Vector Space Semantics and Information Retrieval

Jean-Cédric CHAPPELIER
Martin RAJMAN

LIA

Reminder: *Textual* Data Classification

- What is classified? (what objects?)
  - authors (1 object = several documents)
  - documents
  - paragraphs
  - "words"(/tokens) (vocabulary study, lexicometry)
- How to represent the objects? (what features?)
  - document indexing
  - choose the textual units that are meanigfull
  - choice of the metric/similarity

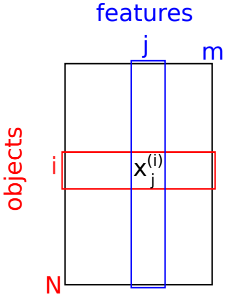
☞ preprocessing: "unsequentialize" text, suppress (meaningless) lexical variability

Frequently: lines = documents, columns = "words" (tokens, words, *n*-grams)

Jean-Cédric Chappelier & Martin Rajman

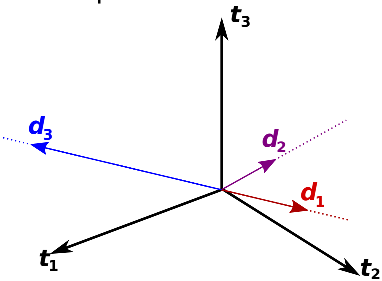
Starting point (reminder)

N "row" objects (e.g. documents) $x^{(i)}$ characterized by m "features" (e.g. "words") $x_j^{(i)}$



$x_j^{(i)}$ = "importance" of feature j for object i

Vector space model:



- **indexing** tokens/words define the axis
- documents are point in the vector space

Jean-Cédric Chappelier & Martin Rajman

Vector Space model

Objective

Representing documents as vectors derived from the distributions of indexing terms in the document collection.

Principle

- V, a finite vocabulary of **indexing terms**
- R : **representation space**
- $\mathscr{R}_D : V^* \to R$ **representation function**
- **similarity:** $\mathscr{M}_{\text{prox}} : R \times R \to \mathbb{R}^+$

Definition

Representation: translating a document (words) into computable data (numbers) adequate to the task (typically: that capture semantics)

Indexing: selecting relevant elements (features) to support the representation

Jean-Cédric Chappelier & Martin Rajman
EPFL

Indexing and representation of documents

(Pre-)processing tools:

- Tokenization
- Part-of-Speech tags
- Stemming and lemmatization
- Stop words
- frequencies (Zipf and Luhn)
- Bag of words model

Indexing terms

Choose (see next slides) a subset of the input tokens and keep only those:

Example

Now so long, Marianne
it's time that we began
to laugh and cry and cry
and laugh about it all again.

V, a finite vocabulary: `aardvark, begin, cry, information, laugh, long, Marianne, retrieval, time, ...`

☞ Now so long Marianne it's time that we began to laugh and cry and cry and laugh about it all again.

In practice

the vocabulary is several thousands of terms large

Jean-Cédric Chappelier
& Martin Rajman
EPFL

Tokenization (reminder)

Definition

Tokenization: splitting the text into words (Pre-requisite to choosing indexing terms)

Example

- easy: whitespaces

  Now is the winter of our discontent
  Made glorious summer by this son of York

- less easy: space not always indicative of a term segmentation (compounds):
  Distributional Semantics Information Retrieval and Latent Semantics Indexing performance comparison
- agglutinative languages are a problem: *Rinderkennzeichnungs- und Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz*
- Technical terms

  (e.g. Methionyl-glutaminyl-arginyl-tyrosyl-glutamyl-seryl-leucyl-phenyl-alanyl-alanyl-glutaminyl-leucyl-lysyl-glutamyl-arginyl-lysy-glutamyl-gycyl-alanyl-phenyl-alanyl-valyl-prolyl-phenyl-alanyl-valyl-threonyl-leucyl-glycyl-aspartyl-prolyl-glycyl-isoleucyl-glutamyl-glutaminyl-seryl-leucyl-lysyl-isoleucyl-...)

Jean-Cédric Chappelier
& Martin Rajman

Word Entities

Definition

Semantic entity: compound word (group of tokens) bearing a semantic meaning

Example

- "Information retrieval"
- "rendez-vous"
- "radio antenna"
- "Singing Lily" (a type of pastry)
- "Dolphin striker" (a spar [part of boat])

Stemming and lemmatization

Definition

Stem: morphological root of a word.
Stemming: Process of reducing words to their *stem*.
Lemmatization: better informed (e.g. PoS tag) choice of the root form

Example

- prepaid, paid $\longrightarrow$ paid
- interesting, uninteresting $\longrightarrow$ interest

Stemming: non-trivial process

factual $\longrightarrow$ fact	OK
equal $\longrightarrow$ eq	wrong (“eq” is too short)

Benefits

- Reduces lexical variability $\Rightarrow$ reduces index size
- Increases information value of each indexing term

Jean-Cédric Chappelier & Martin Rajman
EPFL

Choice of indexing terms

Filtering

Choice of indexing terms using various filters:

- on morpho-syntactic categories
- on stop-words
- on frequencies

Benefits

- more informative indexes
- smaller indexes (tractability)

Indexing terms: filtering with Morpho-syntactic categories

PoS-tag filtering

Some morpho-syntactic categories (e.g. determiners, conjunctions, ...) do not have much semantic content, so others (e.g. nouns, verbs, ...) do!

☞ Keep only the terms in a selected set of morpho -syntactic categories (e.g. nouns, adjectives and verbs)

Jean-Cédric Chappelier
& Martin Rajman

Stop words

Definition

Stop word: term explicitely to be excluded from indexing.

Example

stoplist: `the; a; 's; in; but; I; we; my; your; their; then`

Young men's love then lies
Not truly in their hearts, but in their eyes.

Indexed document: `Young men love lies truly hearts eyes`

Benefits

- cheap way to remove classes of words without semantic content

Problems

To be or not to be

☞ this sentence would be entirely removed.

Jean-Cédric Chappelier
& Martin Rajman
EPFL

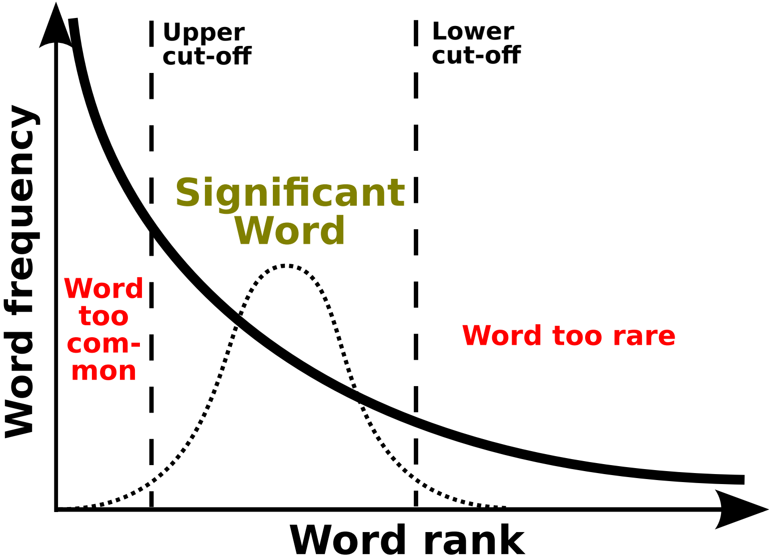
Indexing terms: filtering with frequencies

Zipf and Luhn

If r is the rank of a term and n is its number of occurrences (frequency) in the collection:

- Zipf (1949): $n \sim 1/r$
- Luhn (1958): mid-rank terms are the best indicators of topics

Choice of indexing terms: frequencies



Desequentialisation: bag of words model

Assumption

Positions of the terms are ignored. Term distribution is indicative enough of the meaning.

Model

$$d_1 = \{(t_1, n(d_1, t_1)); (t_2, n(d_1, t_2)); \ldots\}$$

$$d_2 = \{(t_1, n(d_2, t_1)); (t_2, n(d_2, t_2)); \ldots\}$$

A document is a multiset of terms

Example

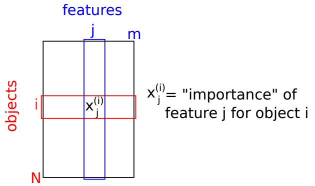
Now so long, Marianne ; it's time that we began
to laugh and cry and cry ; and laugh about it all again.

→ (`[begin,1] [cry,2] [laugh,2] [long,1] [Marianne,1] [time,1]`)

Conclusions on indexing

- Bad indexing can ruin the performances of an otherwise sophisticated system
- Good indexing is anything but trivial

Representation function



Objective

To represent documents as vectors, we need to associate the indexing terms with **weights**

Example

Now so long, Marianne
it's time that we began
to laugh and cry and cry
and laugh about it all again.

R representation space: $\mathbb{R}^{|V|}$

☞ ([aardvark,?] [begin,?] [cry,?] [information,?] [laugh,?] [long,?] [Marianne,?] [retrieval,?] [time,?])

Weighting schemes: tf, tf.idf

Term Frequency

$\mathrm{tf}(w_i, d_j) =$ nb of occurrences of term w_i in document d_j

Sometimes $1 + \log(\mathrm{tf}(w_i, d_j))$ is used in place of $\mathrm{tf}(w_i, d_j)$

Term Frequency - Inverse Document Frequency

$$\text{tf-idf}(w_i, d_j) = \mathrm{tf}(w_i, d_j) \cdot \mathrm{idf}(w_i)$$

with

$$\mathrm{idf}(w_i) = \log\left(\frac{|D|}{nb(d_k \supset w_i)}\right)$$

$|D|$: number of documents
$nb(d_k \supset w_i)$: number of documents which contain term w_i

Jean-Cédric Chappelier & Martin Rajman

Weighting

Example

Now so long, Marianne
it's time that we began
to laugh and cry and cry
and laugh about it all again.

$\mathscr{R}_D : V^* \rightarrow R$ representation function: here: Term Frequency
☞ ([aardvark,0] [begin,1] [cry,2] [information,0] [laugh,2] [long,1] [Marianne,1] [retrieval,0] [time,1])

$$\longrightarrow (0\ 1\ 2\ 0\ 2\ 1\ 1\ 0\ 1\ \ldots)$$

In practice

the vector is **very sparse** ☞ **dimension reduction**

Jean-Cédric Chappelier & Martin Rajman

Proximity measure between documents

Cosine similarity

$$\cos(\mathbf{d_1}, \mathbf{d_2}) = \frac{\mathbf{d_1}}{||\mathbf{d_1}||} \cdot \frac{\mathbf{d_2}}{||\mathbf{d_2}||} = \frac{\sum_{j=1}^{N} d_{1j}\, d_{2j}}{\sqrt{\left[\sum_j {d_{1j}}^2\right] \left[\sum_j {d_{2j}}^2\right]}}$$

- bounded ($0 < \cos(\mathbf{d_1}, \mathbf{d_2}) < 1, \forall \mathbf{d_1}, \mathbf{d_2}$)
- it is a similarity: the greater, the more similar the documents (as opposed to a *metric*)
- **independent on the length of the document**

Note: choose similarity measure well behaved for the representation (depends on the representation)

☞ see other similarity measures/metrics in last week lecture

Proximity measure between documents

Document 1

- Now so long, Marianne, it's time that we began to laugh and cry and cry and laugh about it all again.
- `...,[long,1] [Marianne,1] [time,1] [begin,1] [laugh,2] [cry,2],...`
- $\mathbf{d_1} = (\ldots,1,1,1,1,2,2,\ldots)$

Document 2

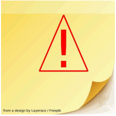
- I haven't seen Marianne laugthing for some time, is she crying all day long ?
- `...,[long,1] [Marianne,1] [time,1] [begin,0] [laugh,1] [cry,1],...`
- $\mathbf{d_2} = (\ldots,1,1,1,0,1,1,\ldots)$

Example

$$\cos(\mathbf{d_1},\mathbf{d_2}) = 7/(\sqrt{12}\cdot\sqrt{5}) = 0.904$$

Jean-Cédric Chappelier & Martin Rajman

Information Retrieval (IR)



Example of a task making use of Vector Space Semantics: Information Retrieval

Definition

selection of documents relevant to a query in an unstructured collection of documents.

- **unstructured**: not produced with IR in mind, not a database.
- **document**: here, natural language text (but could also be video, audio or images)
- **query**: utterance in natural language (possibly augmented with commands)
- **relevant**:
  1. users-wise: answering the IR requirements
  2. mathematically: maximizing a defined “proximity measure”

Jean-Cédric Chappelier
& Martin Rajman
EPFL

Ambiguity

Sometimes unintended results occur

Example

query: "*Chicago school*"

wanted?

- schools in Chicago (IL)?
- body of works in sociology?
- architectural style?
- where to learn how to play Chicago (game):
  - bridge?
  - or poker??

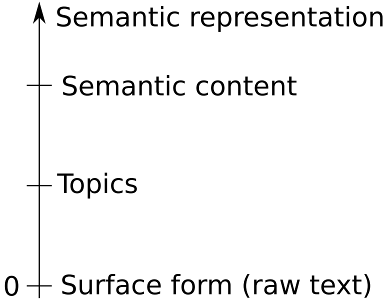
Jean-Cédric Chappelier
& Martin Rajman
EPFL

Relevance? Content versus topic

"*Relevant*" documents:

What does "*relevant*" mean?

- useful?
- new?
- topically related?
- content related?
  - at word level?
  - at semantic/pragmatic level?



Relevance? Content versus topic

Semantic content:
what the document talks about (topic) *vs* what it says (content).

Example

Document 1:

Note how misty the river banks are.

Document 2:

She got misty by the river of bank notes falling on the table.

Document 3:

Money had never interested her.

Doc. 1 & 2 have similar word content but are not topically related.
Doc. 2 & 3 have similar topics but opposite semantic content.

Jean-Cédric Chappelier & Martin Rajman
EPFL

How is IR done?

Tasks

- have the computer **represent documents** (at the adequate level): preprocessing, indexing, ...
- **represent the queries**, not necessarily the same way as documents (short queries, operators, . . .)
- **define a relevance measures** between representations

Similarities with other NLP tasks

- Classification (no query)
- Data mining (formatted data)
- Information extraction (retrieve *shorts parts* of documents)

Jean-Cédric Chappelier
& Martin Rajman

Okapi BM25 weighting scheme

More ad-hoc weighting scheme used in IR

BM25 weight for term *t* in document *d*

$$w^{\mathrm{BM25}}(t,d) = \frac{(k+1)}{k(1+b(\frac{dl}{avdl}-1))+\mathrm{tf}(t,d)} \cdot \text{tf-idf}(t,d)$$

with *dl* = document length
avfl = average document length

BM25 is a very good model and used as reference for comparison with new models

Jean-Cédric Chappelier
& Martin Rajman
EPFL

Queries: definition

Definition

Queries (or "topics") are "questions" asked to the system

- Typically *keywords*
  possibly augmented with operators: `dreamt WITHIN 5 philosophy`
- Supposed unknown at indexing time
  (difference between IR and classification where classes are known a priori)

See `https://trends.google.com/` for real-life examples

Jean-Cédric Chappelier & Martin Rajman

Query representation

Example

- easy: as for documents

  `more things in heaven and earth`

- less easy (verbatim sentence)

  `"more things in heaven and earth"`

- quite different from the document (positional information)

  `dreamt WITHIN 5 philosophy`

Key point

Query representation is not necessarily trivial (not always the same as representation of documents).

Jean-Cédric Chappelier
& Martin Rajman

Problem with short queries

Web queries

On the web,

- the average query length is under three words
- very few users use operators

Language being ambiguous, three-word queries are difficult to satisfy.

Solutions

- *query expansion*: use knowledge about the query terms to associate them with other terms and improve the query.
- *query term reweighting*: weight the terms of the query as to obtain maximum retrieval performance.
- *relevance feedback*: User provides the system an evaluation of the relevance of its answers.

Evaluation of IR systems

Referential:

- document collection
- set of queries
- list of documents from the collection to be retrieved for one given query

Metrics (reminder):

Precision

Precision is the proportion of the documents retrieved by the system that are relevant (according to the referential)

Recall

Recall is the proportion of the relevant documents which were retrieved by the system

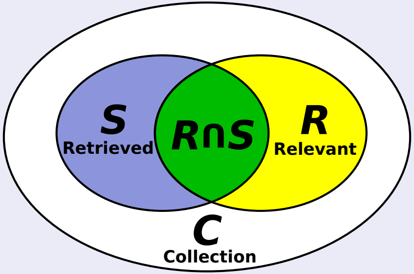
- Precision can be cheated by returning no document
- Recall can be cheated by returning all documents

Jean-Cédric Chappelier & Martin Rajman

Reminder: precision and recall

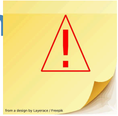
Given an IR system, a document collection and a referential; for a query q, the results returned by the system is evaluated with:

- Precision: $\Pr(q) = \frac{|R(q) \cap S(q)|}{|S(q)|}$
- Recall: $\mathrm{Rec}(q) = \frac{|R(q) \cap S(q)|}{|R(q)|}$



Jean-Cédric Chappelier
& Martin Rajman

Other performance measures: P@n and R-Precision



P@n

Precision at n document (for a given query q):

$$\Pr_n(q) = \frac{|R(q) \bigcap S_n(q)|}{|S_n(q)|}$$

with $S_n(q) =$ first n documents retrieved by the system (for query q)

R-Precision

precision obtained after retrieving as many documents as there are relevant documents, averaged over queries (N: total number of queries)

$$\text{R-Precision} = \frac{1}{N} \sum_{i=1}^{N} \Pr_{|R(q_i)|}(q_i)$$

Other performance measures: Mean Average Precis

Average Precision

Average of the precisions whenever all relevant documents below rank $\mathrm{rk}(d,q)$ are retrieved:

$$\mathrm{AvgP}(q) = \frac{1}{|R(q)|} \sum_{d \in R(q)} \mathrm{Pr}_{\mathrm{rk}(d,q)}(q)$$

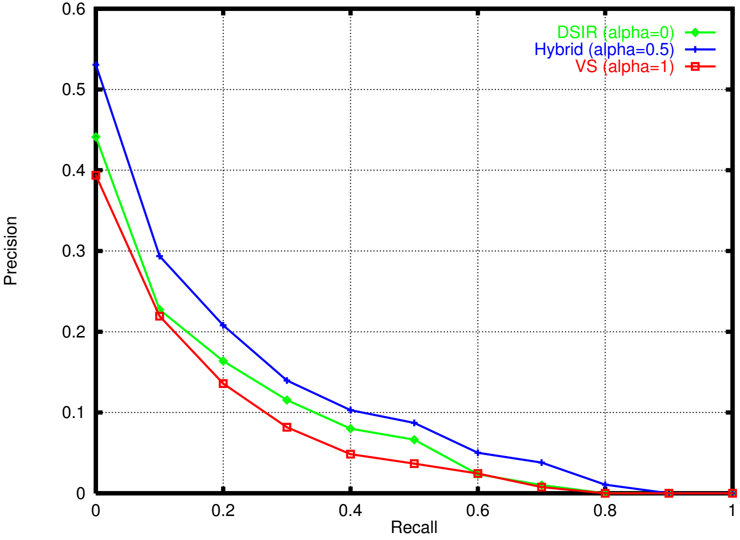
Mean Average Precision (MAP)

Mean over the queries of the Average Precisions

$$\frac{1}{N} \sum_i \mathrm{AvgP}(q_i)$$

MAP measures the tendency of the system to retrieve relevant documents first.

Plotting average Precision and Recall



Aim of the game: push the curve towards the upper right corner

Limitations

Problem

Basic vector space model has problems notably with

- Polymesy
- Synonymy

Polymesy

Example

Query includes term `Bank`
→ Bank of England? Bank of fishes? Grand bank? Airplane bank?

Consequences

Negative impact on precision

Jean-Cédric Chappelier
& Martin Rajman

Synonymy

Example

Query includes term `freedom`
$\rightarrow$ *liberty* will not be seen as relevant

Consequences

Negative impact on recall

Jean-Cédric Chappelier & Martin Rajman

Topic-based models

Idea

Apply a transformation to the representation space as to emphasize the most relevant features: index senses rather than mere words

☞ try to get **more dense** (less sparse) representation

Note

Part of the indexing (in particular stemming) is already a step in this direction (less dependent on mere words)

Jean-Cédric Chappelier & Martin Rajman

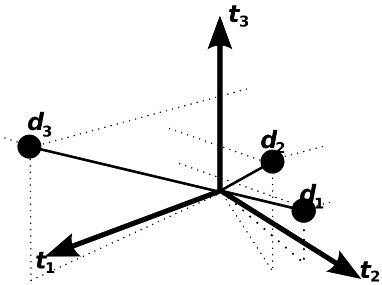
Latent Semantic Indexing

Reduction of dimensionality of the original representation space

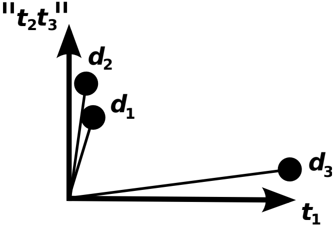
- approximation of the occurrence matrix
- filtering of the occurrence matrix

LSI Idea

Create a matrix close to the occurrence matrix but of smaller rank (= PCA)



Before



After

Latent Semantic Indexing

Advantages

- More significant representation

Drawbacks

- Out-performed by other models
- Too expensive to compute on large bases (requires iterative methods)
- Meaning of the axis (indexing features): ??
- IR: Projection of queries is problematic

Other more advanced Topic Models

LDA: Latent Dirichlet Allocation (Blei, Ng, Jordan 2003)
(not to be confused with Linear discriminant analysis!!)

☞ probabilistic model with hidden states (“topics”) making use of Dirichlet priors

Reference:

- D. Blei. Probabilistic topic models. Communications of the ACM, 55(4):77–84, 2012.
- J.-C. Chappelier, Topic-based Generative Models for Text Information Access, In Textual Information Access – Statistical Models, E. Gaussier and F. Yvon eds, ch. 5, pp. 129-178, Wiley-ISTE, April 2012.

Word vectors (a.k.a. word embeddings)

Key ideas:

- make use of more abstract/algebraic representation of **words**:

  use "**word embeddings**":

  go from sparse (& high-dimensional)
  to **dense** (& less high-dimensional) representation of documents,
  combining "embeddings" and dimension reduction operations:
  a bit like K-means and non-linear PCA at the same time (and several times)

- Learning Word Representations

  Typical NLP: Corpus $\xrightarrow{\text{some algorithm(s)}}$ tokens/words/n-grams/phrase vectors $\xrightarrow{\text{further processing}}$ ...

  Key idea in recent approaches: could we do the first step(s) **task independent**?

  so as to then reduce whatever NL P(rocessing) to some algebraic vector manipulation:

  no longer start "core (NL)P" from words anymore,
  but from vectors (learned once for all) that capture general syntactical and semantic information

Jean-Cédric Chappelier & Martin Rajman
EPFL

From "word" vectors to "word" embeddings

embedding = vectorial representation + dimension reduction

from sparse ($m \simeq 10^4$–10^5) to dense (=more compact) representation ($m \simeq 10^2$–10^3)

Why should dense vectors be better?

- More efficient (lower dimension: less data to handle, store, estimate, ...)
- capture “the essence” (capture statistical invariants): less noisy?
  (☞ generalize better)

How to? → Distributional Semantics

Idea (dates back to Harris (1954) and Firth (1957))

There is a high degree of correlation between the observable **co-occurrence** characteristics of a term and its **meaning**

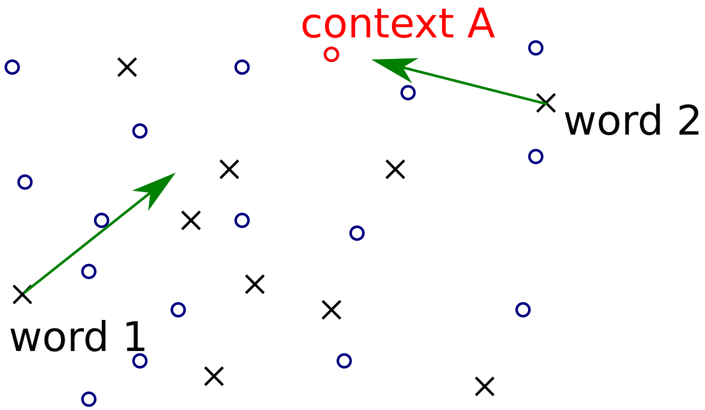
Example

- Some X, for instance, naturally attack rats.
- The X on the roof was exposing its back to the shine of the sun.
- He heard the mewings of X in the forest .
- X is a: ...

Typically, word embeddings are trained by "predicting a word based on its context" (or vice-versa) from a large (unlabeled) corpus

Jean-Cédric Chappelier
& Martin Rajman
EPFL

Key idea: illustration



Word Embeddings

"*Word embedding*":

- numerical representation of "words"(/"tokens")
  a.k.a. "*Semantic Vectors*", "*Distributional Semantics*"
- **Objective**: relative similarities of representations correlate with syntactic/semantic similarity of words/phrases.
- Two **key ideas**:
  1. representation(**composition** of words) = vectorial-composition(representations(word))
     for instance: representation(phrase) $= \sum_{\text{word} \in \text{phrase}}$ representation(word)
  2. remove **sparseness**, compactify representation: dimension reduction
- have been around *for a long time*
  Harris, Z. (1954), "*Distributional structure*", Word 10(23):146–162.
  Firth, J.R. (1957), "*A synopsis of linguistic theory 1930-1955*", Studies in Linguistic Analysis. pp 1–32.

Word Embeddings: different techniques

"*Many recent publications (and talks) on word embeddings are surprisingly oblivious of the large body of previous work* [...]"
(from `https://www.gavagai.se/blog/2015/09/30/a-brief-history-of-word-embeddings/`)

Main techniques:

- co-occurrence matrix; often reduced (PCA, Hellinger-PCA (2013), GloVe (2014))
- probabilistic/distribution (DSIR, LDA)
- shallow (Mikolov et al. 2013) or deep Neural Networks (ELMo)

There are theoretical and empirical correspondences between these different models [see e.g. Levy, Goldberg and Dagan (2015), Pennington et al. (2014), Österlund et al. (2015)].

Jean-Cédric Chappelier
& Martin Rajman
EPFL

Word embedding “geometry”

- The geometry of embeddings should account for desired properties (e.g. syntactic, semantics, synonymy, word classes, ...)

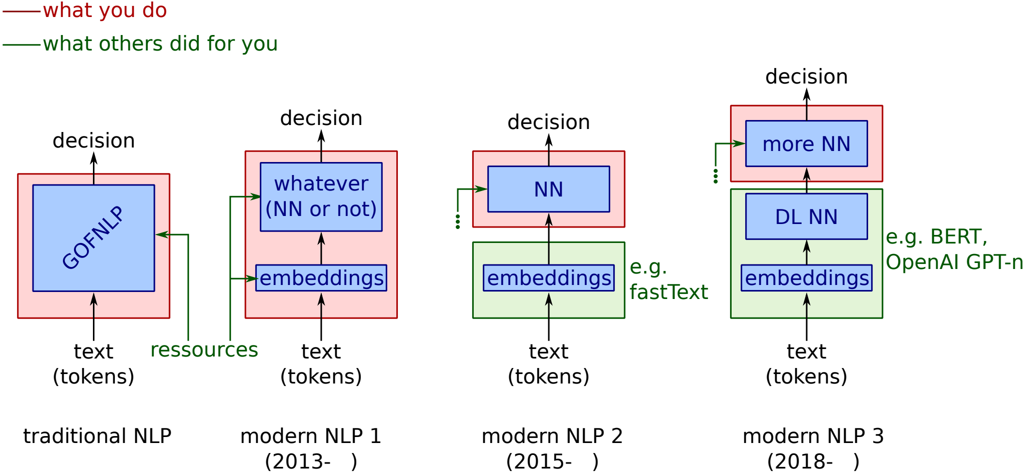
  e.g. predict new word representation (embedding) from some linear combination of embeddings of words around it

- Word embedding indeed exhibit some semantic compositionality

  Some theoretical justification for this behavior was given by Gittens et al. (2017): words need to be uniformly distributed in the embedding space.

A. Gittens et al. (2017), "*Skip-Gram – Zipf + Uniform = Vector Additivity*", proc. ACL.

NLP evolution



Corpus-based linguistics: the evolution

- (Before corpora ($<$ 1970): introspection and hand written rules)
- First wave ($\simeq$ 1980-2015): probabilistic models (HMM, SCFG, CRF, ...)
- Neural-nets (NN) and "word" embeddings (1986, 1990, 1997, 2003, 2011, 2013+):
  - MLP: David Rumelhart, 1986
  - RNN: Jeffrey Elman, 1990
  - LSTM: Hochreiter and Schmidhuber, 1997
  - early NN Word Embeddings:
    Yoshua Bengio et al., 2003; Collobert & Weston (et al.) 2008 & 2011
  - word2vec (2013), GloVe (2014)
  - ...
- Transfer learning (2018–):

  ULMFiT (2018), ELMo (2018), BERT (2018), OpenAI GPT2 (2019), GPT3 (2020)

  beyond "word" embeddings: **pre-trained** early layers to feed the later layers of some NN to some (shallow?) task-specific architecture that is trained in a supervised way

Jean-Cédric Chappelier
& Martin Rajman

Summary / Keypoints

- Vector-space model
- Indexing (and its important role)
- Weighting schemes, tf-idf
- Evaluation: Precision and Recall.

Jean-Cédric Chappelier
& Martin Rajman

References

[1] C. D. Manning, P. Raghavan and H. Schütze, "*Introduction to Information Retrieval*", Cambridge University Press. 2008.

[2] R. Baeza-Yates and B. Ribeiro-Neto, "*Modern Information Retrieval*", Addison Wesley, 1999.

[3] "*Topics in Information Retrieval*", chap. 15 in "Foundations of Statistical Natural Language Processing", C. D. Manning and H. Schütze, MIT Press, 1999.

Word Embeddings: some references

R. Lebret and R. Collobert (2013), "*Word Emdeddings through Hellinger PCA*", proc. EACL.

T. Mikolov et al. (2013a), "*Distributed Representations of Words and Phrases and their Compositionality*", proc. NIPS.

T. Mikolov et al. (2013b), "*Efficient Estimation of Word Representations in Vector Space*", proc. ICLR.

J. Pennington, R. Socher, and C. D. Manning (2014) "*GloVe: Global Vectors for Word Representation*", proc. EMNLP.

O. Levy, Y. Goldberg and I. Dagan (2015), "*Improving distributional similarity with lessons learned from word embeddings*", Journ. Trans. ACL, vol. 3, pp. 211-225.

Österlund et al. (2015) "*Factorization of Latent Variables in Distributional Semantic Models*", proc. EMNLP.

A. Joulin et al. (2017), "*Bag of Tricks for Efficient Text Classification*", proc. EACL.

P. Bojanowski et al.(2017), "*Enriching Word Vectors with Subword Information*", Trans. ACL, vol. 5.

A. Gittens et al. (2017), "*Skip-Gram – Zipf + Uniform = Vector Additivity*", proc. ACL.

E. P. Matthew et al. (2018), "*Deep Contextualized Word Representation*", proc. NAACL.