Text categorization

DEAD team (Dmitry, Elena, Alexey, Dmitry)

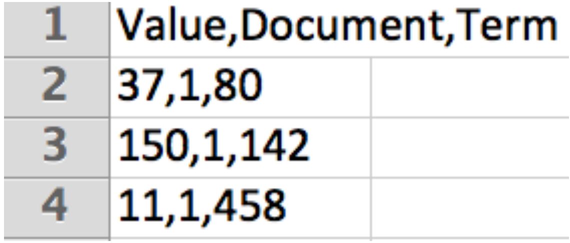
Problem

Input data:

- set of triples: (value, document, term)

Value - "importance" of the term in the article
Document - number of the article
Term - number of the term

10000x25640 sparse matrix of features

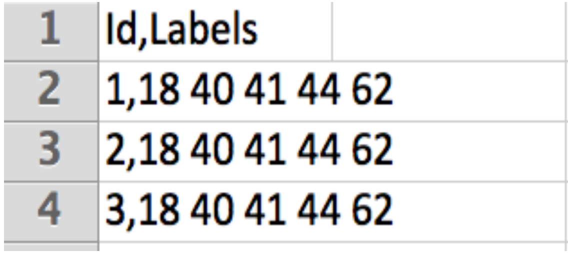


1	Value,Document,Term
2	37,1,80
3	150,1,142
4	11,1,458

Output data:

- One or several categories arranged to the text

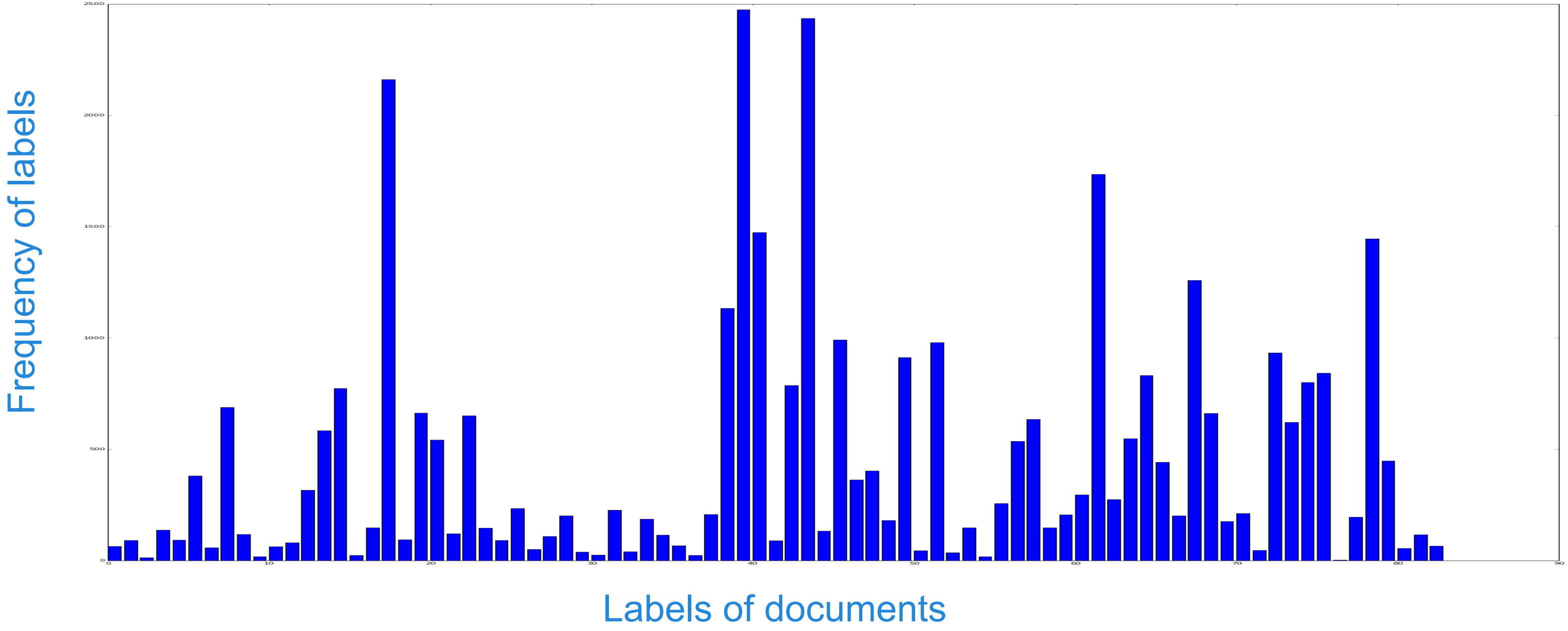
10000x83 matrix of labels



1	Id,Labels
2	1,18 40 41 44 62
3	2,18 40 41 44 62
4	3,18 40 41 44 62

Input data representation

Categories of documents distribution

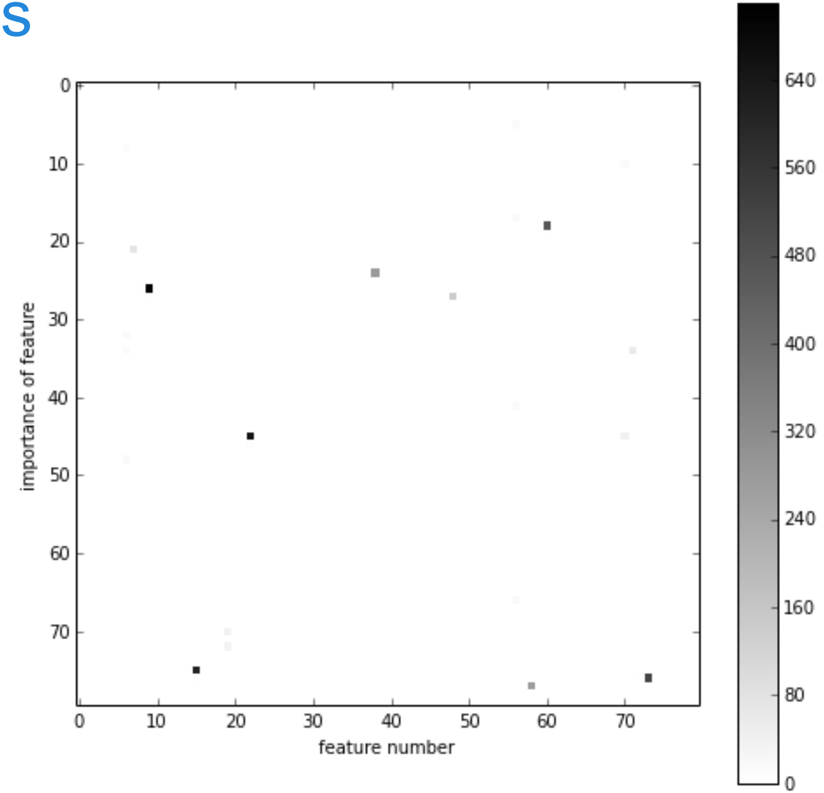


Classes are unbalanced, it leads to

Input data representation

Frequency of features in documents

The matrix of features is sparse, there are features which are extremely seldom distributed.



Methods

1. Multilabel classification approach

2. Two steps classification approach: multilabel with binary classification

Preprocessing data

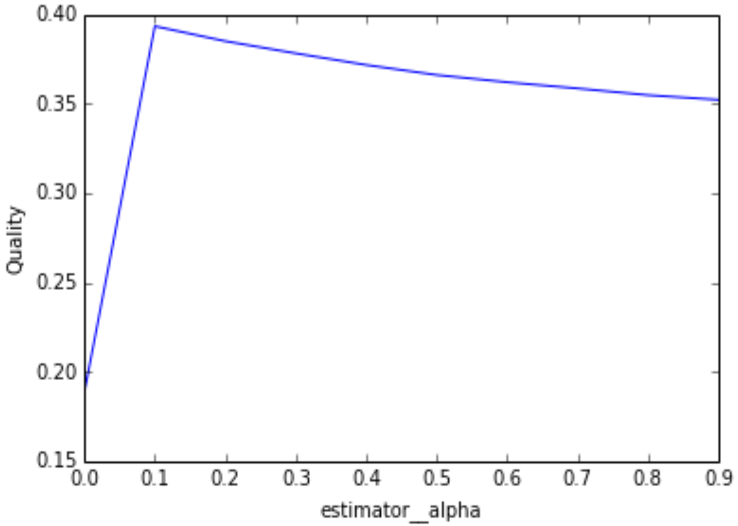
Before classification transform raw feature vectors into a representation that is more suitable for estimators

TF-IDF: term frequency–inverse document frequency

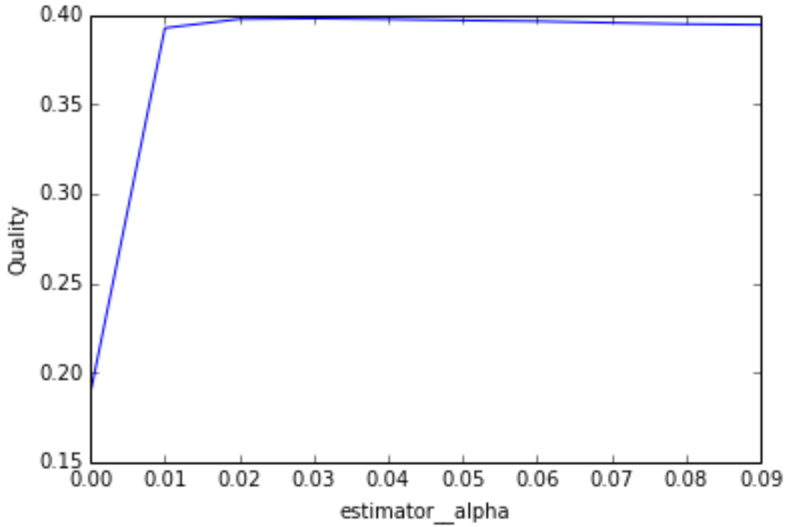
TF-IDF measures importance of words in documents

Choosing Naive Bayes

Gaussian NB



MultinomialNB

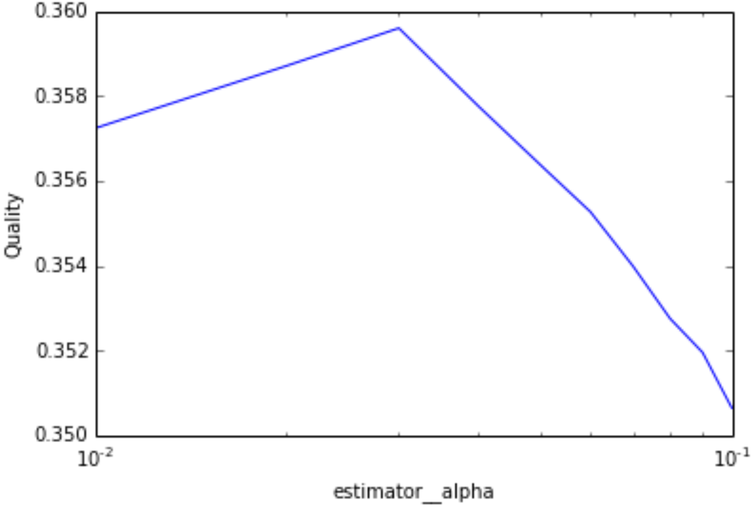


The result of classification depends on prior distribution of features

Preprocessing: TF-IDF for NB

f1 score = 0.3596

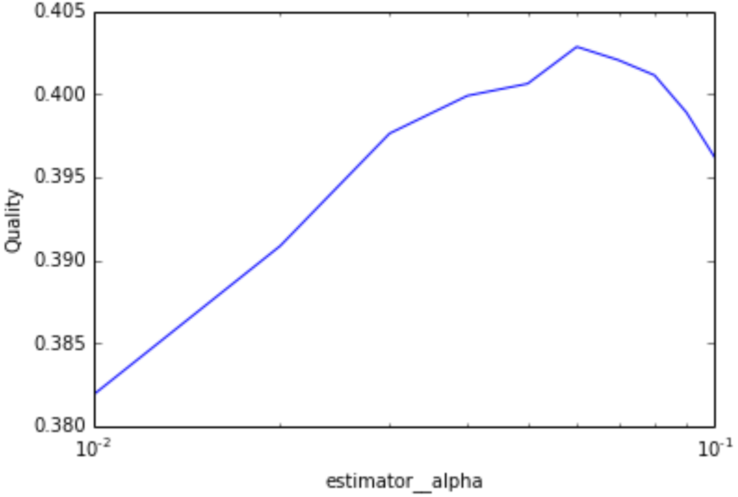
alpha = 0.03



before

f1 score = 0.4028

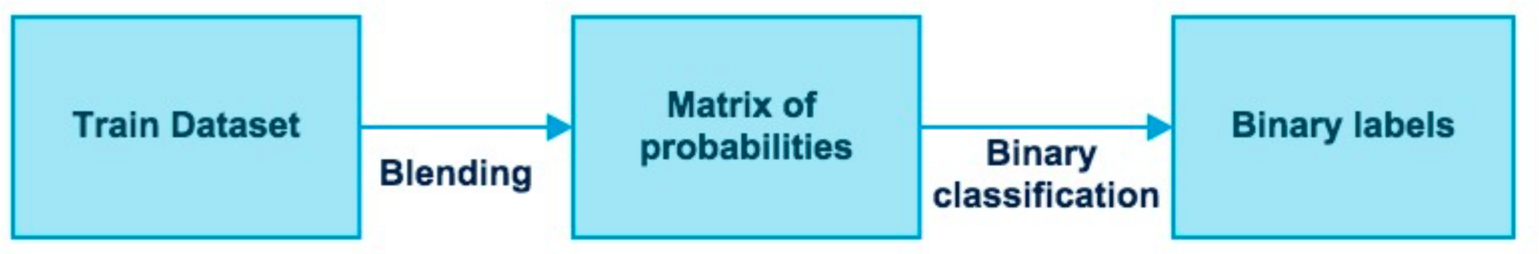
alpha = 0.06



after

Preprocessing data such as TF-IDF increases score of classification

Blending


Train Dataset

Blending

Matrix of probabilities

Binary classification

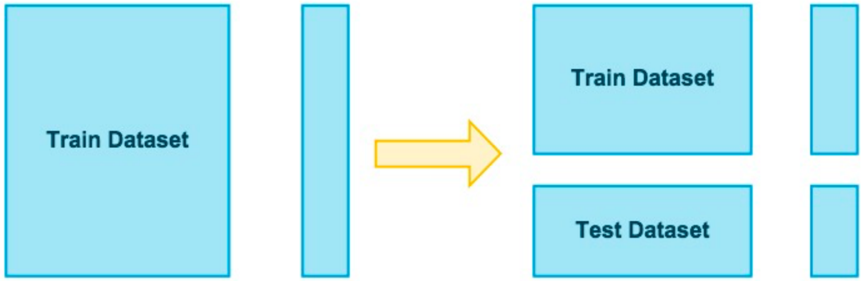
Binary labels

1. Blending - mix up outcomes from many models and improve final result
   αNaiveBayes + βLogisticRegression + (1-α-β)KNN

2. Binary classification - use matrix of probabilities as features for binary classifier or define threshold for binarization

Parameters for optimization

1. Find optimal split of train set into two sets: one for training, one for testing and finding parameters



Train Dataset

Train Dataset

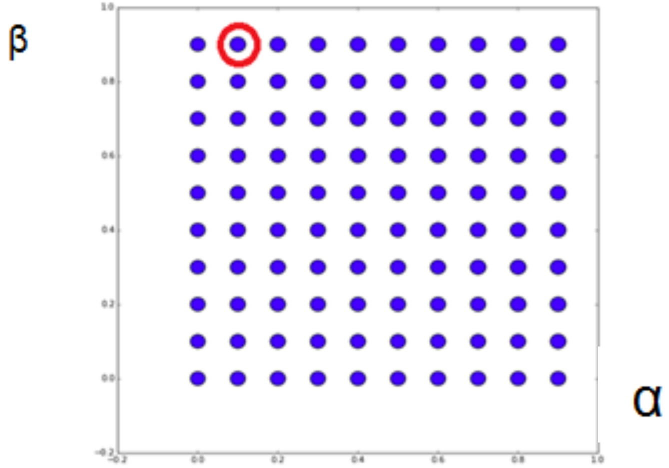
Test Dataset

2. Choose optimal blending coefficients:

alfa optimal = 0.1 (weight for NB algorithm)

beta optimal = 0.9 (weight for LR algorithm)

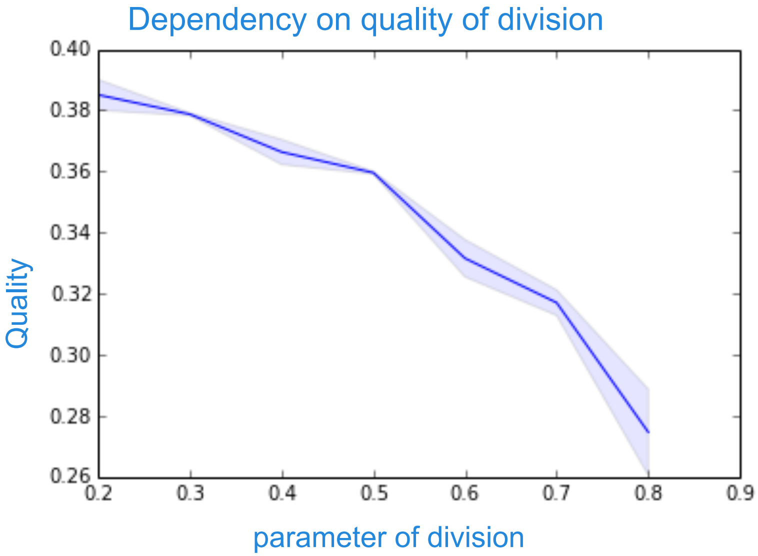
1-alfa-beta = 0 (weight for KNN algorithm)



Splitting the set

Searching the best test/train sets division:

- High quality
- Low deviation



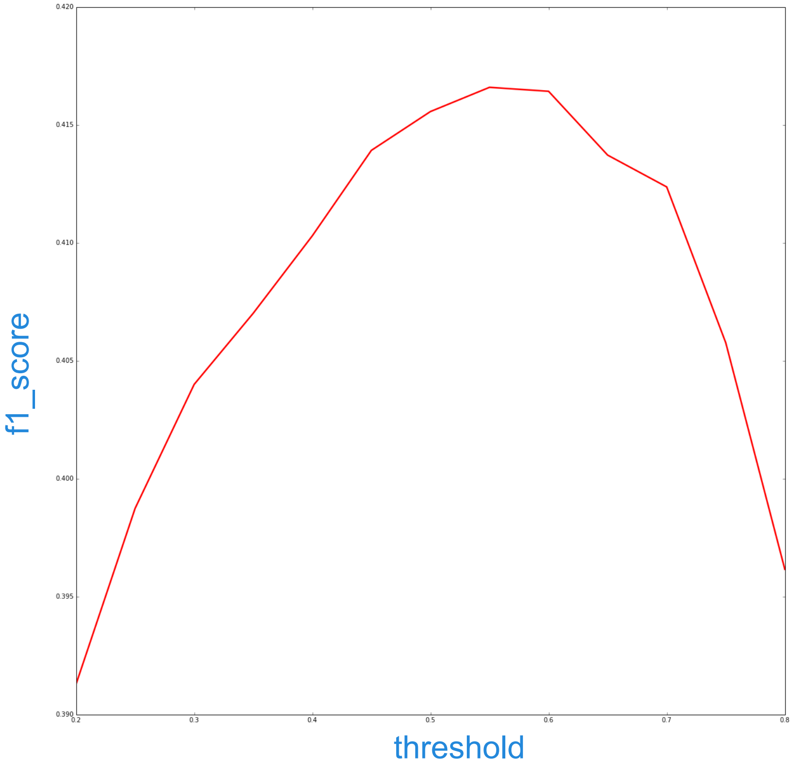
Blending: optimal threshold

Optimization of binarization threshold:

extremum conditions:

f1_score = 0.41659

threshold = 0.55



Results

The best result in competition was given by **Naive Bayes** algorithm - 0.45860

Blending approach gives worse result - 0.42778

Team project

Dmitry Zarifyan

Data visualisation

Elena Shirokova

Implementation of multilabel classification and blending

Alexey Boyko

Making of final presentation

Dmitry Zhestkov

Implementation of multilabel classification and blending

Thank you for your attention!