



MIDDLE EAST TECHNICAL UNIVERSITY

CENG 463

Term Project

Onur Yılmaz

Contents

1. Introduction	3
2. Implementation Idea	4
2.1. Gathering and Preprocessing Input	4
2.1.1. Dividing Input into Sets	5
2.2. k-fold Training and Evaluation	5
2.2.1. Feature Sets	5
2.2.2. Training and Evaluation	6
2.2.3. Analysis of Cross Validation	6
3. Analysis of Implementation	8
4. Possible Problems and Improvements	10
5. Conclusion	11
6. References	12

1. Introduction

In this term project, a sentiment analysis system for Turkish is built. This system is focused on Turkish movie reviews and classifying them as positive or negative. For movie reviews, Beyazperde web portal [beyazperde.com] and their ratings are used for development and test purposes.

In this report, all steps of this project are presented in detail. Firstly, gathering and preprocessing data is explained to make clear how the data is taken as input from Beyazperde website. Following that, how the dataset is divided is mentioned and training / evaluation steps are presented. Finally, analysis of this implementation is made in the statistical approach and possible problems and improvements are given.

Throughout the report, code segments are provided or mentioned which are the parts of all source codes provided with this report.




2. Implementation Idea

Implementation steps of this project can be gathered into two as gathering and preprocessing the input data and k-fold cross validation of training. Detailed explanations of these steps will be presented in this section.

2.1. Gathering and Preprocessing Input

Since this term project is based on sentiment analysis of movie reviews written on beyazperde.com cinema portal, randomly selected 250 review URLs are taken as input data. In this project, labels of reviews, which are assigned according to their ratings will be estimated by doing sentiment analysis on the review text. Therefore, extraction of review text was necessary for quick implementation. This extraction process is done with the help of BeautifulSoup[1] and HTMLParser[2] packages. For this purpose, “returnText” function is implemented for taking URL and returning the text of the related division of the review.

Labels which are assigned to reviews are shown in Table 2.1.1 below. Considering half-star values, any rating below 3-star is labeled as negative whereas any rating above 3-star is labeled as positive.

Rating	Label
 1,0	NEGATIVE
 2,0	NEGATIVE
 3,0	NEUTRAL
 4,0	POSITIVE
 5,0	POSITIVE
Table 2.1.1: Assignment of labels	

2.1.1. Dividing Input into Sets

As mentioned in the last section, 250 review text and their labels are taken for training and evaluation of the implementation. Before training, validation set is taken away as 50 URLs to analyze performance of the method on unseen data. Rest of the 200 URLs are used for k-fold cross validation and training. These URLs are saved into “trainListMake.py” and “validationListMake.py” and then written into files using YAML package [3] for further steps.

2.2. k-fold Training and Evaluation

In this project, k-fold cross validation is used for training and extracting a final classifier. For this reason, firstly feature sets are presented and then cross validation steps are mentioned. Finally, analysis of this method is described.

2.2.1. Feature Sets

For classifying review texts, a combination of bigrams and bag-of-words model [4] is implemented. Firstly, bag-of-words model is used for counting the occurrences of the words and then occurrences of bigrams are counted. These counts are saved into “features” set and used for further steps of training. In order to extract opinions of the review writers, how often they use words and collocations is thought to be a good idea for implementation. For this reasoning, a function named “document_features” and a helper function named “document_features_helper” which take the review text and returns a set of features is used.

2.2.2. Training and Evaluation

In order to train and evaluate a classifier, 200 URLs which are saved in the prior sections is randomly divided into training and test set by 80% and 20% respectively. Then, a Naïve Bayes Classifier is trained with the training set and accuracy levels are recorded. With the idea of 10-fold cross validation, training and test sets randomly selected 10 times and the classifier is trained again. For each iteration, accuracy levels are saved in order to find a classifier which has a good trade-off between over fitting and accuracy.

2.2.3. Analysis of Cross Validation

When the accuracy levels of 10-fold validation is checked, the following plot is constructed. In this step, a classifier which is good at estimating the label of unseen data is tried to be found. Therefore, for each iteration, accuracy levels of training and test sets are plotted and a classifier which is good at test set is found. As can be seen from the Figure 2.2.3.1 below, the classifier in the iteration #9 is has the best accuracy level in test set.

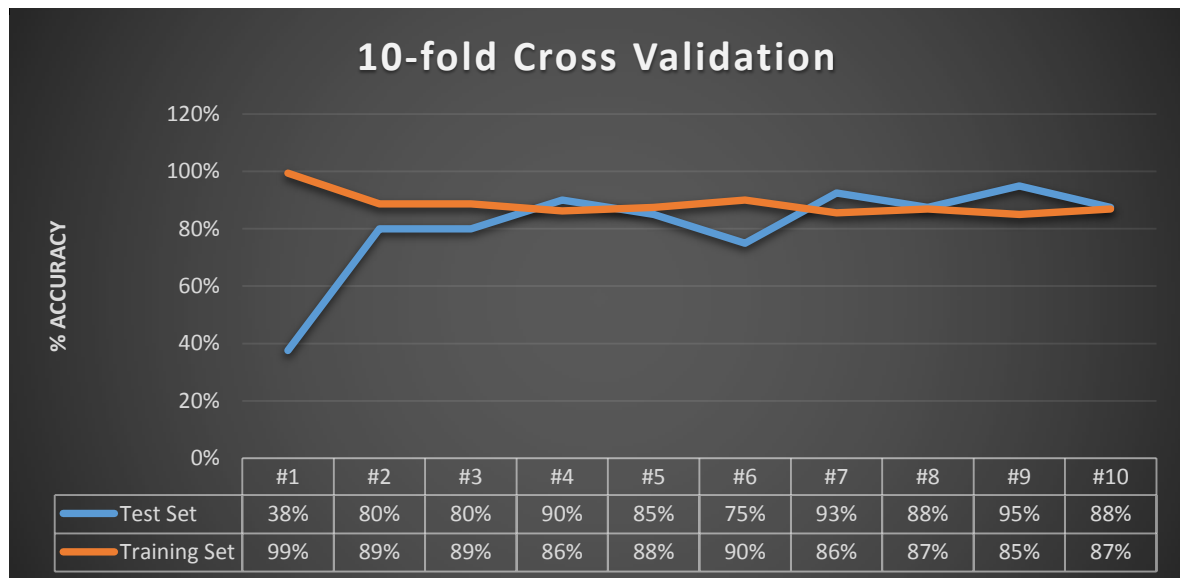


Figure 2.2.3.1: Accuracy levels of 10-fold cross validation

As mentioned above, classifier in the 9th iteration has the highest accuracy, however, in order to avoid over fitting, classifier in the 7th iteration is selected and will be analyzed for further steps. This classifier is saved into a file for further usage and a code segment is implemented in “Classifier.py” so that in Python this classifier can be used easily. The usage can be shown as the Figure 2.2.3.2 below.

```
>>> from Classifier import myClassify
>>> myClassify("http://www.beyazperde.com/filmler/film-191856/elestiriler-beyazperde")
URL is opening: http://www.beyazperde.com/filmler/film-191856/elestiriler-beyazperde
'POSITIVE'
```

Figure 2.2.3.2: Usage of myClassify(URL) function

To use classifier, firstly import operation is made. Then, only the URL of the review is provided and internet connection is checked with the “URL is opening ...” text. Finally, classification result is provided as a label.

3. Analysis of Implementation

In this step, the selected classifier in the last section will be analyzed further. Firstly, informative features will be presented then the performance of the method on a completely unseen data will be analyzed.

Firstly, when the most informative features are checked, the following table is constructed below.

#	Word / Bigram	Count	Labels				Ratio		
1	hayatı	2	NEUTRAL	:	NEGATIVE	=	8.6	:	1.0
2	hal	3	NEUTRAL	:	POSITIVE	=	8.1	:	1.0
3	hedef	1	NEGATIVE	:	POSITIVE	=	7.4	:	1.0
4	ada	9	NEUTRAL	:	NEGATIVE	=	6.8	:	1.0
5	adı	1	POSITIVE	:	NEGATIVE	=	6.7	:	1.0
6	bunun	2	NEUTRAL	:	POSITIVE	=	6.5	:	1.0
7	filmin	3	NEGATIVE	:	NEUTRAL	=	6.3	:	1.0
8	mi	15	NEUTRAL	:	POSITIVE	=	6.2	:	1.0
9	duygu	2	NEUTRAL	:	POSITIVE	=	5.9	:	1.0
10	tesadüf	1	NEUTRAL	:	POSITIVE	=	5.6	:	1.0
11	final	2	NEUTRAL	:	POSITIVE	=	5.6	:	1.0
12	yine	2	NEUTRAL	:	POSITIVE	=	5.6	:	1.0
13	ada	3	NEUTRAL	:	POSITIVE	=	5.5	:	1.0
14	sever	1	POSITIVE	:	NEUTRAL	=	5.4	:	1.0
15	hikayesi	2	POSITIVE	:	NEGATIVE	=	5.4	:	1.0
16	açısı	2	POSITIVE	:	NEGATIVE	=	5.4	:	1.0
17	çeşitli	1	POSITIVE	:	NEGATIVE	=	5.4	:	1.0
18	baskın	1	NEUTRAL	:	NEGATIVE	=	5.3	:	1.0
19	yeterince	1	NEUTRAL	:	NEGATIVE	=	5.3	:	1.0
20	olabilir	2	NEUTRAL	:	NEGATIVE	=	5.3	:	1.0

Table 3.1: Most informative features

Most informative features indicate some important issues. Firstly, usage of “hedef (target in English)” indicates tendency to be negative review. Secondly, question suffix “mi” in

Turkish highly effects ratio of neutral and positive reviews. Finally, the most important issue about this table was not seeing any bigrams as informative features. When this list is extended it is seen that the first bigram were listed as 219th in the most informative features.

#	Word / Bigram	Count	Labels			Ratio		
219	"ama çok"	1	NEGATIVE	:	POSITIVE	=	3.2	: 1.0

Table 3.2: Most informative feature list (extended)

Secondly, confusion matrix for the validation set is constructed. As illustrated in the Table 3.3 below, in general the classifier is good at catching "positive" label whereas it is not very good at correctly labeling "neutral" ratings. Overall accuracy level on this dataset is calculated at 64 %.

		Predicted		
		NEGATIVE	NEUTRAL	POSITIVE
Reference	NEGATIVE	10	-	4
	NEUTRAL	5	1	4
	POSITIVE	5	-	21

Table 3.3: Confusion matrix

When the confusion matrix is further analyzed, for negative reviews, 71 % of the reviews are labeled correctly; however, total of other 29 % are labeled positive. For neutral reviews, only 10 % accuracy level can be achieved. Finally, for the positive reviews, 81 % of the reviews are labeled correctly. This confusion matrix shows that the implementation method cannot differentiate neutral ratings correctly. As can be seen, it splits 50% of them as negative

and 40 % as positive which shows its drawbacks on estimating the correct label for neutral reviews.

4. Possible Problems and Improvements

When the relatively low accuracy of the method on validation set compared to test sets in k-fold validation steps is considered, some possible problems and improvements can be listed as following:

- Firstly, as most informative features show, count of words seem to be more important than count of bigrams. On the other hand, when semantics of the sentences are considered, bigrams should be more important because “great” means a positive comment without checking any other words around it; where “not great” means a negative comment where bigram of the same sentences are checked. Therefore, different weights for the counts can be implemented for words and bigrams.
- Secondly, instead of Naïve Bayes Classifier, Maximum Entropy classifier is tested and 52 % accuracy is achieved on the validation set. Although, most informative features in this trial were to be more realistic, due to low accuracy level it is not used. Therefore, with a more appropriate set of features, different classifiers can be implemented.
- Thirdly, as a different classifier, a decision tree model can be used, because most of the reviews are written in a well-organized manner where we can derive some rules and apply them for classification.

5. Conclusion

To conclude, in this term project, a sentiment analysis system for Turkish movie reviews is presented, where the data is gathered from Beyazperde web portal.

Steps and results of the implementation is presented in this report. Starting from how input data is gathered, k-fold cross validation and training and then finally analysis of the method is mentioned. The implemented method, achieved nearly 60% accuracy on completely unseen data and possible problems related to this accuracy level are presented with potential future developments.

6. References

- [1] BeautifulSoup 3.2.1
<http://pypi.python.org/pypi/BeautifulSoup>

- [2] HTMLParser - Simple HTML and XHTML parser
<http://docs.python.org/2/library/htmlparser.html>

- [3] YAML implementations for Python
<http://pyyaml.org>

- [4] Bag-of-words model
http://en.wikipedia.org/wiki/Bag_of_words_model