07 – Topic Identification, Topic Modeling

IA161 Natural Language Processing in Practice

Zuzana Nevěřilová

NLP Centre, FI MU, Brno

October 31, 2023


wrong kiss attentive
sit nice busy minute
friendly friend
waitress fast meal fun
enjoy drink fry dinner
ambiance edamame lunch
quick tasty tempura salad
yummy
yellowtail sashimi shrimp
happy nigiri kimono pack
wait night day visit town two high seat
leave start drive
live hand long fire staff year
last first strip
people spot cut local joint
flavor bar big full week worth top new
table dish henderson area
super japanese portion
hot rice serve another huge large offer
fantastic couple item
salmon specialty better review
excellent bad awesome
spot
meal rice
dish
option
flavor portion
ambiance piece
attentive
area

Topic modeling

- organize, summarize, and understand large collections of documents with no a priori knowledge
- discover unknown topical patterns in collection of documents
- dimensionality reduction – instead of taking into account every word in the document, take into account only words representing the document topics
- topic – group of related words representing concepts ($\rightarrow$ document tagging)
- statistical, unsupervised modeling

Topic Modeling and Topic Classification

topic modeling – find document representation by discovering topics present in the document + how much they are present (e.g. 10% horror, 70% fun, 25% Australia, 30% nature)
topic classification – categorize documents into a set of (predefined) topics

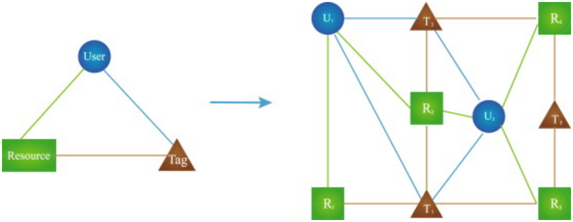
- supervised method
- best approach is to train for a specific set of documents, e.g.,
  - cluster company documents into invoices, contracts, purchase orders, delivery notes, other
  - cluster customer emails into customer complaints, request for contract end, relocation notice, other

Topic Modeling – Applications

- recommender systems
- document classification (one or more categories a document fits into)
- bio-informatics (interpret biological data)
- chatbots, topic tracking in dialogues
- document summarization (via topic names, a document is seen as a collection of topics, each with a weight)

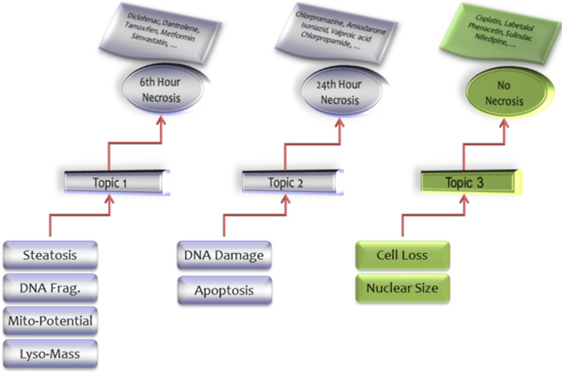
Recommender Systems

- recommend the best product for the user
- clusters of users, based on preference
- clusters of products
- Netflix prize



Bio-informatics

- categorize patients into risk groups based on text protocols
- detect common genomic features based on gene sequence data
- group drugs by diagnosis



Topic Modeling Approaches

- Latent Semantic Analysis, Latent Semantic Indexing (LSA/LSI) – matrix factorization
- Probabilistic Latent Semantic Analysis (pLSA) – probabilistic decomposition
- Latent Dirichlet Allocation (LDA) – iterative probabilistic method
- other decomposition techniques (e.g., Non-negative Matrix Factorization, NMF)
- other clustering techniques (e.g., k-means of word vectors)

Latent Semantic Analysis

Works because the distributional hypothesis works.

> *...words that occur in the same contexts tend to have similar meanings*
>
> *(Harris, 1954)*[1]

LSA computes how frequently words occur in:

- documents
- the whole corpus

... and assumes that similar documents have similar distribution of word frequencies

(syntax + semantics are ignored)

[1] `https://aclweb.org/aclwiki/Distributional_Hypothesis`

Latent Semantic Analysis

- document = bag of words
- vector representation of documents
- compare by vector distance (angle)
- topic = set of words

See [Ioana, 2020] for detailed explanation.

LSA – step 1

- count document-term matrix (word frequency in documents)
- rows = term (words or multi-word expressions), columns = documents
- *sparse matrix*

term	D1	D2	D3	D4	D5	D6	D7	D8
abnormality	0	0	0	1	0	1	1	0
blood	0	1	1	2	1	0	1	1
culture	3	0	0	0	0	0	0	0
disease	0	2	3	0	1	1	0	0
rate	0	3	7	0	0	3	1	0

LSA – step 2

- weighting matrix elements
- most popular TF-IDF
  (Term Frequency × Inverse Document Frequency)
- term occurring in many documents is not interesting for analysis

word	D1	D2	D3	D4	D5	D6	D7	D8
abnormality	0	0	0	.6	0	.3	.5	0
blood	0	.1	.01	.4	.2	0	.2	.4
culture	.8	0	0	0	0	0	0	0
disease	0	.3	.1	0	.2	.03	0	0
rate	0	.8	.04	0	0	.2	.01	0

LSA – step 3

- Singular Value Decomposition (SVD), suitable decomposition for sparse data
  document-term matrix X ($m \times n$) is decomposed into the product of 3 matrices $X = U\Sigma V$, where
    - U – term-topic matrix $m \times m$
    - V – document-topic matrix $n \times n$
    - Σ – diagonal matrix

  U, V are unitary matrices ($AA^T = I$, I – identity matrix)

SVD $X = U\Sigma V$

$$X = \begin{pmatrix} 0 & 0 & 0 & .6 & 0 & .3 & .5 & 0 \\ 0 & .1 & .01 & .4 & .2 & 0 & .2 & .4 \\ .8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .3 & .1 & 0 & .2 & .03 & 0 & 0 \\ 0 & .8 & .04 & 0 & 0 & .2 & .01 & 0 \end{pmatrix}$$

$\leftarrow$ document-term matrix

$$U = \begin{pmatrix} 0.72 & 0.44 & 0. & -0.52 & 0.13 \\ 0.51 & 0.2 & -0. & 0.81 & -0.21 \\ 0. & -0. & 1. & 0. & -0. \\ 0.18 & -0.32 & -0. & 0.2 & 0.91 \\ 0.44 & -0.81 & -0. & -0.17 & -0.34 \end{pmatrix}$$

$\leftarrow$ term-topic matrix

$\swarrow$ document similarity matrix

$$V = \begin{pmatrix} 0. & 0.46 & 0.04 & 0.64 & 0.14 & 0.31 & 0.47 & 0.21 \\ -0. & -0.85 & -0.07 & 0.41 & -0.03 & -0.05 & 0.3 & 0.09 \\ 1. & -0. & -0. & 0. & -0. & 0. & 0. & -0. \\ 0. & 0.01 & 0.05 & 0.02 & 0.46 & -0.42 & -0.23 & 0.74 \\ -0. & -0.1 & 0.41 & -0.04 & 0.76 & -0.01 & 0.1 & -0.47 \\ -0. & -0.17 & -0.38 & -0.21 & 0.33 & 0.77 & -0.21 & 0.2 \\ 0. & 0.03 & -0.2 & -0.58 & 0.07 & -0.12 & 0.76 & 0.15 \\ 0. & -0.13 & 0.79 & -0.22 & -0.26 & 0.35 & 0.06 & 0.34 \end{pmatrix}$$

SVD $X = U\Sigma V$

$$\Sigma = \begin{pmatrix} 0.99 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0.85 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.8 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0.44 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0.18 & 0. & 0. & 0. \end{pmatrix}$$

LSA – step 4

dimensionality reduction: throw away rows and columns of the matrices[2]
$\sigma = (0.99, 0.85, 0.8, 0.44, 0.18)$
Keep first t singular values (and therefore first t columns from U + first t rows from V)
$t = 3$

$$U = \begin{pmatrix} 0.72 & 0.44 & 0. \\ 0.51 & 0.2 & -0. \\ 0. & -0. & 1. \\ 0.18 & -0.32 & -0. \\ 0.44 & -0.81 & -0. \end{pmatrix} \begin{array}{l} \text{abnormality} \\ \text{blood} \\ \text{culture} \\ \text{disease} \\ \text{rate} \end{array}$$

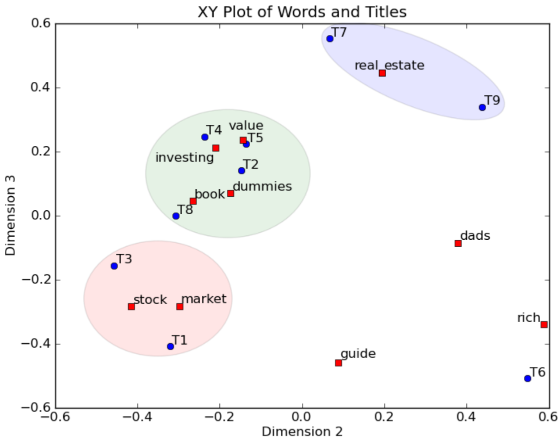
$$V = \begin{pmatrix} 0. & 0.46 & 0.04 & 0.64 & 0.14 & 0.31 & 0.47 & 0.21 \\ -0. & -0.85 & -0.07 & 0.41 & -0.03 & -0.05 & 0.3 & 0.09 \\ 1. & -0. & -0. & 0. & -0. & 0. & 0. & -0. \end{pmatrix}$$

(check absolute values)

[2]see Truncated SVD `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html`

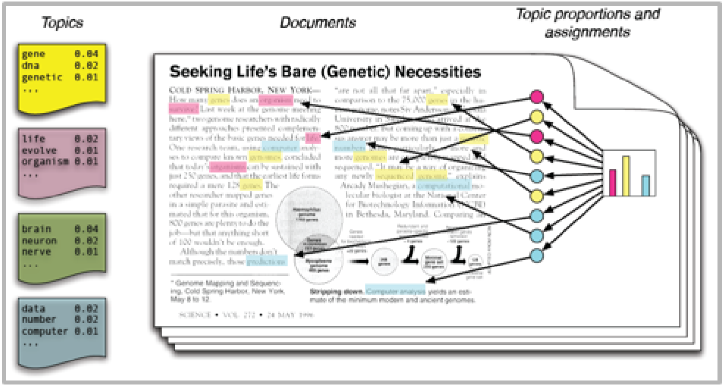
LSA – step 5

cluster close vectors (documents and terms)



Latent Dirichlet Allocation

- same assumptions as in LSA (distributional hypothesis + mixture of topics in one document)
- each document is a mix of topics
- LDA discovers topics and their ratio
- each word in document was generated by one of the topics



Example

Document 1: I like to eat broccoli and bananas.
Document 2: I ate a banana and spinach smoothie for breakfast.
Document 3: Chinchillas and kittens are cute.
Document 4: My sister adopted a kitten yesterday.
Document 5: Look at this cute hamster munching on a piece of broccoli.

Example

Topic A: 30% broccoli, 15% bananas, 10% breakfast, 10% munching
Topic B: 20% chinchillas, 20% kittens, 20% cute, 15% hamster

Example

Document 1 and 2: 100% Topic A
Document 3 and 4: 100% Topic B
Document 5: 53% Topic A, 47% Topic B

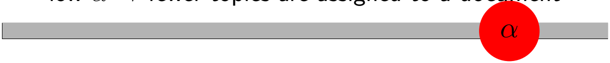
LDA process

- pick fixed number of topics K
- for each document $d \in D$, randomly assign topic to each word
- improve, for each document d:
  - for each word w and topic t:
  - assume all topic assignments are correct, except for current word
  - calculate $p(\textit{topic } t | \textit{document } d)$ – how many words in document have topic t?
  - calculate $p(\textit{word } w | \textit{topic } t)$ – how many assignments to topic t for word w?
  - new topic: probability $p(\textit{topic } t | \textit{document } d) \times p(\textit{word } w | \textit{topic } t)$
- repeat and reach almost steady state

LDA – generative probabilistic model

parametrized vectors of topics and documents
(α and β are concentration parameters)

low $\alpha \rightarrow$ fewer topics are assigned to a document



low $\beta \rightarrow$ fewer words model a topic

LDA Output

ψ – the distribution of words for each topic $k \in K$
ϕ – the distribution of topics for each document $d \in D$

Vector containing coverage of every topic for the document
$d_1 = [0.3, 0.4, 0.1, \dots]$

Topical characteristic of the corpus

LSA and LDA: Similarities and differences

- preprocessing: lowercase, punctuation removal, stopword removal, (stemming or lemmatization))
- both LDA and LSA ignore the syntactic structure
- the number of topics k is the input parameter
- LDA assumes arrangements of the words (n-grams)
- LDA assumes distribution of words in topics and distribution of topics in documents are Dirichlet distributions $\rightarrow$ topics might be more transparent
- output: wordcloud
- topic labels are difficult (and not part of LSA/LDA)

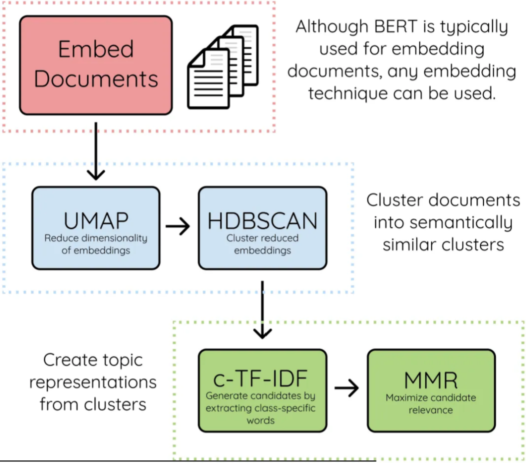
Weaknesses of LSA and LDA

predefined number of topics, wordlist (stopwords), stemming/lemmatization, ignore text structure

- Hierarchical Dirichlet Process (HDP) – unknown number of topics
- top2vec: word + document embeddings [Angelov, 2020] – captures the document semantics using word embeddings
- BERTopic – c-TF-IDF (class-based TF-IDF) + embeddings + document structure

BERTopic

- not a single algorithm
- parametrized: topics, hierarchical topics, semi-supervised (guided)



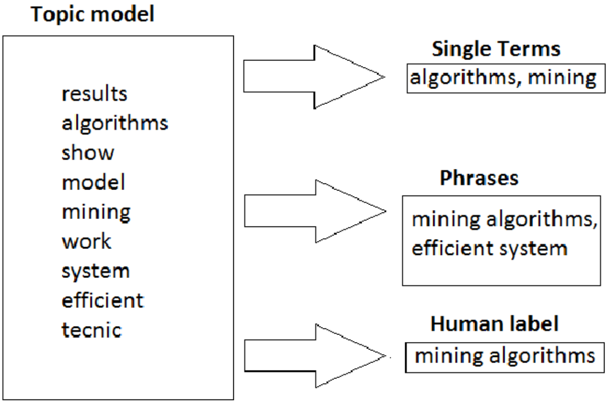
[3]

[3] https://medium.com/data-reply-it-datatech/bertopic-topic-modeling-as-you-have-never-seen-it-before-abb48bbab2b2

Topic Labeling

represent topic with human-friendly label from the label set of the topic

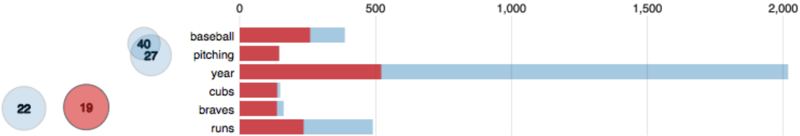
- find Wikipedia articles based on word list
- document summarization from topic documents



Topic Evaluation Methods

Good topics = interpretable topics
Evaluation methods comprise:

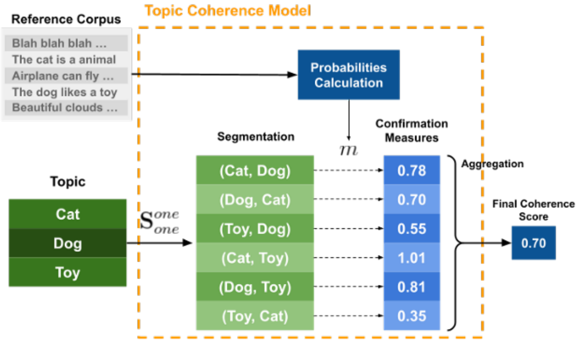
- eyeballing – `pyLDAVis`
- human judgement
- intrinsic methods – perplexity, coherence measures
- extrinsic methods – how does the resulting model influence subsequent task



Topic Coherence

measuring score for single topic quality by semantic similarity between words in topic [Röder et al., 2015]

- Segmentation – segment topic into pairs of word subset
- Probability Estimation – probability of words in documents
- Confirmation Measure – "how well" one subset support the other
- Aggregation – compute single score (e.g. by arithmetic mean)



Gensim – LSA

```
gensim.models.lsimodel.LsiModel(corpus=None,
num_topics=200, id2word=None, chunksize=20000, decay=1.0,
distributed=False, onepass=True, power_iters=2,
extra_samples=100)
```

- `chunksize` – number of documents in memory (more documents, more memory)
- `decay` – newly added documents are more important?
- `power_iters` – more iterations improve accuracy, but lower performance
- `onepass` – False to use multi-pass algorithm, for static data increase accuracy

Gensim – LDA

```
gensim.models.ldamodel.LdaModel(corpus=None,
num_topics=100, id2word=None, distributed=False,
chunksize=2000, passes=1, update_every=1,
alpha='symmetric', eta=None, decay=0.5, offset=1.0,
eval_every=10, iterations=50, gamma_threshold=0.001,
minimum_probability=0.01, random_state=None, ns_conf=None,
minimum_phi_value=0.01, per_word_topics=False)
```

- `chunksize` – number of documents in memory (more documents, more memory)
- `update_every` – number of chunks before moving to next step
- chunksize=100k, update_every=1 equals to chunksize=50k, update_every=2 (saves memory)
- `decay` – newly added documents are more important?
- `alpha, eta` – preset expected topics and word probability for start
- `eval_every` – log perplexity is estimated after x updates (lower number, slower training)

References I

Angelov, D. (2020).
Top2vec: Distributed representations of topics.

Blair, S. J., Bi, Y., and Mulvenna, M. D. (2020).
Aggregated topic models for increasing social media topic coherence.
Applied Intelligence, 50(1):138–156.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003).
Latent Dirichlet Allocation.
Journal of Machine Learning Research, 3:993 – 1022.

Castellanos, A., Cigarrn, J., and Garca-Serrano, A. (2017).
Formal concept analysis for topic detection.
Inf. Syst., 66(C):24–42.

References II

Curiskis, S. A., Drake, B., Osborn, T. R., and Kennedy, P. J. (2020).
An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit.
Information Processing & Management, 57(2):102034.

Grootendorst, M. (2022).
Bertopic: Neural topic modeling with a class-based tf-idf procedure.
arXiv preprint arXiv:2203.05794.

Ioana (2020).
Latent semantic analysis: intuition, math, implementation.
Available at `https://towardsdatascience.com/latent-semantic-analysis-intuition-math-implementation-a19`

References III

Lim, K. H., Karunasekera, S., and Harwood, A. (2017).
Clustop: A clustering-based topic modelling algorithm for twitter using word networks.
In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2009–2018. IEEE.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012).
Recommender systems.
Physics Reports, 519(1):1 – 49.
Recommender Systems.

Röder, M., Both, A., and Hinneburg, A. (2015).
Exploring the space of topic coherence measures.
In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.

References IV

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006).
Hierarchical Dirichlet processes .
Journal of the American Statistical Association, 101:1566 – 1581.

Wan, X. and Wang, T. (2016).
Automatic labeling of topic models using text summaries.
In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2297–2305.

Xie, P. and Xing, E. P. (2013).
Integrating document clustering and topic modeling.
CoRR, abs/1309.6874.