Text Mining & Search

Brambatti Eleonora 858098
Fracchia Camilla 898235
Privitera Marta 898017

Università degli Studi di Milano Bicocca
Anno Accademico 2023/2024

Task performed

Text Classification

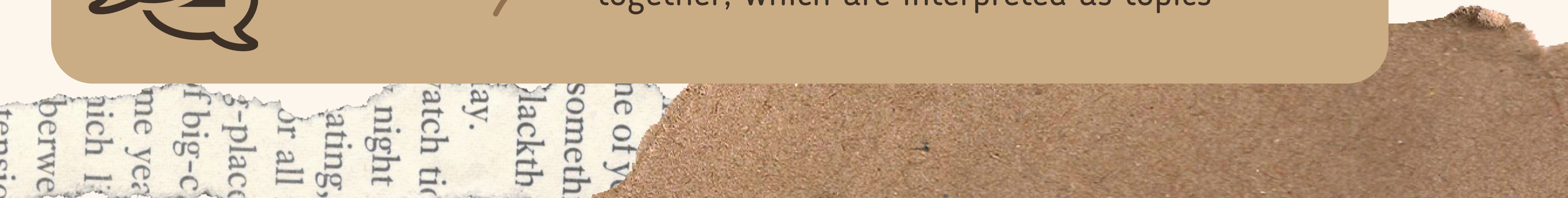
predicting to which, among a predefined finite set of classes a data item belongs to

Binary classification

By sentiment

Topic modeling

provides collections of words that make sense together, which are interpreted as topics



Data Preparation

Dataset

Goodreads website

Reviews about different genres

- Crime
- Thriller
- Mystery

01

HANDLING MISSING INFORMATION

- Missing values removal
- Empty rows removal

02

HANDLING CORPUS DIMENSION

- Stratified sampling to reduce corpus dimension
- Representativeness of each class is guaranteed

Data Preparation

Dataset

Goodreads website

Reviews about different genres

- Crime
- Thriller
- Mystery

03

HANDLING DEFAULT CLASS

- Inconsistency between zero class and semantic meaning of reviews

```
Just amazing.

I just want to be stephanie plum.
```

04

ENGLISH REVIEWS

- ‘Langid‘ library

05

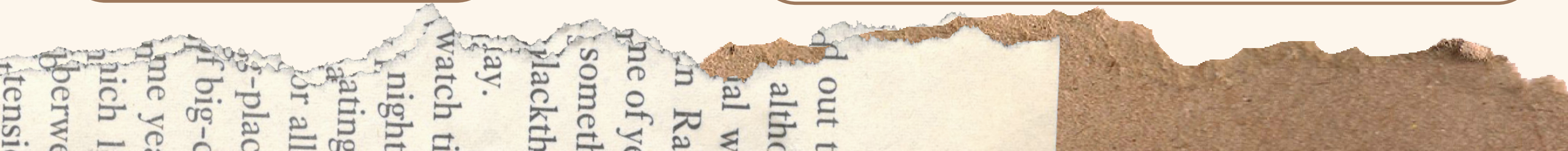
MAPPING LABELS

- Classes 1-2-3: ‘negative’
- Classes 4-5: ‘positive’

Pre Processing

General pre-processing to expedite algorithms applications and to have suited/proper dataset

1. Regular expressions	Numbers and new empty strings remotion
2. Normalization	Lower casing and links remotion
3. Stop Words removal	Elimination of stop words exception for negation
4. Regular expressions	Punctuation and white spaces remotion



Text classification Pre Processing

1. Label Encoding

‘Positive’ mapped to 1
‘Negative’ mapped to 0

2. Shuffling Dataset

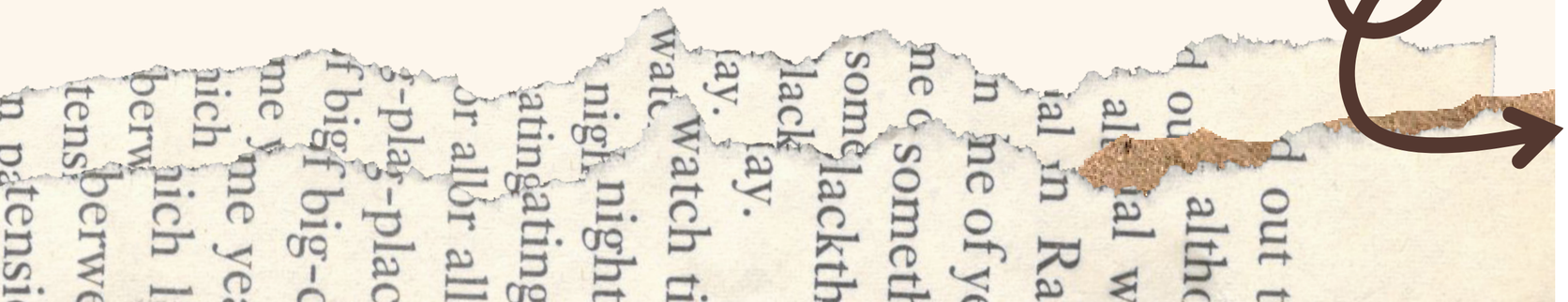
To prevent biases
Both classes in train and test set are shuffled

4. Word count Analysis

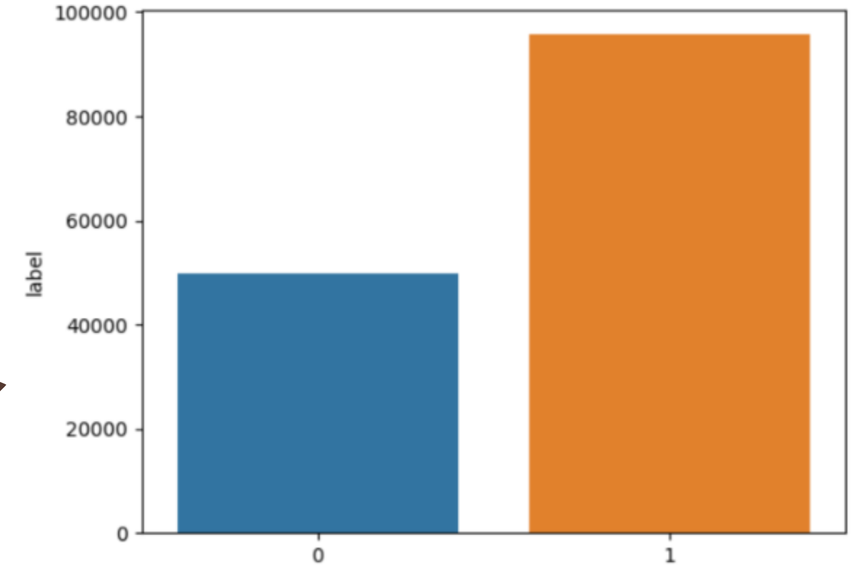
Keeping only 250 words reviews

3. Balancing Dataset

Undersampling technique



```
1     95706
0     49945
Name: label, dtype: int64
<Axes: ylabel='label'>
```



Document Representation

TF-IDF

Importance of a word in a document relative to a corpus

Tf - idf Vectorizer

max_features: limit to 5000 most frequent words

MODELS TRAINED

- Random Forest
- Logistic Regression
- XGboost
- Decision Tree

Models Evaluation

Random Forest Accuracy: 0.732
Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.73	0.75	0.74	10035
1	0.74	0.72	0.73	9943
accuracy			0.73	19978
macro avg	0.73	0.73	0.73	19978
weighted avg	0.73	0.73	0.73	19978

XGBoost Accuracy: 0.733
XGBoost Classification Report:

	precision	recall	f1-score	support
0	0.72	0.76	0.74	10035
1	0.75	0.70	0.72	9943
accuracy			0.73	19978
macro avg	0.73	0.73	0.73	19978
weighted avg	0.73	0.73	0.73	19978

Logistic Regression Accuracy: 0.755
Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.76	0.75	0.75	10035
1	0.75	0.76	0.76	9943
accuracy			0.76	19978
macro avg	0.76	0.76	0.76	19978
weighted avg	0.76	0.76	0.76	19978

Decision Tree Accuracy: 0.628
Decision Tree Classification Report:

	precision	recall	f1-score	support
0	0.63	0.63	0.63	10035
1	0.63	0.62	0.63	9943
accuracy			0.63	19978
macro avg	0.63	0.63	0.63	19978
weighted avg	0.63	0.63	0.63	19978

Word Embedding I

WORD2VEC

- Tokenization
- Window size: 5
- Minimum word count: 5
- Number of workers: 4
- Average pooling

MODELS TRAINED

- Logistic Regression
- KNN
- Decision Tree
- Random Forest

Models Evaluation

Model	Accuracy	Precision	Recall	F1 Score
① Logistic Regression	0.744	0.743	0.743	0.743
Random Forest Classifier	0.726	0.730	0.712	0.721
KNN Classifier	0.665	0.655	0.689	0.672
Decision Tree Classifier	0.613	0.611	0.613	0.612

Word Embedding 2

GLOVE

- Keras Tokenizer
- Padding
- Mapping word vector to 100 dimensions vector
- Embedding matrix

MODELS TRAINED

- Basic Deep Learning Models 1 & 2
- LSTM
- CNN
- LSTM + CNN

Baseline Model

25 epochs

01

EMBEDDING LAYER

- Initialized with embedding matrix
- Non Trainable

02

LSTM LAYER

- 100 units
- 10% drop out

03

DENSE LAYER

- Sigmoid as activation
- 10% drop out

Model Performance

METRICS

78% Accuracy

0.2 Loss value

Baseline Model 2

50 epochs, halted at 25

01

EMBEDDING LAYER

- Initialized with embedding matrix
- Non Trainable

02

LSTM LAYER

- 128 units
- 10% drop out

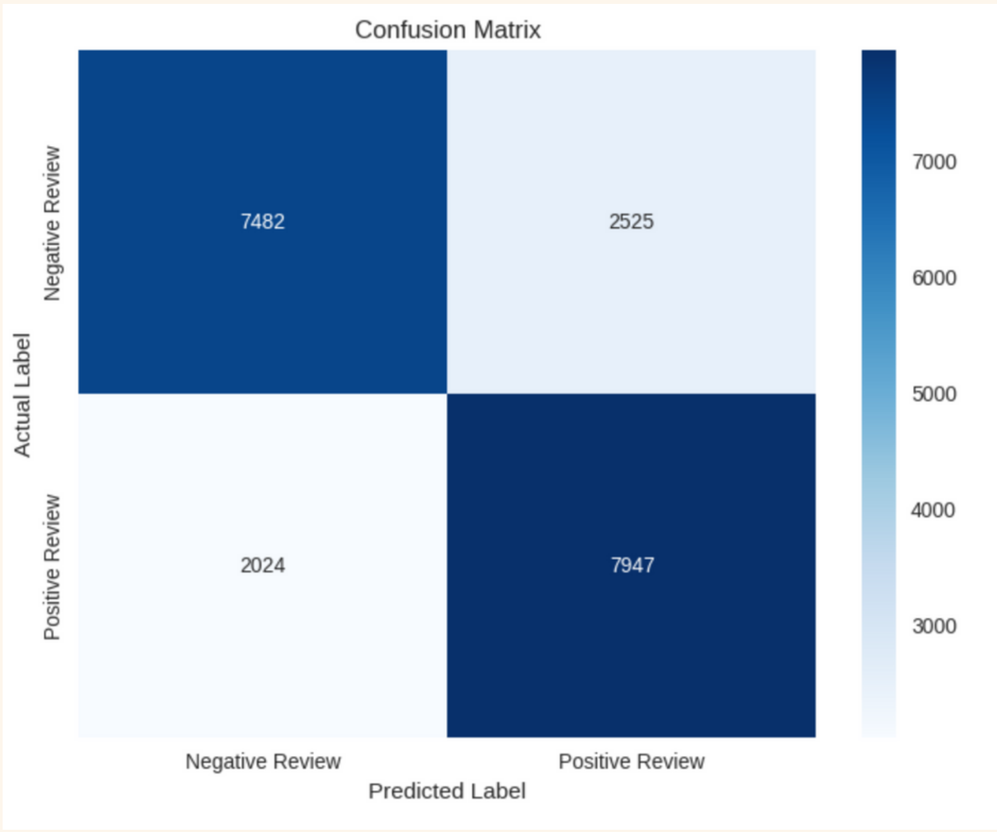
03

DENSE LAYER

- Sigmoid as activation
- 10% drop out

Model Performance

Improved with respect to model 1



Higher recall

LSTM

35 epochs

01

EMBEDDING LAYER

- Initialized with embedding matrix
- Non Trainable

02

3 LSTM LAYERS

- 256 units

03

DROPOUT LAYERS

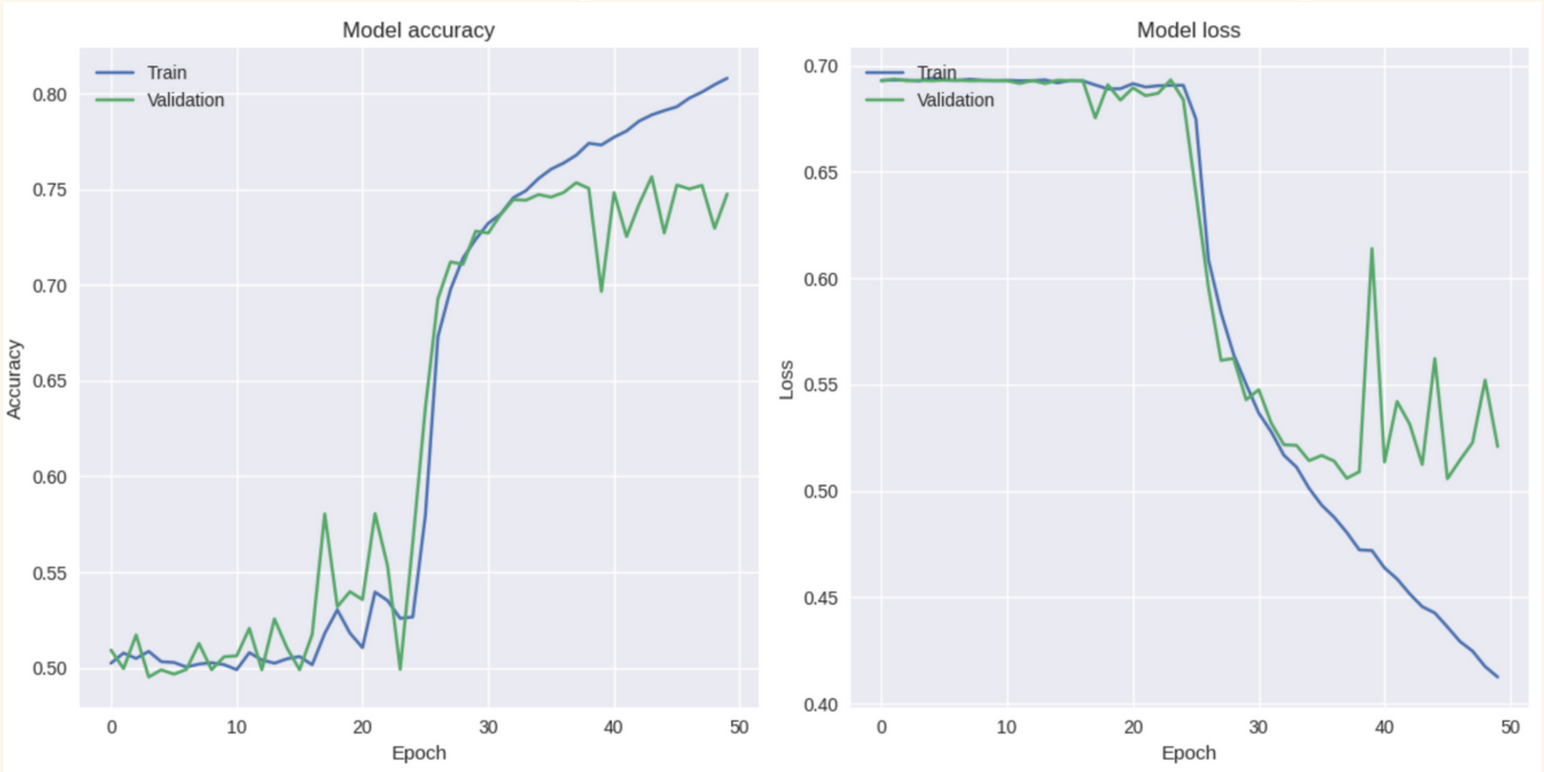
- 20% drop out

04

DENSE LAYER

- Sigmoid as activation

Model Performance



FLUCTUATING PATTERN

Both on validation accuracy and loss

LSTM wasn't the most robust architecture, so we tried CNN

	precision	recall	f1-score	support
0	0.76	0.72	0.74	10007
1	0.74	0.77	0.75	9971
accuracy			0.75	19978
macro avg	0.75	0.75	0.75	19978
weighted avg	0.75	0.75	0.75	19978

cnn

20 epochs + early stopping

01

EMBEDDING LAYER

Initialized with embedding matrix

02

3 - 1D CONV. LAYERS

- 256 filters each
- kernel size of 3 and
- ReLU activation
- Maxpooling of size 3

03

DROPOUT LAYERS

- 50% drop out
- after 2 an 3 layer

04

GLOBAL MAX POOLING + 2 DENSE LAYERS

05

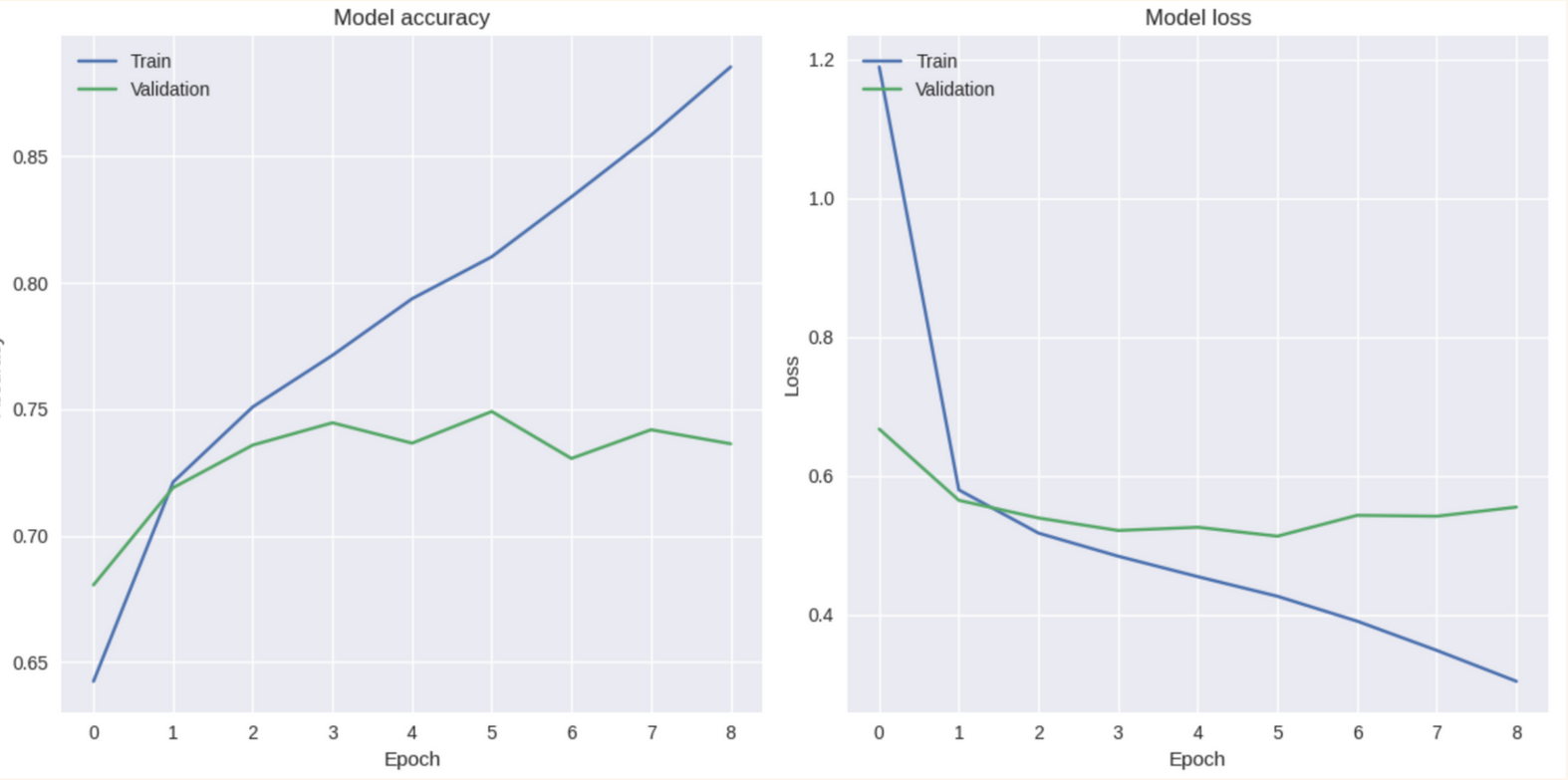
DENSE LAYER

- Sigmoid as activation

Model Performance

OVERFITTING PROBLEM

from 5th epoch



IMPROVEMENTS

Model 2: reduced nodes in the layers + kernel regularization into the dense layers.

Model 3: removed callback dynamic learning rate + kernel regularizers from the dense layers.

Model 4: added two kernel regularizers to the first two layers of the neural network.

LSTM + CNN

capture long-term dependencies and spatial patterns simultaneously.

10 epochs

01

EMBEDDING LAYER

Non trainable

02

LSTM LAYER

- 128 units
- 20% drop out
- 20%recurrent drop out

03

1D CONVOLUTIONAL LAYER

- 128 units
- kernel size 5
- ReLu activation

04

GLOBAL MAX POOLING + 2 FULLY CONNECTED LAYERS

05

DENSE LAYER

- Sigmoid as activation

Model Performance

GOOD PERFORMANCES → **Epochs reduced** → **VERY INEFFICIENT**
but overfitting problem **(hours to complete each epoch)**

- CNN deliver superior performance on our dataset
- LSTM layers appear to introduce instability
- HYBRID ARCHITECTURE goes in overfitting
- BEST SOLUTION: simpler architecture

Final Considerations

TF-IDF — **Logistic Regression** → 76% Accuracy, 76% F1 score

Word Embedding GloVe — **Baseline model 1** → 77% Accuracy, 77% F1 score

Baseline model 2 → Same performances of Baseline model 1 but less balanced measures between classes

Balanced dataset → No need to prioritize F1 score

Topic modeling Pre-Processing

01

SHUFFLING DATASET

02

WORD NET LEMMATIZER

Application of lemmatization on the text

03

SHORT WORDS REMOTION

Remotion of words shorter than 3 characters

04

FREQUENT WORDS REMOTION

Remotion of words that appear in more than 10000 documents

Pipeline

1. Choose a model

2. Select a subset

3. Find from 2 to 25 topic

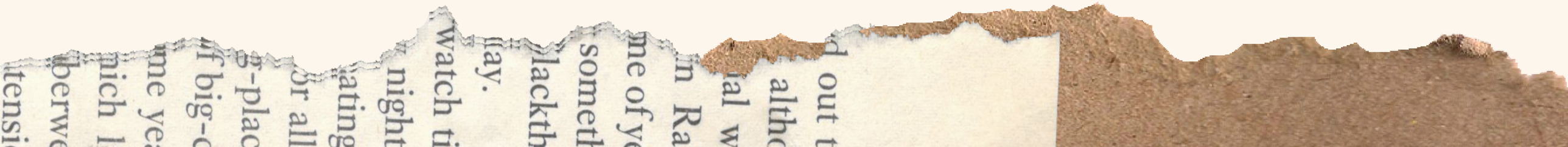
4. Change the size of the subset

5. Find the best result for that model

6. Change the model

7. Repeat point 2, 3, 4 and 5

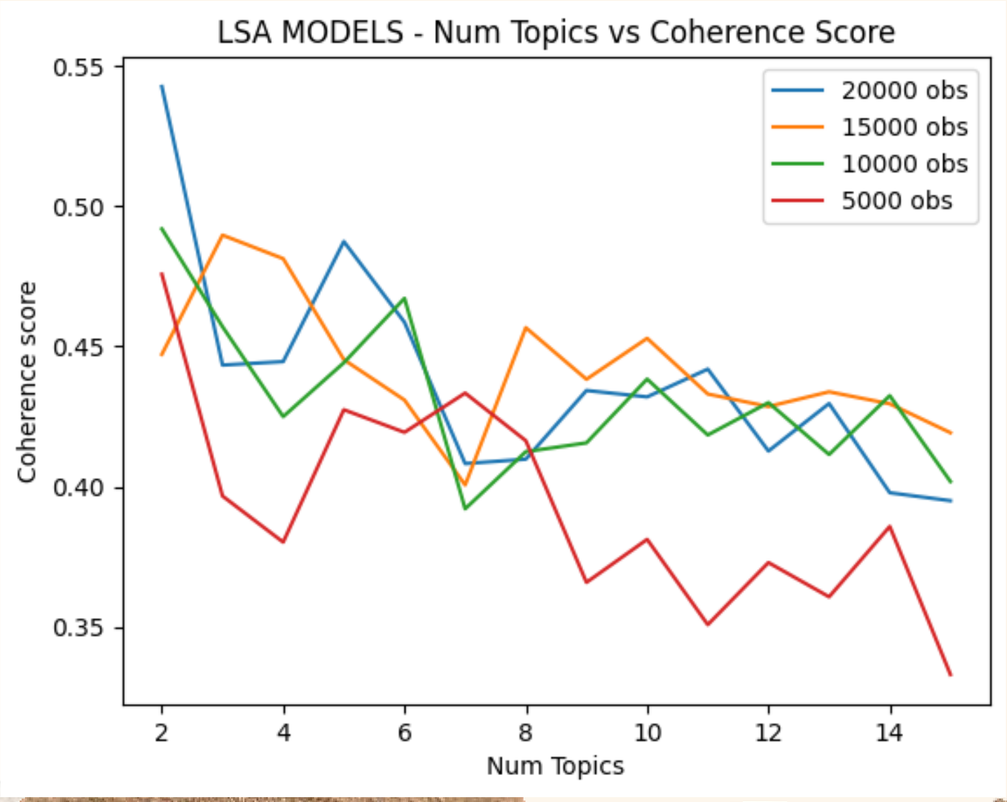
8. Find the best result overall



LSA

Latent Semantic Analysis: similar documents contain similar word frequencies for certain words

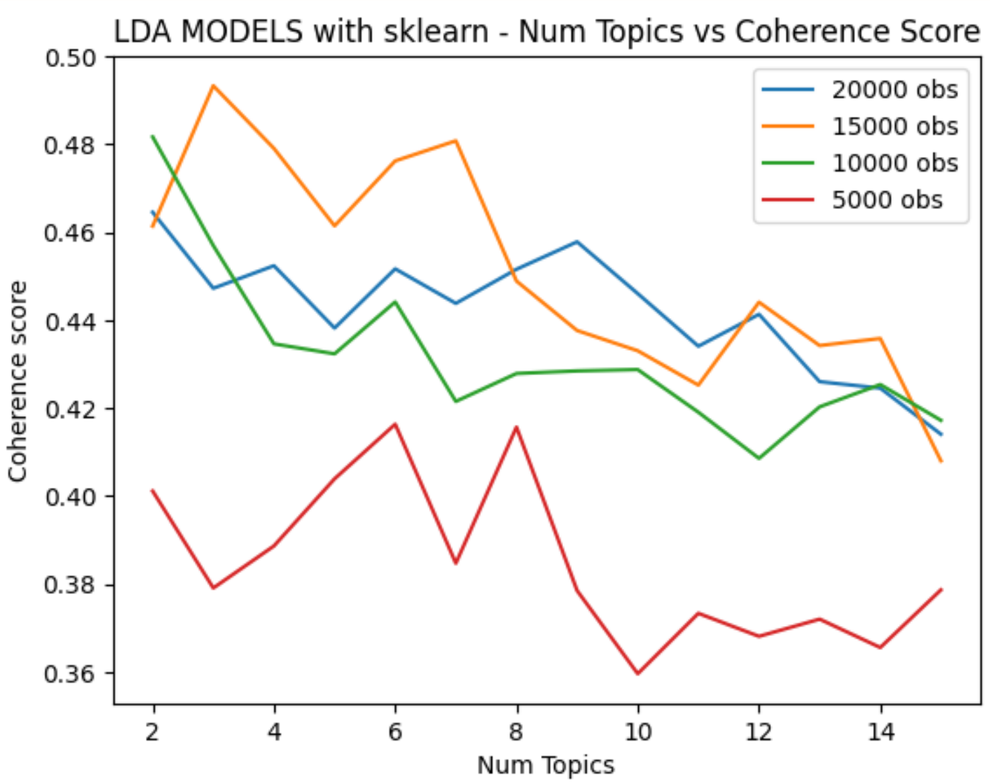
- **LsiModel() function**
- **Document-term Matrix**
- **Fast truncated SVD**
- **Syntactic and semantic information ignored**



LDA

Latent Dirichlet Allocation: documents are mixtures of topics, topics are mixtures of words. Both topics and words distributions are Dirichlet distributions

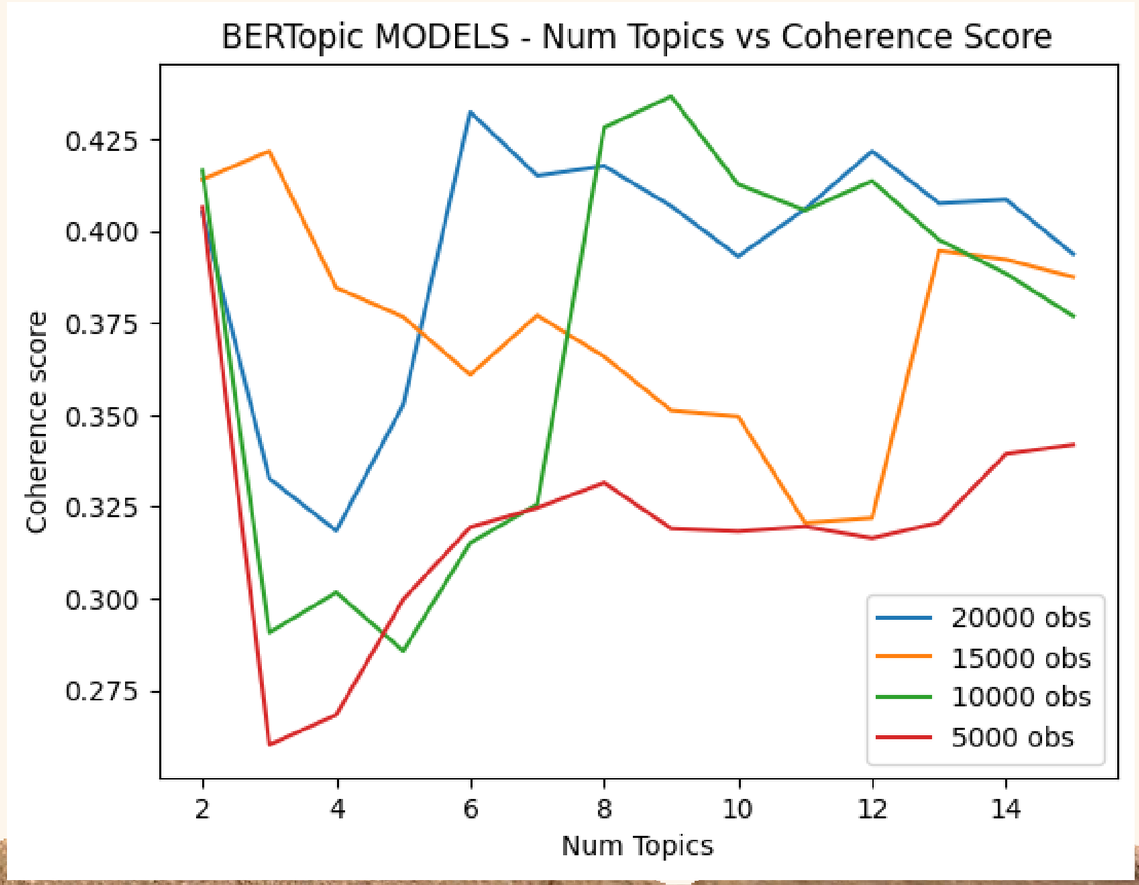
- **LatentDirichletAllocation() from sklearn**
- **LdaModel() from Gensim**
- **LdaMulticore() from Gensim**



BERTopic

Bidirectional Encoder Representations from Transformers: contextual embeddings clustered together to identify topics

- **BERTopic()**
- **UMAP for clustering**



TOP2VEC

Top2Vec: semantically similar documents identify an underlying topic. Automatically identification of the optimal number of topics

- **Joint embedding of document and word vectors**
- **Dimensionality reduction of embedding of documents**
- **Identification of clusters of documents**
- **Creation of Topics and topic words**

Models Performances

Metric: coherence, give a values about the interpretability of each topic aiming to assess how well the words assigned to a particular topic align semantically

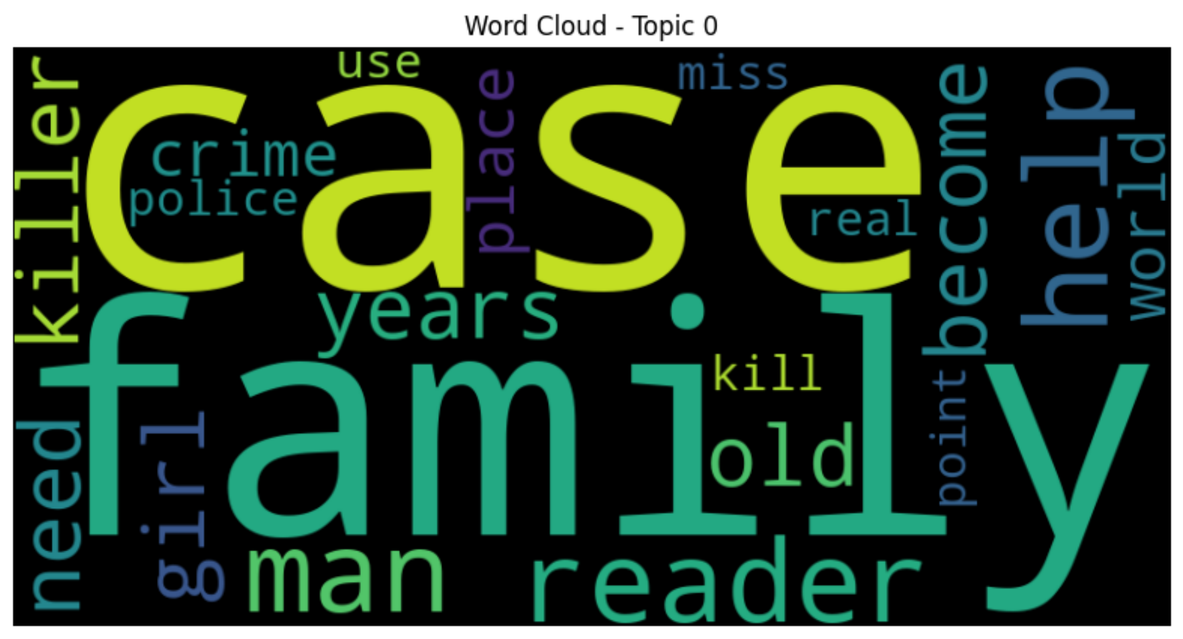
Subset - n° topics	LSA	LDA with sklearn	LDA with Gensim	LDA with Multicore	BERTopic
20k - 2	**0.542739**	0.464570	0.485323	0.419881	0.405016
15k - 3	0.489735	**0.493352**	0.466433	0.460943	0.421726
20k - 3	0.443380	0.447257	**0.490684**	0.473658	0.332595
10k - 2	0.491949	0.481712	0.440287	**0.479033**	0.416406
10k - 9	0.415634	0.428460	0.405664	0.382402	**0.436607**

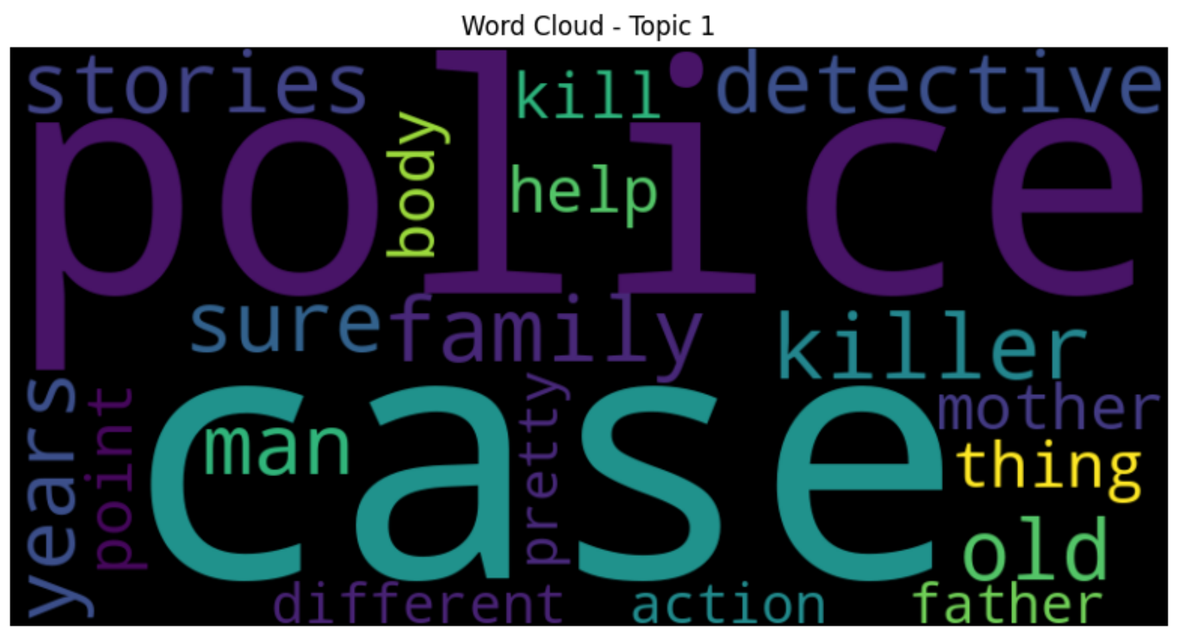
Number of topics found with Top2Vec: 3

LSA or LDA ?

Word Cloud

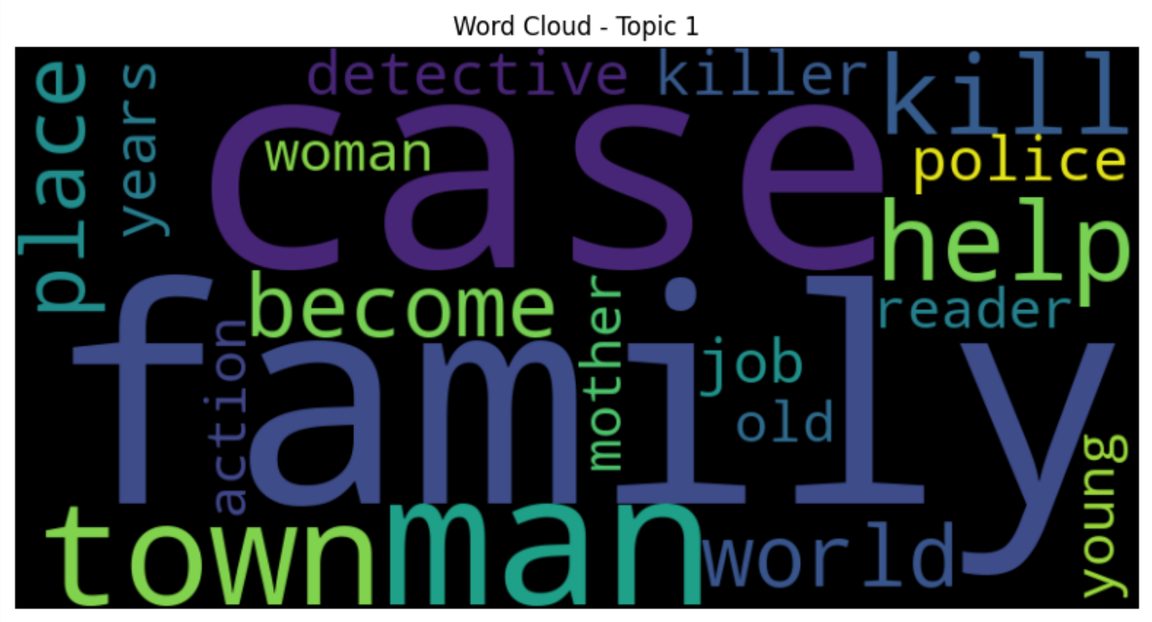
LSA best model word clouds

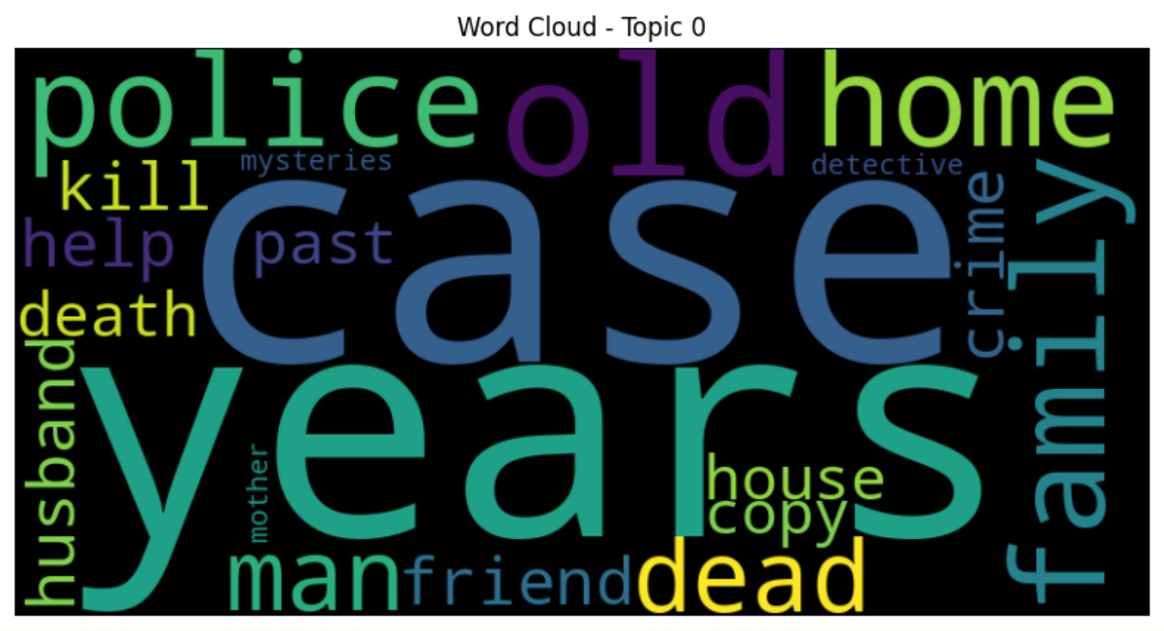




Word Cloud

LDA best model word clouds





Conclusion

TEXT CLASSIFICATION

Good results!

TOPIC MODELING

Bad results...
(in the future we can think about adding reviews of other genres book to obtain better results!)

Thank You

Brambatti Eleonora
Fracchia Camilla
Privitera Marta