Text mining (II)

David Tomás Díaz
@d_tomas

Universidad de Alicante

Departamento de Lenguajes y Sistemas Informáticos

TRANSFORM 4EUROPE

Co-funded by the Erasmus+ Programme of the European Union

Contents

- Feature vectors
- Tokenisation
- Normalisation / pre-processing
- Weighting schema
- Word embeddings

Contents

- **Feature vectors**
- Tokenisation
- Normalisation / pre-processing
- Weighting schema
- Word embeddings

Feature vectors

- The data mining process
  - There is an initial dataset (instances/examples)
  - These instances should be in a format that is suitable to be processed by a data mining algorithm
    - Every **instance** is represented by a **feature vector**
    - **Features** (attributes) are variables that correspond to the properties of the instances

Features

Instances

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa

Feature vectors

- Bag-of-words representation
  - A.k.a. *unigrams* or *1-grams*
  - The most common way of representing text
  - Simple and useful for many text mining tasks
  - Any text item *D* (document) is represented as list of terms *t* (*features*) and associated weights *w*

$$D = [(t_1;w_1),(t_2;w_2),\ldots,(t_n;w_n)]$$

t_i = keywords or content-descriptors

w_i = measure of the importance of a term in representing the information contained in the document

Feature vectors

- Bi-grams, tri-grams, … n-grams
  - Extract all of two, three… *n* words in a row in the text

The brown fox jumps over the lazy dog →

Bi-grams
- *The brown*
- *brown fox*
- *fox jumps*
- *jumps over*
- *over the*
- *the lazy*
- *lazy dog*

Tri-grams
- *The brown fox*
- *brown fox jumps*
- *fox jumps over*
- *jumps over the*
- *over the lazy*
- *the lazy dog*

Feature vectors

- For a different representation, a feature may be something else
  - POS-tags
  - Noun phrases
  - Multi-words
  - Synsets
  - Named Entities
  - …

Feature vectors

As a text mining expert you must decide the most useful features for each task

Feature vectors

- Characteristics of the feature vectors
  - A given dataset may be drawn from a set of about 100.000 words…
  - … but a given text document may contain only a few hundred of them
  - Text data is sparse and high dimensional
  - Generally feature vectors are very sparse
    - Large number of features, most of them only occurring rarely
    - Most of the values are 0
    - High proportion of noisy and irrelevant features

Feature vectors

- How to create the feature vector
  - Tokenisation
  - Normalisation / Pre-processing
  - Weighting schema

Contents

- Feature vectors
- **Tokenisation**
- Normalisation / pre-processing
- Weighting schema
- Word embeddings

Tokenisation

- Identify individual words (tokens)

The Netherlands earned sweet revenge on Spain on Friday at the Fonte Nova in Salvador, hammering Spain 5-1 to put an emphatic coda on their loss in the 2010 World Cup finals.

↓

The
Netherlands
earned
sweet
…

Tokenisation

Let's practice!

https://bit.ly/3NQR1h9

Contents

Normalisation / pre-processing

- Convert words into normalised forms
  - Lower-case

    The → the ; NASA→ nasa; Claude Shannon → claude shannon

  - Lemmatisation (to basic forms)

    jumps → jump ; jumping → jump; jumped → jump

  - Stemming (mechanically remove/change suffixes)

    computer → comput ; computation → comput; compute → comput

Normalisation / pre-processing

- Stopword removal
  - Eliminate common words (e.g., and, of, the, …)

> *The Netherlands earned sweet revenge on Spain on Friday at the Fonte Nova in Salvador, hammering Spain 5-1 to put an emphatic coda on their loss in the 2010 World Cup finals.*

  - The SMART stopword list is widely used (571 words)

> *a, a's, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, ain't, all, allow, allows, almost, alone, along, already, also, although, always, am, among…*

Normalisation / pre-processing

Let's practice!

https://bit.ly/3NQR1h9

Normalisation / pre-processing

- Feature selection
  - Keep just the most indicative words in the vocabulary
  - Remove noises or abnormalities
  - Feature selection may make a particular algorithm feasible (some algorithms cannot deal with e.g. 1,000,000 features)
  - Different (automatic) approaches
    - Term Frequency
    - Document Frequency
    - Information Gain
    - Mutual Information
    - Chi-square (χ^2)
    - …

Contents

- Feature vectors
- Tokenisation
- Normalisation / pre-processing
- **Weighting schema**
- Word embeddings

Weighting schema

- Measure of the importance of a term in representing the information contained in the document
- Every term in the feature vector is assigned a numeric representation
  - Term occurrence
  - Term Frequency (TF)
  - Inverse Document Frequency (IDF)
  - TF-IDF
  - …

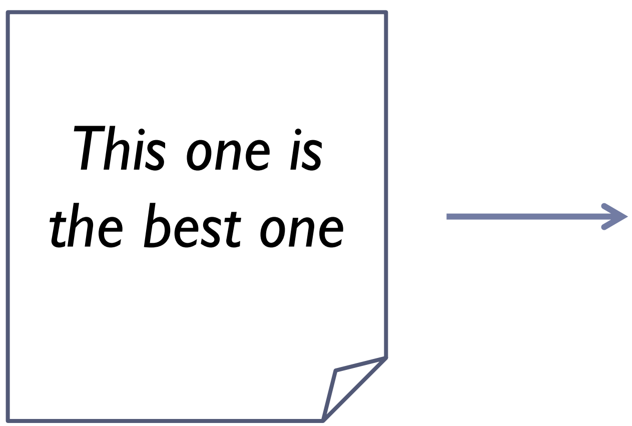
Weighting schema

- Term occurrence
  - Binary assignment
  - Terms occur (1) or do not occur (0) in the document

Term	Weight
best	1
dog	0
is	1
one	1
over	0
the	1
this	1

Weighting schema

- Term Frequency (TF)
  - Assumption: repeated words are strongly related to content
  - TF_i = number of times that term t_i appears in the document



Term	Weight
best	1
dog	0
is	1
one	2
over	0
the	1
this	1

Weighting schema

- Inverse Document Frequency (IDF)
  - Assumption: uncommon terms are more important
  - IDF_i = the inverted rate of documents that contain term t_i against the whole set of documents

Term *best* occurs in 3 out of 10 documents in the dataset

$$IDF_{best} = 10/3 = 3.33$$

Term *one* occurs in 8 out of 10 documents in the dataset

$$IDF_{one} = 10/8 = 1.25$$

Weighting schema

- TF-IDF
  - High value to frequent words that appear only in few documents
  - Combination of *TF* and *IDF* (*TF-IDF* = *TF* · *IDF*)

This one is the best one

Term	TF	IDF	TF-IDF
best	1	3.33	3.33
dog	0	5	0
is	1	1.11	1.11
one	2	1.25	2.5
over	0	10	0
the	1	1	1
this	1	1.43	1.43

Weighting schema

Let's practice!

https://bit.ly/3NQR1h9

Contents

- Feature vectors
- Tokenisation
- Normalisation / pre-processing
- Weighting schema
- **Word embeddings**

Word embeddings

- Definition
  - *Word embeddings* is a way to represent words as vectors of real numbers
  - The representation is based on the use of words in context
  - Words with a similar meaning are represented with a similar vector
  - The vector contains dozens or hundreds of dimensions (instead of thousands or millions)
  - This vectors are obtained using different methods (such as neural networks)

Word embeddings

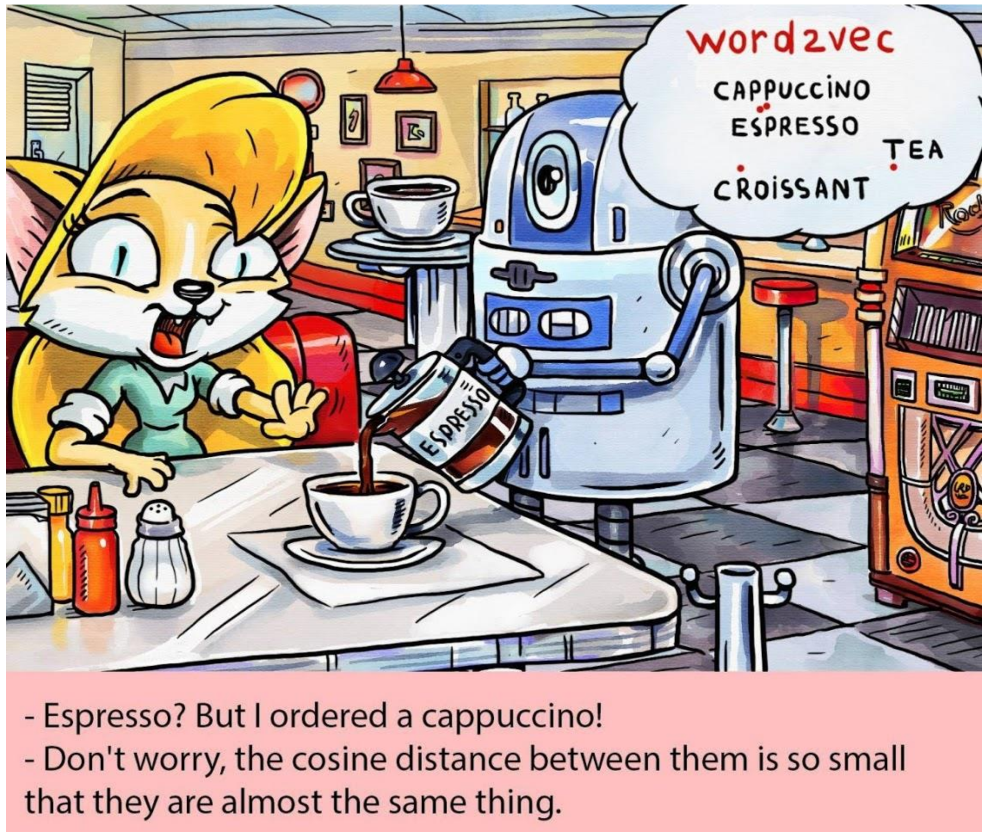
- Word2vec
  - *Word2vec* is a statistical method to efficiently learn *word embeddings* from a corpus of documents
  - Allows learning vectors with more dimensions from larger corpus (billions of words)
  - Words sharing a similar context in the corpus are close in the vector space defined

```
king - man + woman = queen
```

"king is to queen as man is to woman"

Word embeddings

- Word2vec



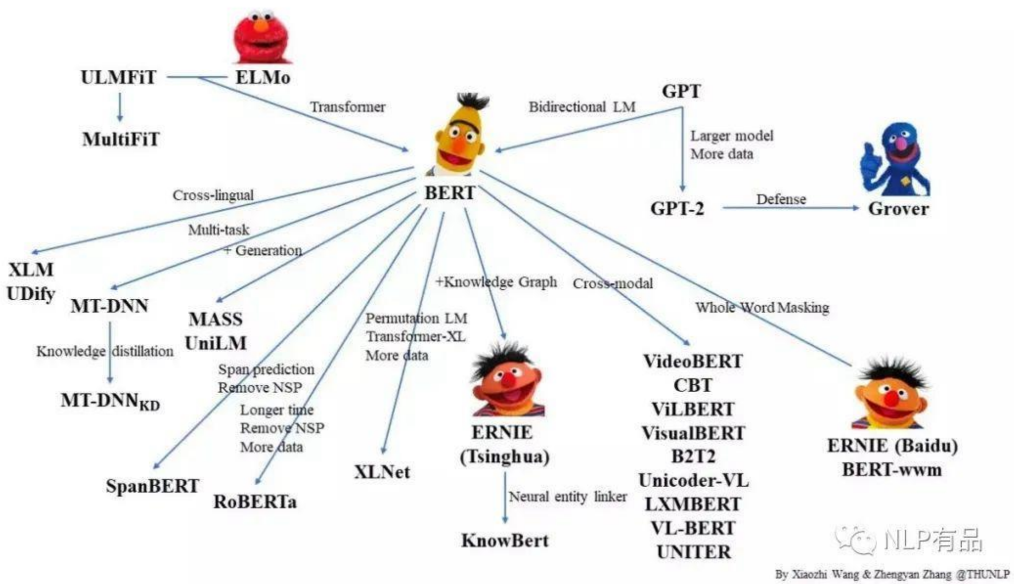
- Espresso? But I ordered a cappuccino!
- Don't worry, the cosine distance between them is so small that they are almost the same thing.

Let's practice!

https://bit.ly/3NQR1h9

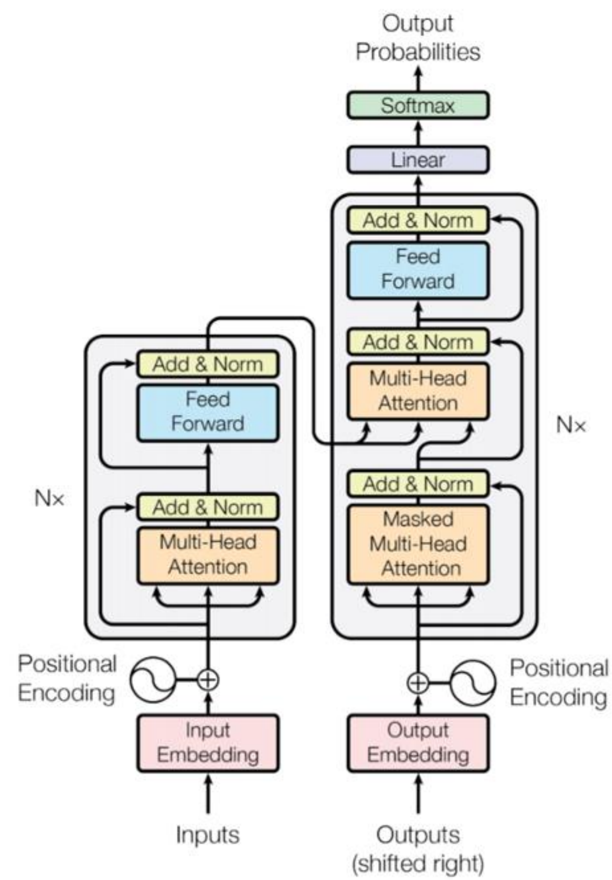
Word embeddings

- Contextual word embeddings
  - Create different vectors for the same Word, depending on the sense in the context
  - Ex. 'table' has a vector when it stands for 'furniture' and another vector when it stands for 'tabular data'



Word embeddings

- Contextual word embeddings
  - Transformers
    - Novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies
    - Attention allowed us to focus on parts of our input sequence while we predicted our output sequence
    - Have become the state-of-the-art in many natural language processing task (and also in other modalities such as image and video processing)



Let's practice!

https://bit.ly/3NQR1h9

@d_tomas

David Tomás Díaz