LSA, LDA, NMF feature extraction

Davide Buldrini

University of Bologna
Text mining project

10/10/2023

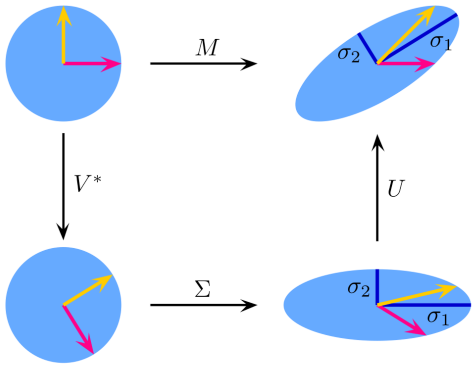
Methods

Focus on three methods for dimensionality reduction (feature extraction)

- Latent Semantic Analysis (LSA)
- Latent Dirichlet Allocation (LDA)
- Non-Negative Matrix Factorization (NMF)

LSA

LSA assumes that words that are close in meaning will occur in similar pieces of text (distributional hypothesis). A matrix containing word counts or (tf-idf values) per document is constructed than singular value decomposition (SVD) is used to reduce the number of rows while preserving similarity structure among columns. document can be compared by cosine similarity.



$$M = U \cdot \Sigma \cdot V^*$$

LSA

- PROS
  - LSA effectively reduces dimensionality
  - Fast
  - Ordered features, good also for 2D visualizations
  - Mathematically Sound
- CONS
  - Feature not interpretable

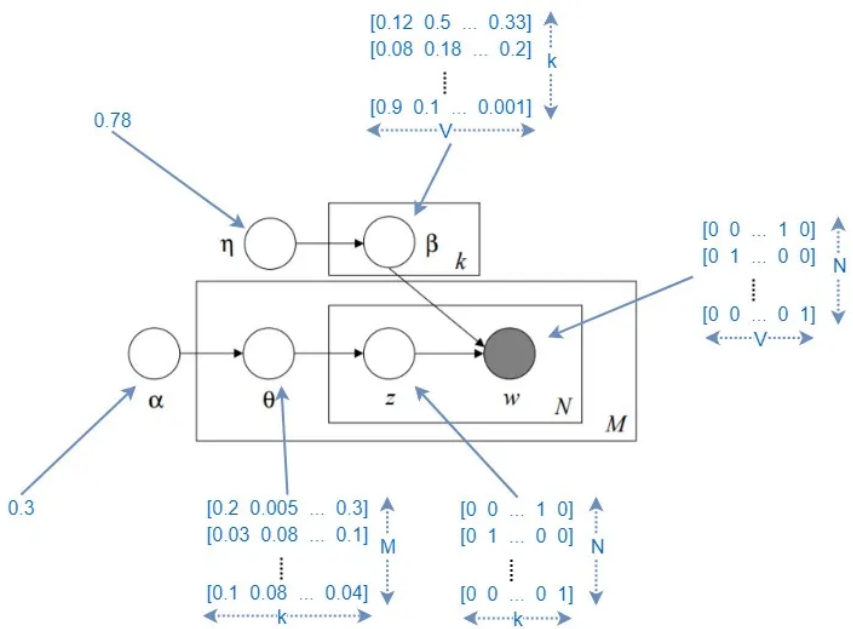
LDA

- LDA is a generative probabilistic model that represents documents as mixtures of topics
- A topic is considered to be a set of terms that, taken together, suggest a shared theme.
- LDA tries to uncover these latent (hidden) topics by analyzing the distribution of words in the documents, assigning a probability distribution of **topics to each document** and a probability distribution of **words to each topic**.
- These probability distributions can be used as features

LDA explained

M denotes the number of documents
N is number of words in a given document
α is the parameter of the Dirichlet prior on the per-document topic distributions
η is the parameter of the Dirichlet prior on the per-topic word distribution
θ_i is the topic distribution for document i
β_k is the word distribution for topic k
z_{ij} is the topic for the j-th word in document

w_{ij} is the specific word.



LDA explained

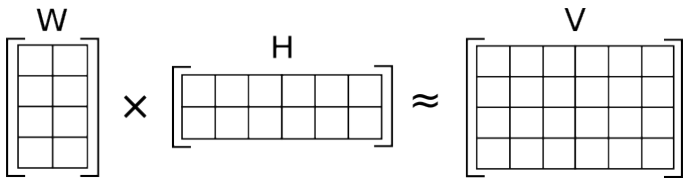
- Initially for each document each word in each document randomly assigned to one of k topics (k is chosen beforehand).
- For each document d, go through each word w and compute :
  - **p(topic t | document d)**:the proportion of words in document d that are assigned to topic t. Tries to capture how many words belong to the topic t for a given document d. Excluding the current word.
  - **p(word w| topic t)**: probability of observing a specific word (w) given a particular topic. If a word has high probability of being in a topic, all the documents having w will be more strongly associated with t as well
  - Update the probability for word w belonging to topic t, as: p(word w with topic t) = p(topic t | document d)x p(word w | topic t)

LDA

- PROS
  - Provides topic proportions for each document, allowing you to understand the mixture of topics in a document.
  - Multiclass classification as stated above (note that topics can be different form classes)
  - Interpretability : Probability distribution and topics
  - Polysemy: a single word can be important for more than one topic
- CONS
  - Parameter tuning
  - Slower than other methods
  - not easily adaptable for visualization purpose

NMF

In Non Negative matrix Factorization a matrix V is factorized in the product of two matrix W H (i.e. $\mathbf{V = WH}$) none of the matrix has negative elements this is not guaranteed to be possible and usually achieved by numerical approximation. The dimensionality can be significantly reduced if V is an m x n matrix, W is an m x p matrix, and H is a p x n and p can be small the product HW should be similar to V minimizing Frobenius norm and enforcing the fact that the values have to be non negative.



NMF

- PROS
  - Effectively reduces dimensionality
  - Usually Fast Positive values make it more interpretable
- CONS
  - Not suited for visualization

Data

All the analysis will be performed in a subset of the fetch 20newsgroups 12487 text documents divided in 13 classes referring to 5 bigger classes (computer, motors, sports, science, politics). initially we will extract 13 features for each method

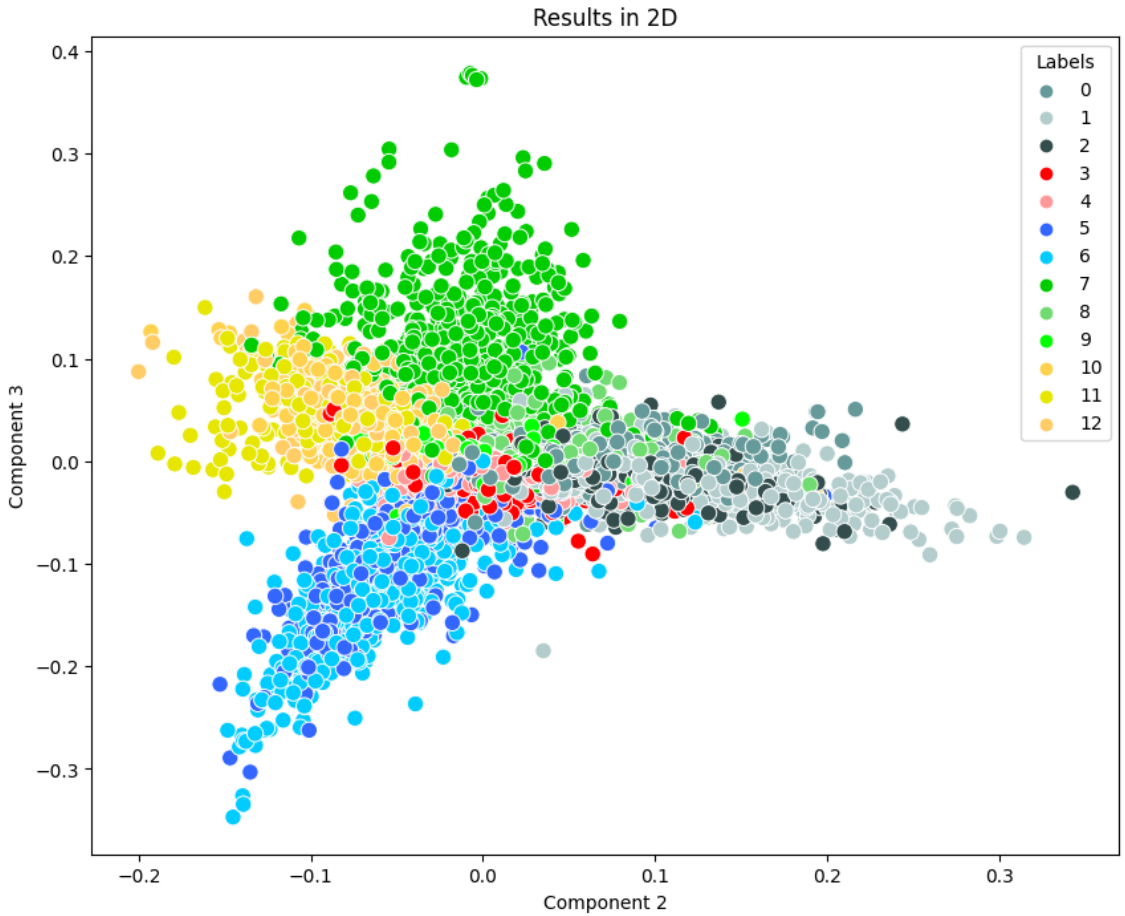
Visualization

The importance of feature is ordered for LSA (due to the Singular Value Decomposition the first few dimensions corresponding to bigger singular values) so it is possible to have meaningful 2D scatter plot, while for the other methods an histogram for each class to see different distribution of topics (averaged).

LSA visualization

The first feature seems not useful for the visualization, the value can be too big w.r.t. others.

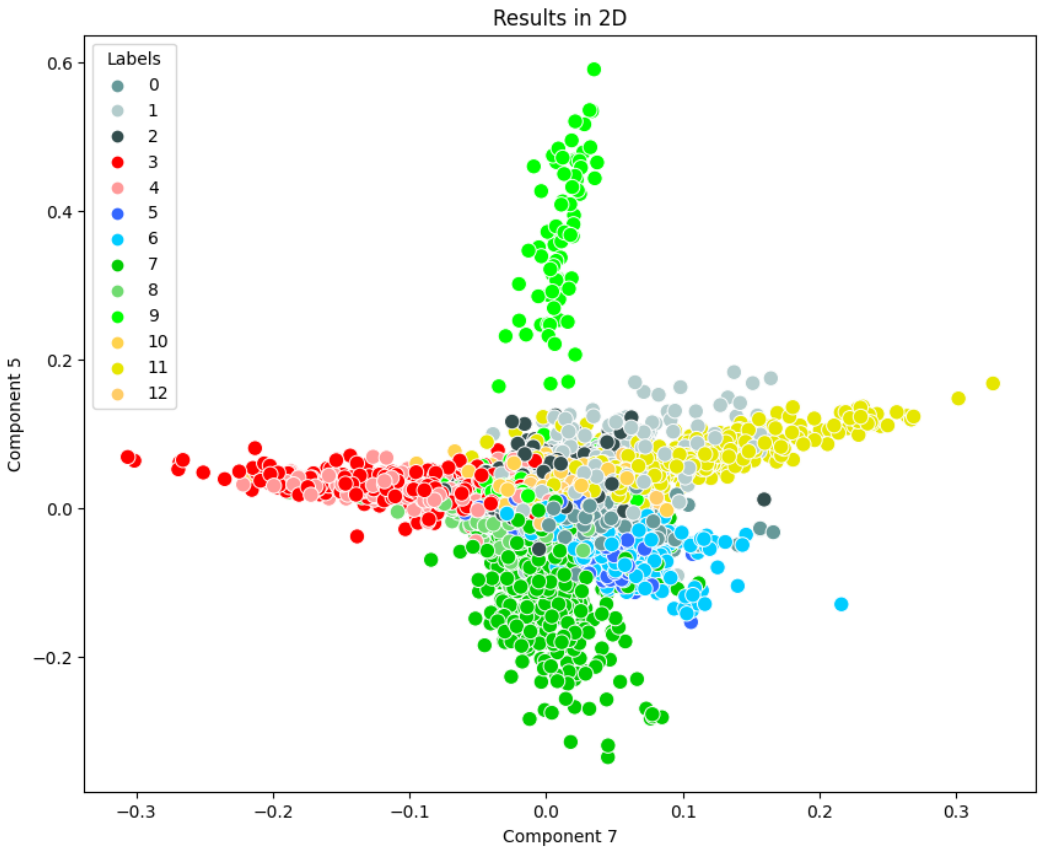
gray = computer
red= vehicles
blue = sport
green = science
yellow = politics



LSA visualization

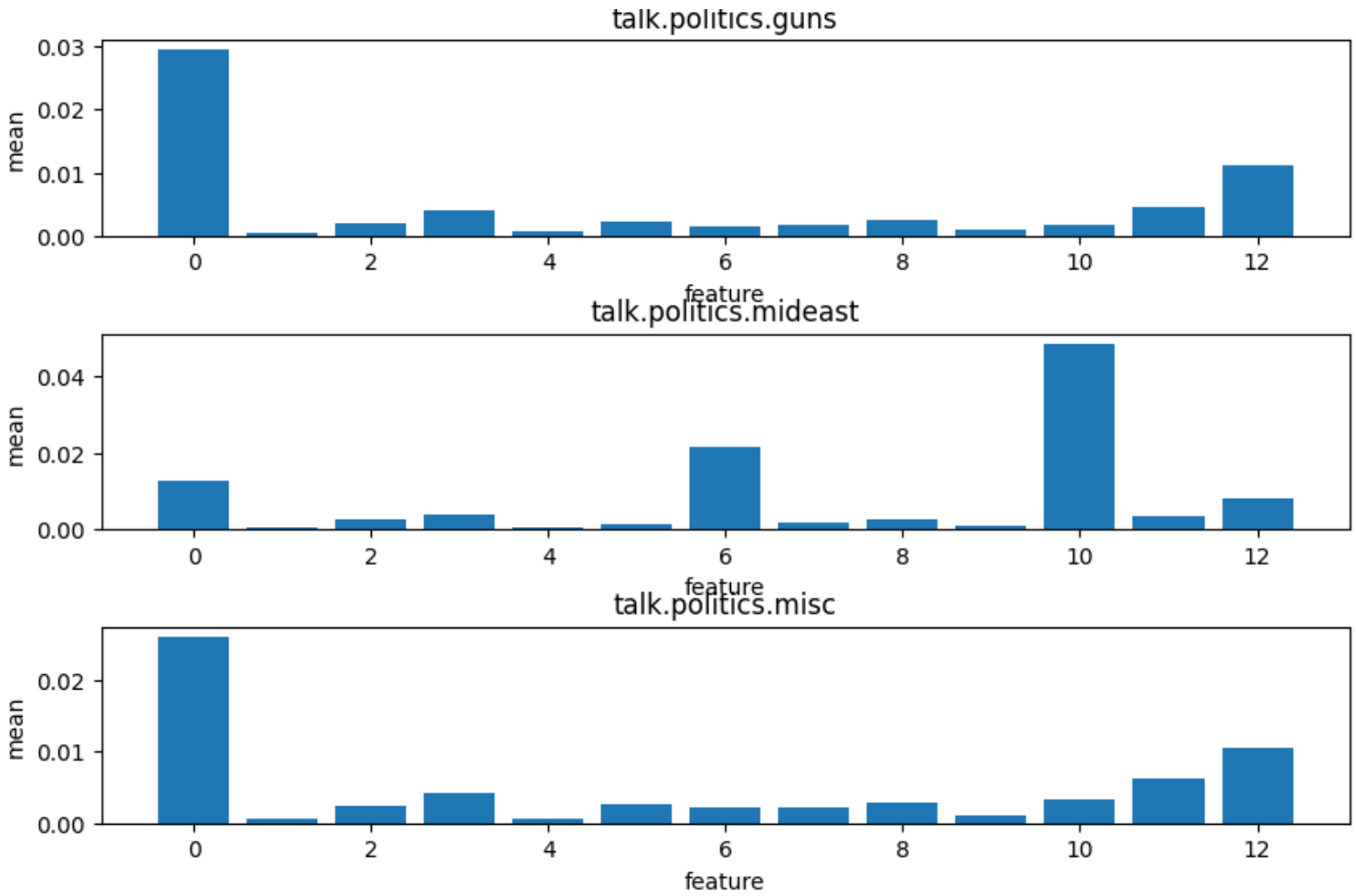
The remaining ones can be highlighted using feature 5 and 7

gray = computer
red= vehicles
blue = sport
green = science
yellow = politics



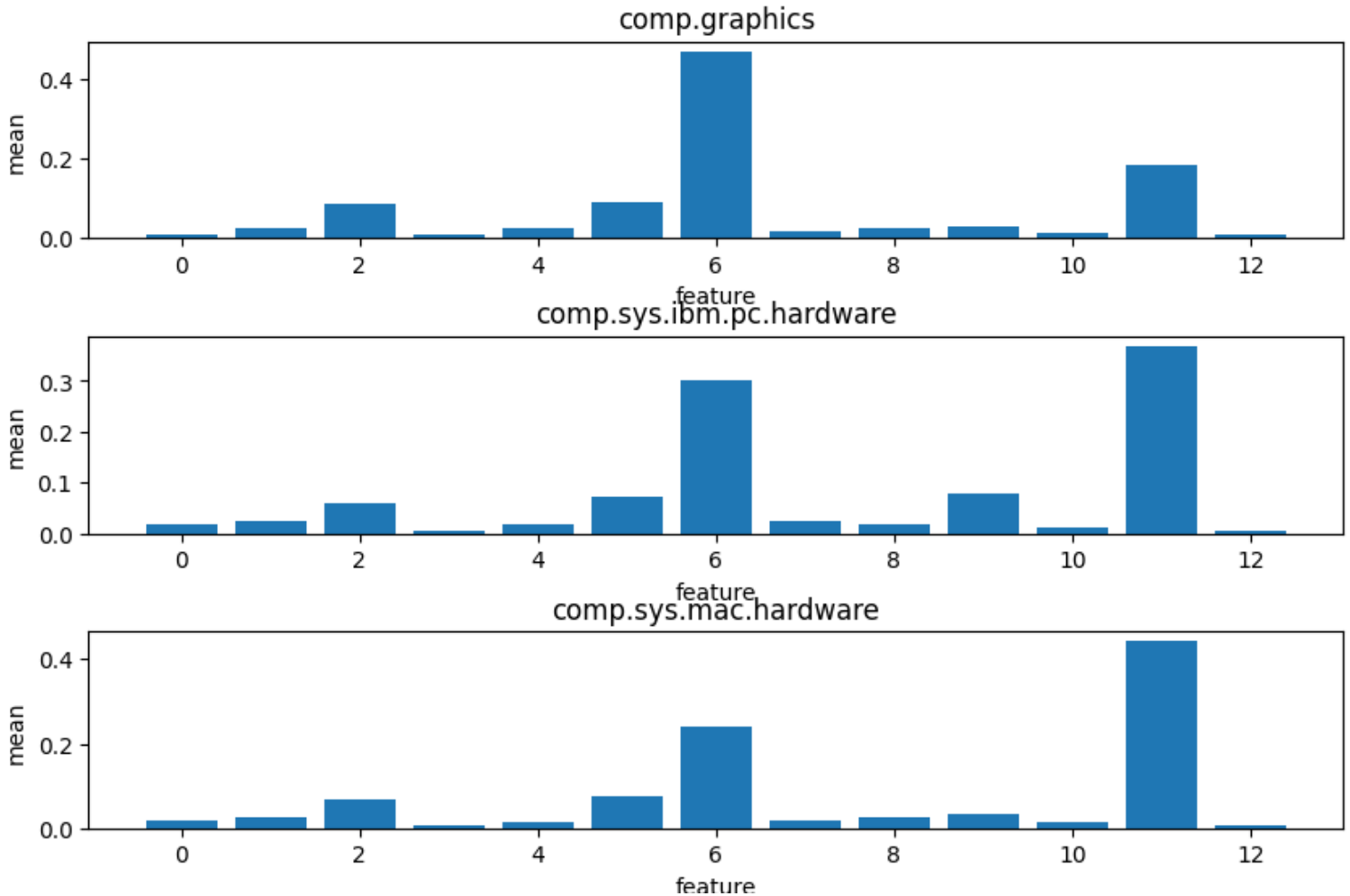
NMF visualization

in NMF feature 10 seems to be important only for politics mideast



LDA visualization

in LDA feature 6 seems important only computer related news.



Explainability

To understand a topic in NMF and LSA is enough to plot the most important words for that topic. Negative values make harder to interpret LSA features.

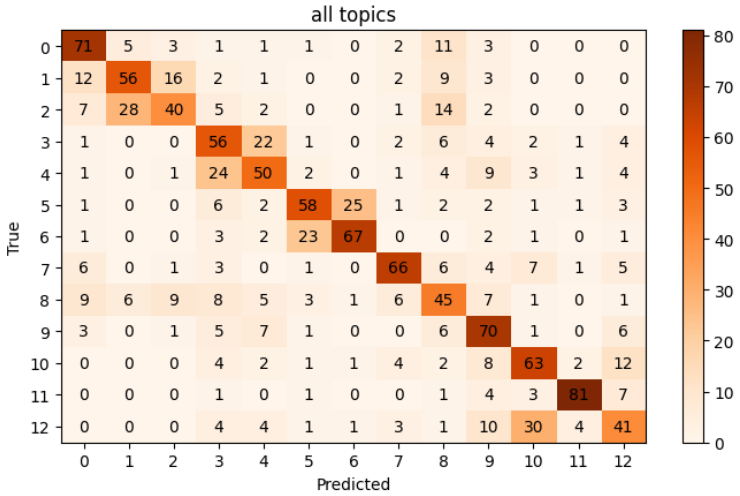
Explainability

- In NMF the topic 2 is important for sports, while the 10 for mideast
  - topic 2 : ['game', 'team', 'games', 'year', 'hockey', 'baseball', 'last', 'season', 'players', 'espn']
  - topic 10 : ['israel', 'jews', 'israeli', 'arab', 'jewish', 'arabs', 'palestinian', 'palestinians', 'peace', 'adam']
- In LDA topic 6 is important for computer
  - topic 6 : ['image', 'file', 'jpeg', 'software', 'graphics', 'use', 'data', 'program', 'pc', 'available']

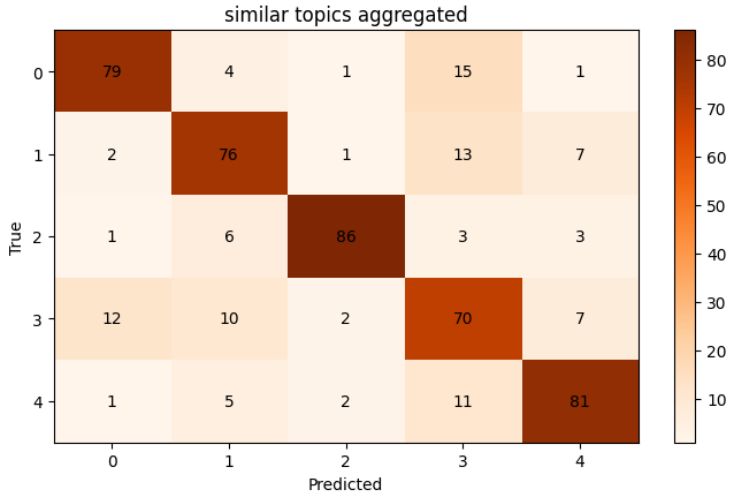
The feature extracted are used to fit a Random Forest and to predict the 13 classes than the prediction are also aggregated in the 5 bigger classes.

LSA

F1 13 classes 0.59

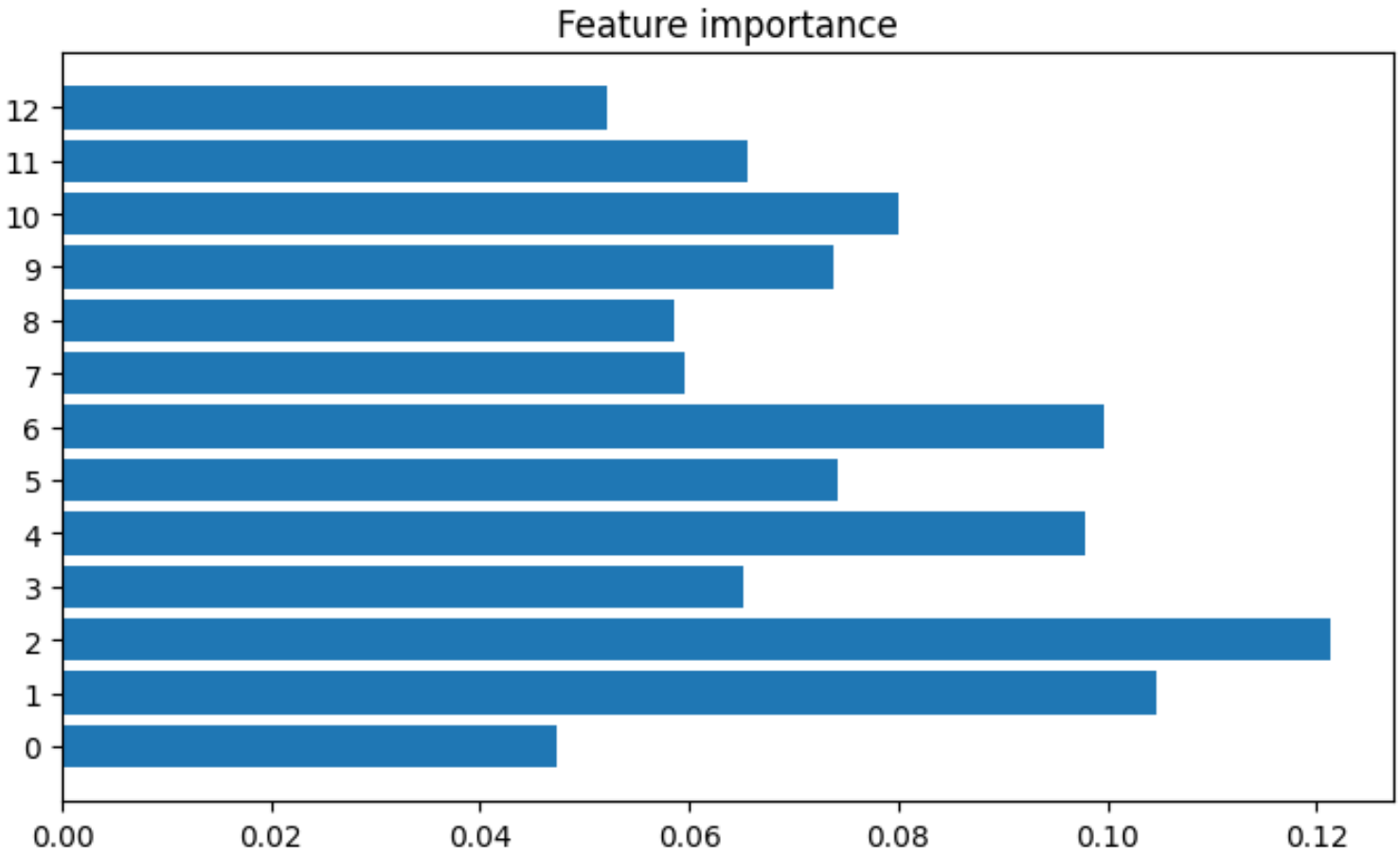


F1 5 classes 0.79



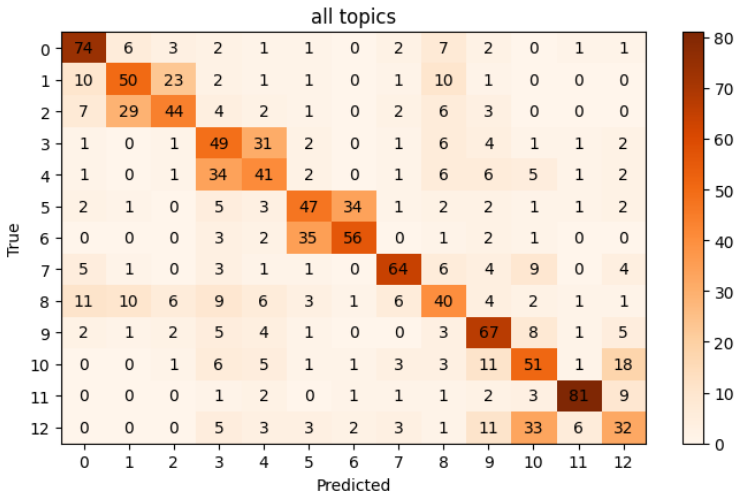
LSA Feature importance

As anticipated the first feature is not so important but then the features are ordered by importance

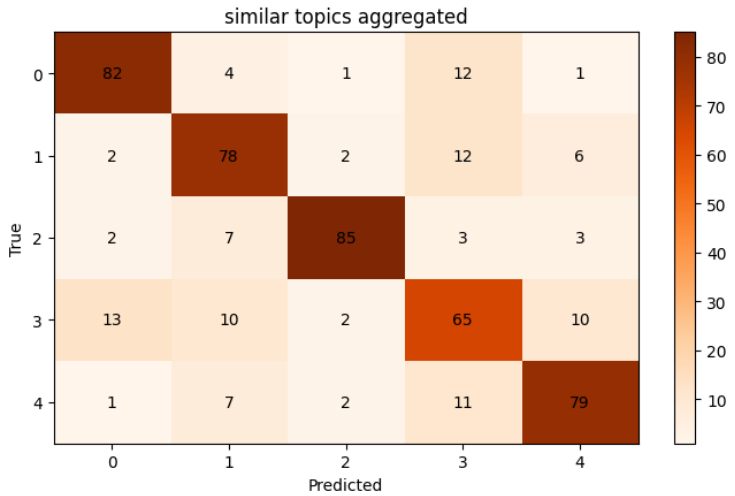


NMF

F1 13 classes 0.54

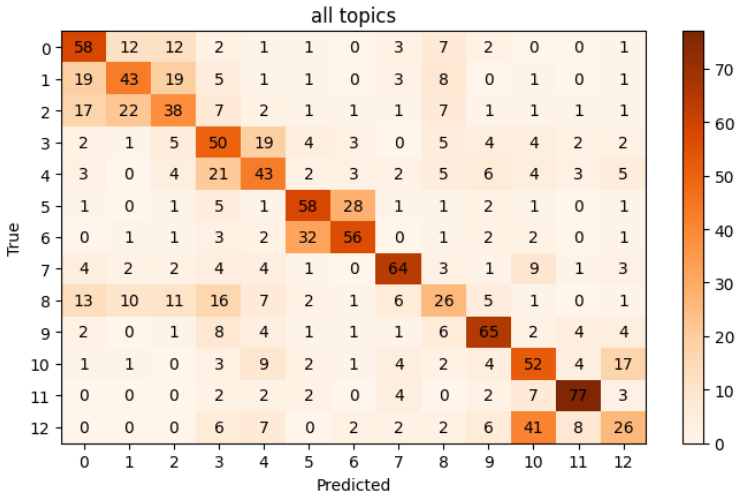


F1 5 classes 0.78

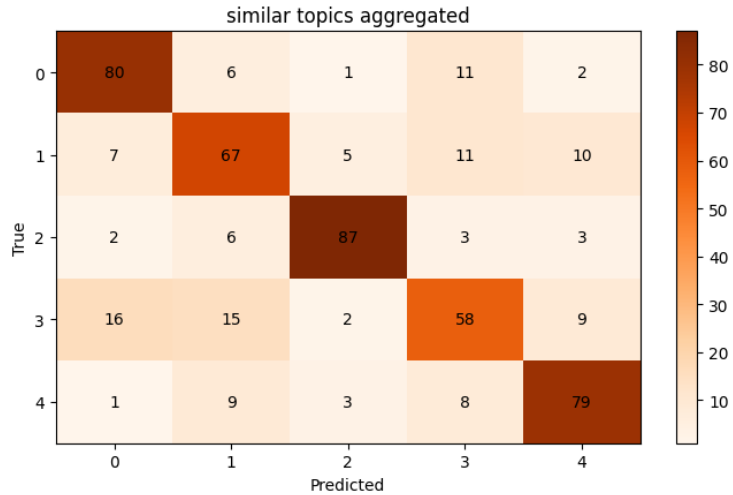


LDA

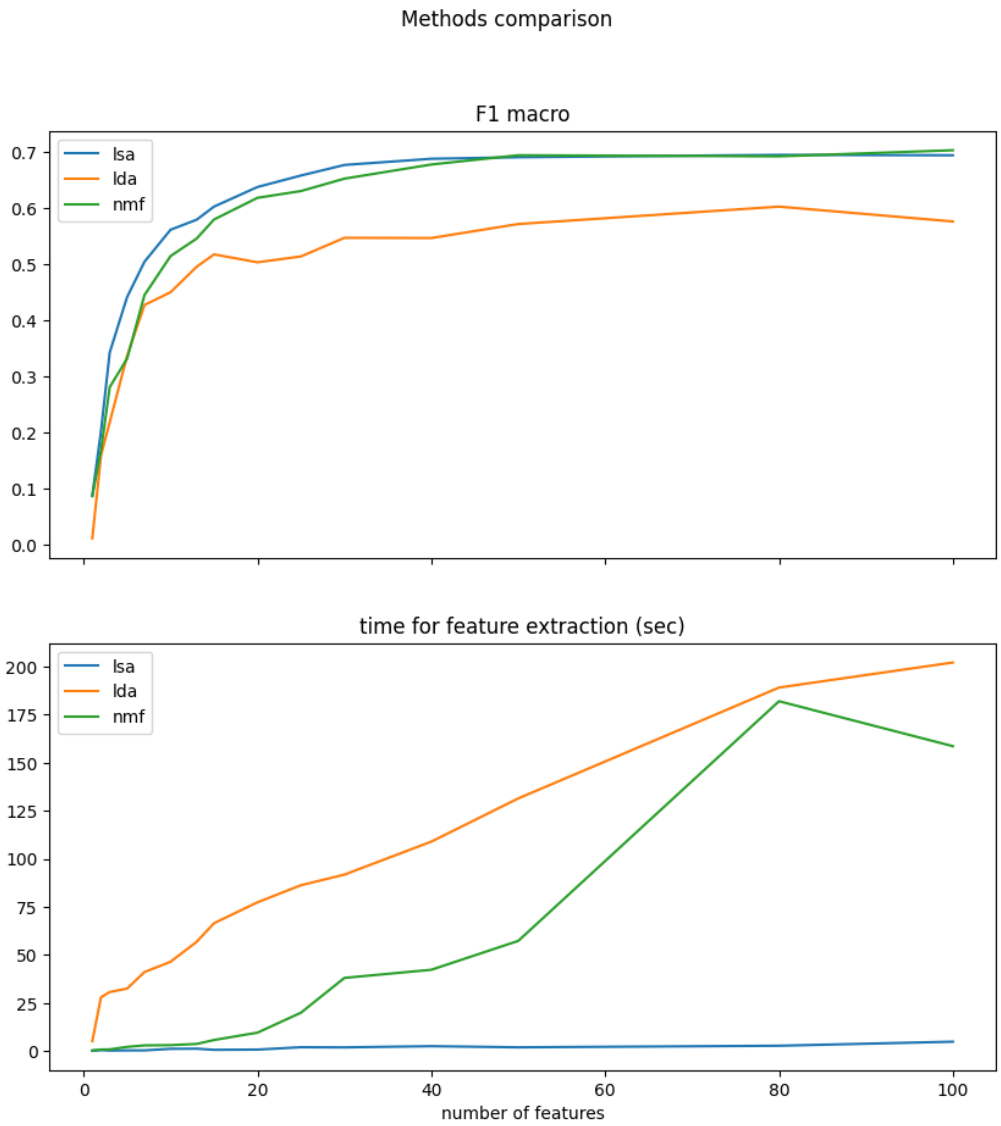
F1 13 classes 0.50



F1 5 classes 0.74



Comparison different features number



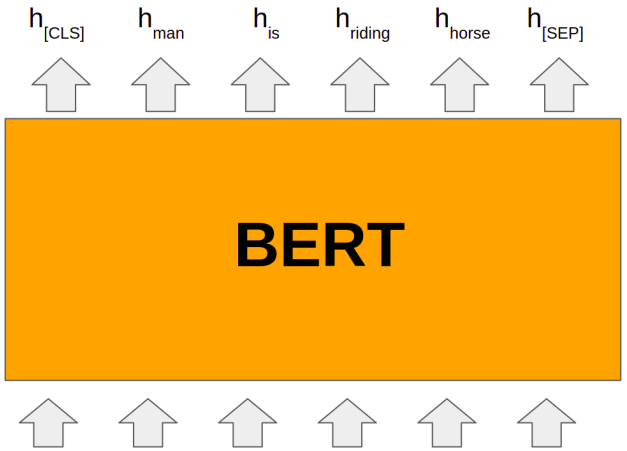
Pretrained language models

Documents representation can be created or extracted from Language model and can be used for classification or other downstream tasks. In this section two ideas are tested:

- distil-BERT last Hidden state as representation of document.
- Universal Sentence Encoder (USE) from Google.

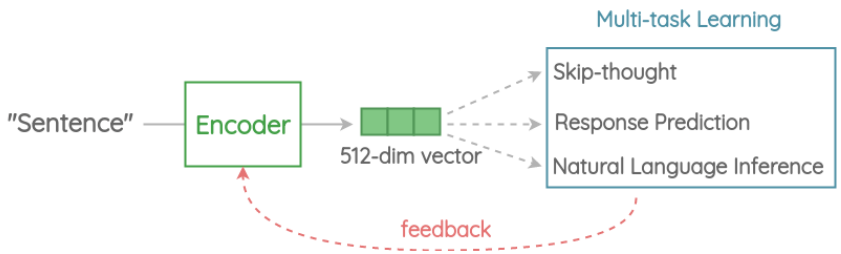
distil-BERT

For the classification is used the output of the last hidden layer, it contains the [CLS] followed by the token embedding. The [CLS] should contain a rich representation of the entire input sentence, but without the fine-tuning is not given that the representation will be meaningful for our task.
The [CLS] and token embedding, are feed to the classifier, to reduce size mean and argmax are used. Unfortunately without any fine-tuning this is not significantly better than a Dummy classifier that always select the most common label



USE

The Universal Sentence Encoder encodes text into high-dimensional vectors that can be used for text classification, semantic similarity, clustering and other natural language tasks.



It reaches F1 macro 0.70 (13 classes) and F1 0.83 (5 classes) that is still comparable with LSA and NMF with 100 features

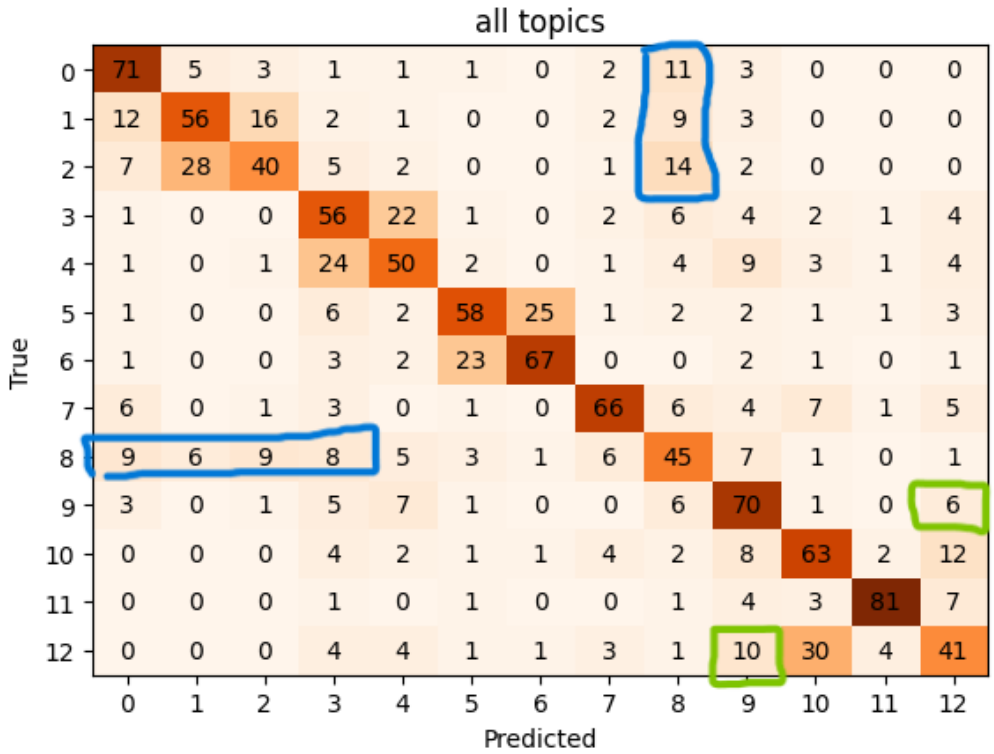
Lbl2Vec

Lbl2Vec is an algorithm for unsupervised document classification and unsupervised document retrieval. It automatically generates jointly embedded label, document and word vectors and returns documents of topics modeled by manually predefined keywords. The plain Lbl2Vec model uses Doc2Vec.

- Keywords are manually defined for each class
- Create jointly embedded document and word vectors using Doc2Vec (or sentence transformers)
- Find document vectors that are similar to the keyword vectors of each topic.
- Clean outlier document vectors for each topic
- Compute centroids
- Compute label vector <-> document vector similarities for each label vector and document vector in the dataset.

Error Analysis

Most of the confusion matrix are similar to the following one.



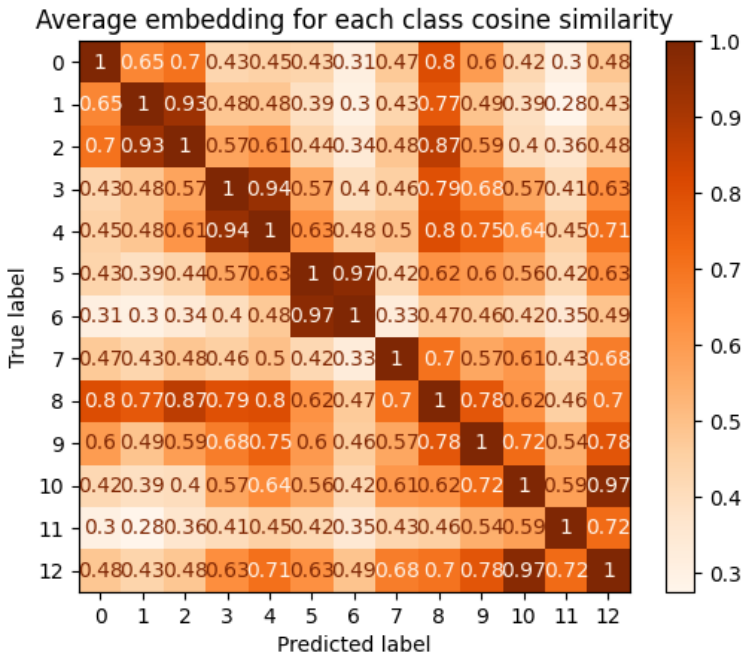
Most of the error are next to the diagonal so they follow under the same aggregated classes. and this error disappear when we pass to the 5 bigger classes problem.

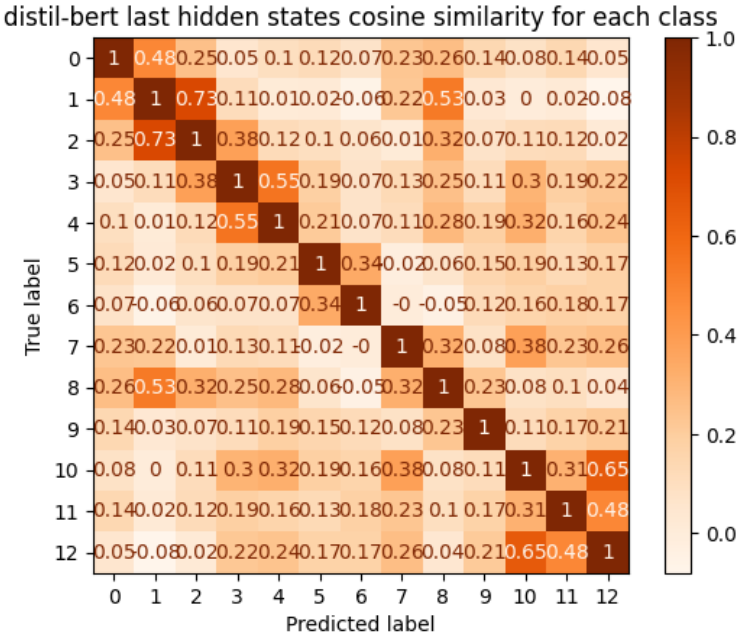
Error Analysis

- As we can see class 8 sci.electronics is often miss-classified as one of the first three that are computer related classes. and sometimes also with the fourth related to cars. (those topic can have lot
- class 9 sci.med is often swapped with talk.politics.misc this can contain medical informations

Embedding analysis

For every class is possible to compute the average of all document embedding related to that class. Then the average embedding can be compared using cosine similarity and obtaining the following matrices. The first is for LSA embedding the latter is from distil-BERT embedding.





Embedding analysis

The similarities in average embedding reflect errors seen in the previous section (i.e. class 8 and 1)
The difference are more clear in the BERT emebedding so we can expect better classification score form that model.

Word Embedding analysis

Computing cosine similarities between word embedding and the average class embedding can be useful to understand the word that are most important for the classification. We expect that word with bigger norm are more important so the similarity can be weighted using the norm. USE embeddings have norm close to 1 and Bert embedding has high dimensionality so the results on them aren't as good as tho one of simpler models.

LSA Most important words

Most important word for LSA for each class

comp.graphics : ['graphics', 'image', 'program', 'file', 'ftp', 'files', 'software', 'gif', 'format', 'available']
comp.sys.ibm.pc.hardware : ['scsi', 'controller', 'ide', 'card', 'drive', 'bus', 'pc', 'port', 'drives', 'mac']
comp.sys.mac.hardware : ['mac', 'card', 'monitor', 'apple', 'board', 'video', 'work', 'ram', 'memory', 'port']
rec.autos : ['car', 'bike', 'cars', 'engine', 'miles', 'dealer', 'oil', 'ford', 'honda', 'rear']
rec.motorcycles : ['bike', 'car', 'ride', 'cars', 'engine', 'riding', 'bikes', 'miles', 'front', 'rear']
rec.sport.baseball : ['game', 'team', 'games', 'year', 'season', 'hockey', 'players', 'last', 'win', 'play']
rec.sport.hockey : ['game', 'games', 'team', 'hockey', 'season', 'play', 'year', 'win', 'last', 'baseball']
sci.crypt : ['key', 'encryption', 'clipper', 'chip', 'keys', 'escrow', 'algorithm', 'nsa', 'secure', 'security']
sci.electronics : ['looking', 'computer', 'used', 'help', 'software', 'etc', 'modem', 'mac', 'car', 'appreciated']
sci.med : ['geb', 'dsl', 'n3jxp', 'skepticism', 'cadre', 'chastity', 'intellect', 'shameful', 'surrender', 'gordon']
talk.politics.guns : ['people', 'gun', 'batf', 'fbi', 'guns', 'koresh', 'fire', 'weapons', 'crime', 'police']
talk.politics.mideast : ['israel', 'jews', 'jewish', 'war', 'israeli', 'arab', 'killed', 'arabs', 'soldiers', 'world']
talk.politics.misc : ['people', 'gun', 'guns', 'batf', 'fbi', 'children', 'government', 'weapons', 'crime', 'us']

Distil-BERT most important word

While those word are quite interpretable for LSA, LDA and NMF their are not the same for language models as is possible to see for distil-BERT

comp.graphics : ['hospitalized', 'puck', 'italians', 'v1', 'journalism', 'jitter', 'jaw', 'khan', 'pseudonymity', 'v8s']
comp.sys.ibm.pc.hardware : ['06066', 'ouch', 'anarchists', 'honorable', 'stubble',
'igor$_I$', *arionov*', *indecent*', *lenarduzzi*', *x*9', *rapacity*']
comp.sys.mac.hardware : ['leaking', 'wimsey', '06066', 'talons', 'trades', 'anarchists', 'tan', 'fishes', 'pkp', 'ouch']
rec.autos : ['7140', '664', 'bayur', '6674', 'mif', 'elitist', 'subset', 'pcx', 'revolutionaries', 'sequential']
rec.motorcycles : ['5500', 'application', 'soldier', 'oto', 'marine', 'premiere', 'tamara', 'mfm', 'guard', 'mcats']
rec.sport.baseball : ['1995', 'neighbours', '91109', '4440', '45', '916', 'fatalities', 'tenth', 'senses', '1kspt']
rec.sport.hockey : ['garaged', 'myopia', 'bli', 'strategies', 'setver', 'msg', 'frost', 'contries', 'commissioner', 'fp1']
sci.crypt : ['foreign', '32w', 'complies', 'diplomats', '773', 'poisonous', '8163', 'seed', '756', 'poised']
sci.electronics : ['johnt', 'humour', 'stability', 'despair', 'warthawks', 'blvd', 'prized', 'purple', 'lori', 'wes']
sci.med : ['gill', 'wordprocessor', 'asset', '42bis', 'unreal', 'pre', 'improvement', 'occurance', '4189', 'transplanted']
talk.politics.guns : ['tomd', 'rejoinder', 'lot', 'royce', 'dystrophy', 'daring', 'speedy', 'rightly', 'demeanor', 'dts']
talk.politics.mideast : ['credited', 'harper', 'hepatotoxicity', 'concealment', 'turk', 'defenders', 'microscopic', 'archo',
'cphv', 'r'_i]

talk.politics.misc : ['powerup', 'categorically', 'yaqouv', 'cauz', 'craving', 'expressed', 'hfd', 'restrictions', 'canard',

'ovens']

Summary of the performances

	macro F1		n features
	13 classes	5 classes	
LSA	0.59	0.80	13
	0.66	0.80	25
	0.69	0.81	100
LDA	0.50	0.74	13
NMF	0.54	0.78	13
	0.71	0.82	100
USE	0.71	0.83	512
Lbl2Vec	0.54	0.76	
BERT fine-tuned	0.91	0.95	

Conclusions

- The methods based on matrix factorization and probability estimation (LSA, LDA, NMF) have a quality of features/embedding similar to the one obtained trough a large pretrained Neural network like USE.
- LSA, LDA, NMF are more interpretable than LM based ones.
- The mentioned method reach this quality of features without using the labels.
- distil-BERT for classification after fine-tuning achieves way better performances, but using its last Hidden state as sentence representation it is not useful (it became useful after fine tuning but it is not unsupervised).
- Features once extracted can be used for clustering (like Lbl2Vec)

The End