{i=1}^{n}\mathbb{I}(w_{j,d_i}=1 \wedge y_i=c)}{\sum_{i=1}^{n}\mathbb{I}(y_i=c)} =: \frac{\mathrm{df}_{t_j}(c)}{n_c}.$$

  - What happens if $\hat{p}_{t_j|c}=0$?

- It would be better to estimate $\hat{p}_{t_j|c}$ by:

$$\hat{p}_{t_j|c} = \frac{\mathrm{df}_{t_j}(c)+1}{n_c+2}.$$

- Considering the log scale, we classify new $\mathbf{d}$ as follows:

$$h_{NB}(\mathbf{d}) := \operatorname*{argmax}_{c\in\mathcal{Y}} \ln\frac{n_c}{n} + \sum_{\substack{j=\{1,\dots,V\}\\ t_j\in d}} \ln\hat{p}_{t_j|c} + \sum_{\substack{j=\{1,\dots,V\}\\ t_j\notin d}} \ln(1-\hat{p}_{t_j|c}).$$

Second Approach: Term Frequencies

- We assume that more often term t_j appears in a document, more important the term is.

Second Approach: Term Frequencies

- We assume that more often term t_j appears in a document, more important the term is.
- Then, for each t_j, we compute the *term frequency*, i.e. the number of occurrences of t_j in the document:

$$w_{j,d} = \mathrm{tf}_{t_j,d}.$$

Second Approach: Term Frequencies

- We assume that more often term t_j appears in a document, more important the term is.
- Then, for each t_j, we compute the *term frequency*, i.e. the number of occurrences of t_j in the document:

$$w_{j,d} = \mathrm{tf}_{t_j,d}.$$

- Let N_d be the number of words in the document. Then, $\sum_{j=1}^{V} \mathrm{tf}_{t_j,d} = N_d$.

- We assume that more often term t_j appears in a document, more important the term is.
- Then, for each t_j, we compute the *term frequency*, i.e. the number of occurrences of t_j in the document:

$$w_{j,d} = \mathrm{tf}_{t_j,d}.$$

- Let N_d be the number of words in the document. Then, $\sum_{j=1}^{V} \mathrm{tf}_{t_j,d} = N_d$.
- *Naive* assumption: all N_d words are *i.i.d.* by the following law:

$$P(\text{word} = t_j) = p_{t_j|c},\ j = \{1, \dots, V\},\ \sum_{j=1}^{V} p_{t_j|c} = 1.$$

$\Rightarrow$ Then, document $\mathbf{d}$ is distributed acc. to multinomial distribution with parameters $(p_{t_j|c})_{j=1}^{V}$.

Multinomial Naive Bayes for Documents

- Let $\mathbf{D}|Y = c$ be distributed acc. to the *multinomial law*. Then, the likelihood of $\mathbf{D}$ to be from the class c is:

$$P(\mathbf{D}|Y = c) = \frac{N_d!}{w_{1,d}! \cdots w_{V,d}!} p_{t_1|c}^{w_{1,d}} \cdots p_{t_V|c}^{w_{V,d}}$$
$$\propto \prod_{j=1}^{V} p_{t_j|c}^{w_{j,d}}.$$

Multinomial Naive Bayes for Documents

- Let $\mathbf{D}|Y = c$ be distributed acc. to the *multinomial law*. Then, the likelihood of $\mathbf{D}$ to be from the class c is:

$$P(\mathbf{D}|Y = c) = \frac{N_d!}{w_{1,d}! \cdots w_{V,d}!} p_{t_1|c}^{w_{1,d}} \cdots p_{t_V|c}^{w_{V,d}}$$
$$\propto \prod_{j=1}^{V} p_{t_j|c}^{w_{j,d}}.$$

- Multinomial naive Bayes decision rule for the document classification task:

$$h_{NB}(\mathbf{d}) := \underset{c \in \mathcal{Y}}{\operatorname{argmax}}\, P(Y = c) \prod_{j=1}^{V} p_{t_j|c}^{w_{j,d}}$$
$$\propto \underset{c \in \mathcal{Y}}{\operatorname{argmax}} \ln P(Y = c) + \sum_{j=1}^{V} w_{j,d} \ln p_{t_j|c}.$$

Multinomial NB: Train/Prediction

- Estimation of priors:

$$P(Y = c) \leftarrow \frac{n_c}{n}.$$

Multinomial NB: Train/Prediction

- Estimation of priors:
$$P(Y=c) \leftarrow \frac{n_c}{n}.$$
- Estimation of the multinomial distribution parameters:
$$\hat{p}_{t_j|c} = \frac{\sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_j,d}}{\sum_{j'=1}^{V} \sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_{j'},d}}.$$

- Estimation of priors:

$$P(Y = c) \leftarrow \frac{n_c}{n}.$$

- Estimation of the multinomial distribution parameters:

$$\hat{p}_{t_j|c} = \frac{\sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_j,d}}{\sum_{j'=1}^{V} \sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_{j'},d}}.$$

- It would be better to estimate $\hat{p}_{t_j|c}$ by:

$$\hat{p}_{t_j|c} = \frac{\sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_j,d} + 1}{\sum_{j'=1}^{V} \sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_{j'},d} + V}.$$

Multinomial NB: Train/Prediction

- Estimation of priors:

$$P(Y=c) \leftarrow \frac{n_c}{n}.$$

- Estimation of the multinomial distribution parameters:

$$\hat{p}_{t_j|c} = \frac{\sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_j,d}}{\sum_{j'=1}^{V} \sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_{j'},d}}.$$

- It would be better to estimate $\hat{p}_{t_j|c}$ by:

$$\hat{p}_{t_j|c} = \frac{\sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_j,d} + 1}{\sum_{j'=1}^{V} \sum_{\substack{i=\{1,\dots,n\} \\ \text{s.t. } y_i=c}} \mathrm{tf}_{t_{j'},d} + V}.$$

- We classify new $\mathbf{d}$ by the following rule:

$$h_{NB}(\mathbf{d}) := \operatorname*{argmax}_{c\in\mathcal{Y}} \ln \frac{n_c}{n} + \sum_{j=1}^{V} w_{j,d} \ln \hat{p}_{t_j|c}.$$

Discussion

- What happens when the training documents of different size?
  - Bernoulli?

- What happens when the training documents of different size?
  - Bernoulli?
  - Multinomial?

Discussion

- What happens when the training documents of different size?
  - Bernoulli?
  - Multinomial?
- What is the main drawback of
  - the binary weighting?

Discussion

- What happens when the training documents of different size?
  - Bernoulli?
  - Multinomial?
- What is the main drawback of
  - the binary weighting?
  - the term frequency weighting?

- What happens when the training documents of different size?
  - Bernoulli?
  - Multinomial?
- What is the main drawback of
  - the binary weighting?
  - the term frequency weighting?
  - the bag of words representation?

Discussion

- What happens when the training documents of different size?
  - Bernoulli?
  - Multinomial?
- What is the main drawback of
  - the binary weighting?
  - the term frequency weighting?
  - the bag of words representation?
  - the naive Bayes classifier?

Outline

Data Storage

- Generally, there is access to a large collection of documents.

Data Storage

- Generally, there is access to a large collection of documents.
- At the same time, the vocabulary size is also large.

Data Storage

- Generally, there is access to a large collection of documents.
- At the same time, the vocabulary size is also large.
- *Problem:* Need to store a matrix of size $n \cdot V$.

Data Storage

- Generally, there is access to a large collection of documents.
- At the same time, the vocabulary size is also large.
- *Problem:* Need to store a matrix of size $n \cdot V$.
  - What is approximately the size of the dataset in GB, if $n = 200,000$, $V = 10,000$ and one entry needs 8 bytes?

Data Storage

- Generally, there is access to a large collection of documents.
- At the same time, the vocabulary size is also large.
- *Problem:* Need to store a matrix of size $n \cdot V$.
  - What is approximately the size of the dataset in GB, if $n = 200,000$, $V = 10,000$ and one entry needs 8 bytes?
  - Around 16 GB!

French Wikipedia Collection: Statistics

Variables	Values
# of documents in the collection	**1,349,539**
Total # of occurrences of words	696,668,157
Average # of words per document	416
Size of the pre-processed collection on the disk	4.6 GB
Total # of types of words	757,476
Total # of types of words after rooting	604,244
Size of the vocabulary	**604,244**
Average # of terms per document	**225**
Size of the collection after removing a stop-list	2.8 GB

LibSVM Format

- *Sparse matrix:* most of entries are zeros.

LibSVM Format

- *Sparse matrix:* most of entries are zeros.
- No need to store entries with zero values.

LibSVM Format

- *Sparse matrix:* most of entries are zeros.
- No need to store entries with zero values.
- *LibSVM format:* the observation is written in the following format:

y	index-value		index-value
2	5:0.356	...	9:1000
3	2:10.2	...	15:0.01

French Wikipedia Collection: Statistics

Variables	Values
# of documents in the collection	1,349,539
Total # of occurrences of words	696,668,157
Average # of distinct words per document	416
Size of the pre-processed collection on the disk	4.6 GB
Total # of types of words	757,476
Total # of types of words after rooting	604,244
Size of the vocabulary	604,244
Average # of terms per document	225
Size of the collection after removing a stop-list	2.8 GB

- The encyclopedia "Grand Robert" contains around 75,000 words. The most extensive record shows that the French language would contain about 700,000 words. Why in French Wikipedia we found even more words (757,476)?

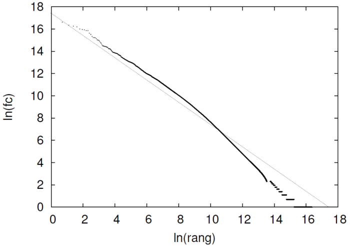
Two Laws of IR

- The encyclopedia "Grand Robert" contains around 75,000 words. The most extensive record shows that the French language would contain about 700,000 words. Why in French Wikipedia we found even more words (757,476)?
- When the collection was filtered by removing a stop-list ("a", "the", "of", etc.) of size 200 words, the average number of terms was reduced in documents from 416 to 225 (around 45% reduction). Why?
  In addition, their filtering reduces the space on the disk of about 39% (from 4.6 GB to 2.8 GB).

The number of occurrences $fc(m)$ of a word m in a document collection is inversely proportional to its rank:

$$\forall m: \ fc(m) \approx \frac{\lambda}{\text{rang}(m)}.$$

$\Rightarrow$ The k-th most frequent word is approximately k times less present than the most frequent one.



Rank	Word	Freq.	%
1	the	22,038,615	4.9%
2	be	12,545,825	2.79%
3	and	10,741,073	2.39%
4	of	10,343,885	2.3%
5	a	10,144,200	2.25%
6	in	6,996,437	1.56%
7	to (i.m.)	6,332,195	1.41%
8	have	4,303,955	0.96%
9	to (p.)	3,856,916	0.86%
10	it	3,872,477	0.86%

Top 10 frequent word from the 450 million word corpus (`https://www.wordfrequency.info`).

Filtering

- We suppress very frequent words that are present in all of the documents and that do not bring any information.

Filtering

- We suppress very frequent words that are present in all of the documents and that do not bring any information.
- *Example*:
  "The cat sat on the mat."
  Before filtering: the, cat, sit, on, mat
  After filtering: cat, sit, mat

Filtering

- We suppress very frequent words that are present in all of the documents and that do not bring any information.
- *Example*:
  "The cat sat on the mat."
  Before filtering: the, cat, sit, on, mat
  After filtering: cat, sit, mat
- How many frequent words we should suppress?

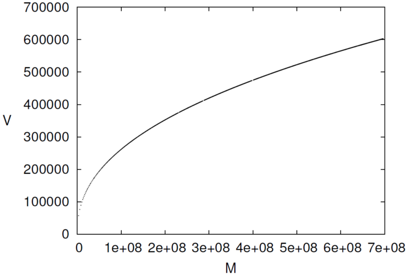
Heaps' Law

The size of the vocabulary V increases sub-linearly with respect to the number of words present in a collection M:

$$V = k \cdot M^{\beta},$$

where k and β are parameters that are dependent on the collection. Typically, in English text corpora $k \in [10, 100]$, and $\beta \in [0.4, 0.6]$.

$\Rightarrow$ Larger the collection size, larger the vocabulary size.



Exercise 1

You are willing to analyse a document containing 1,000,000 words.

- Let $k = 10$, $\beta = 0.5$. Following the Heaps' law, What is the number of distinct words in the document?

Exercise 1

You are willing to analyse a document containing 1,000,000 words.

- Let $k = 10$, $\beta = 0.5$. Following the Heaps' law, What is the number of distinct words in the document?
- It was found that the 7% of all words are "the" article. Following the Zipf's law, estimate the value of λ. What is the frequency of the second most frequent word? The third most frequent one?

Term Frequency vs Document Frequency

- With filtering of stopwords the term frequency weighting may work better.

Term Frequency vs Document Frequency

- With filtering of stopwords the term frequency weighting may work better.
- However, in many applications less frequent words also play a crucial role.

  Example: Medical Prescription vs Recipe.

take	water	glass	eat	wait	...	paracetamol	sugar	stomach
7	6	4	4	4	...	0	2	0
6	7	4	3	5	...	1	0	1

Term Frequency vs Document Frequency

- With filtering of stopwords the term frequency weighting may work better.

- However, in many applications less frequent words also play a crucial role.

  Example: Medical Prescription vs Recipe.

take	water	glass	eat	wait	...	paracetamol	sugar	stomach
7	6	4	4	4	...	0	2	0
6	7	4	3	5	...	1	0	1

- We want to diminish the weight of terms that occur frequently in general and increase the weight of terms that occur rarely in average.

tf-idf weighting is a trade-off between term frequency and document frequency:

- Normalised term frequency (tf part):

$$\frac{\mathrm{tf}_{t_j,d}}{\sum_{j=1}^{V} \mathrm{tf}_{t_j,d}} = \frac{\mathrm{tf}_{t_j,d}}{N_d}.$$

tf-idf weighting is a trade-off between term frequency and document frequency:

- Normalised term frequency (tf part):

$$\frac{\mathrm{tf}_{t_j,d}}{\sum_{j=1}^{V} \mathrm{tf}_{t_j,d}} = \frac{\mathrm{tf}_{t_j,d}}{N_d}.$$

- Inverse document frequency (idf part):

$$\ln \frac{n}{\mathrm{df}_{t_j}} := \ln \frac{n}{\sum_{i=1}^{n} \mathbb{I}(\mathrm{tf}_{t_j,d_i} \neq 0)}.$$

tf-idf weighting is a trade-off between term frequency and document frequency:

- Normalised term frequency (tf part):

$$\frac{\mathrm{tf}_{t_j,d}}{\sum_{j=1}^{V} \mathrm{tf}_{t_j,d}} = \frac{\mathrm{tf}_{t_j,d}}{N_d}.$$

- Inverse document frequency (idf part):

$$\ln \frac{n}{\mathrm{df}_{t_j}} := \ln \frac{n}{\sum_{i=1}^{n} \mathbb{I}(\mathrm{tf}_{t_j,d_i} \neq 0)}.$$

- Then, the tf-idf weight is defined as:

$$w_{t_j,d} = \frac{\mathrm{tf}_{t_j,d}}{N_d} \ln \frac{n}{\mathrm{df}_{t_j}}.$$

Exercise 2

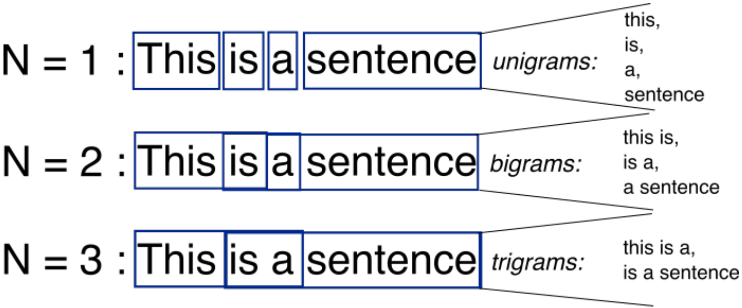
- We have the following training documents:
  d_1 : {"cat", "sit, "mat", "cat", "jump", "bed", "cat", "good", "sit"}

  d_2 : {"cat", "dog", "jump", "sit", "dog", "cat", "animal"}

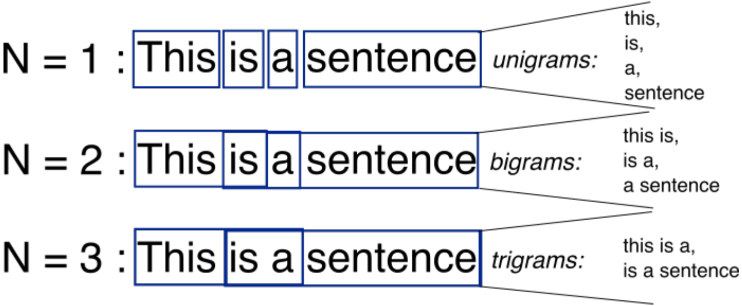
  d_3 : {"table", "sit", "write", "think", "good", "book", "dog", "sit"}

- What is the tf-idf representation for this dataset?

N-gram Representation

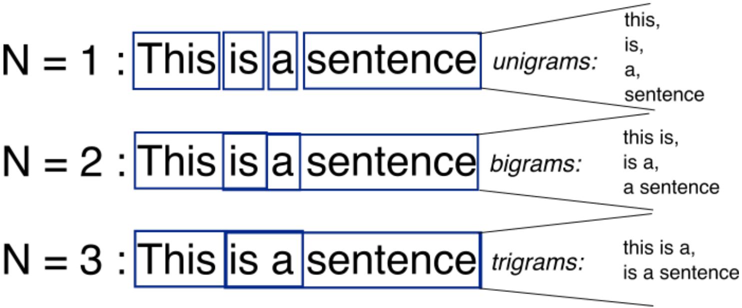


N-gram Representation



- Several words could be more important when they appear together.

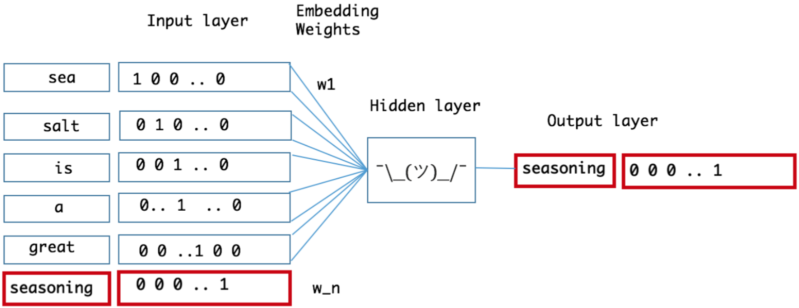
N-gram Representation



- Several words could be more important when they appear together.
- The approach is very costly when V is large.

...ATTACACGGTGACCAACCTATT...

Gram	Frequency
ATTA	4
GACC	3
CGGT	5
...	...


Input layer
Embedding
Weights
sea
1 0 0 .. 0
salt
0 1 0 .. 0
is
0 0 1 .. 0
a
0.. 1 .. 0
great
0 0 ..1 0 0
seasoning
0 0 0 .. 1
w1
w_n
Hidden layer
¯_(ツ)_/¯
Output layer
seasoning
0 0 0 .. 1