



Amirkabir University of Technology
(Tehran Polytechnic)

Topic-based sentiment analysis of Amazon reviews

Professor:

Dr. M.Akbari (akbari.ma@aut.ac.ir)

Teaching Assistant:

A.Malekzade (malekzadeh@ieee.org)

Student:

Roozbeh Bazargani(Roozbehbazargani@gmail.com , 9513703)

Mohammadreza Ardestani (ardestani.zr@gmail.com, **9513004**)

20, Jul, 2020

0) **Introduction and purpose**

- 0.1) Setups and input, output format and data frame clarification
- 0.2) Purposes

Part 1) **Preprocessing the data and preparing it**

- 1.1) cleaning Data
- 1.2) vectorizing
- 1.3) Chi-square

Part 2) **Phase One (Finding Sentiments)**

- 2.1) Unsupervised approach with “textblob” library
 - 2.1.1) Estimating Sentiments
 - 2.1.2) Extrapolating ratings
- 2.2) Deep learning approach

part 3) **Phase Two (Extracting Topics)**

- 3.0) Training LDA model
- 3.1) Finding soft/fuzzy clustering of words
- 3.2) Finding words distribution for each Topic
- 3.3) Finding Topics-distribution for each document

part 4) **Evaluations and visualizations**

- 4.1) visualizing topic-sentiment graph of corpus
- 4.2) Evaluating and visualizing LDA method
- 4.3) Evaluating and visualizing “textblob” method
- 4.4) Evaluating and visualizing Deep learning method

Part 5) **API**

- 5.1) Offline Website
- 5.2) GitHub repository

Part 5) **Contributions chart**

0) Introduction

This report will be concise but thorough. If you need more detail, please contact us.

0.1) Setups and input, output format and data frame clarification

How to run:

- I.** At first you should have been installed all required libraries and Jupyter notebook and website.
- II.** For running website you are supposed to run it with python 3 and then use this link for using website (link: <http://127.0.0.1:5000/>)
- III.** For running Notebook, I have shared the whole folder with you and you can download all text and python on that folder and run the notebook on your local machine or just use jupyter notebook on colab.
- IV.** Make sure you have read the reports before last steps.

Purpose and business plan:

As it's conspicuous from the title of the project we have 2 main phases. At first, we find sentiment of reviews and then we extract topics. Along the way we graph topic-based sentiment analysis of reviews and evaluate both main phases.

We will use Textblob library and Deep learning approach for finding sentiment and we will get 80% accuracy for textblob and 50% accuracy for deep learning approach .

There is no need to mention countless use cases of sentiment analysis, since we can use it for any work for finding more insightful details.

and for Topic extraction, we can think of it as a space-reduction function that maps document with many words to just a few number of topics

Phase 1) **Preprocessing the data and preparing it**

1.1) cleaning data:

We go through following steps:

1. removing user names
2. removing numbers
3. removing URLs
4. removing punctuations
5. removing stopwords
6. removing words with length less than 3
7. transform to the lower case
8. stemming

1.2) vectorizing

1.3) Chi-square

Phase 2) Phase One (Finding Sentiments)

2.1) textblob approach

TextBlob Sentiment



Tom De Smedt

University of Antwerp | UA · Computational Linguistics & Psycholinguistics Research Center (CLIPS)

il 7.62 · PhD

```
<word form="abhorrent" wordnet_id="a-1625063" pos="JJ" sense="offensive to the mind" polarity="-0.7" subjectivity="0.8" intensity="1.0" r
<word form="able" cornetto_synset_id="n_a-534450" wordnet_id="a-01017439" pos="JJ" sense="having a strong healthy body" polarity="0.5" su
<word form="able" wordnet_id="a-00001740" pos="JJ" sense="(usually followed by 'to') having the necessary means or skill or know-how or a
<word form="able" wordnet_id="a-00306663" pos="JJ" sense="having inherent physical or mental ability or capacity" polarity="0.5" subjecti
<word form="able" wordnet_id="a-00510348" pos="JJ" sense="have the skills and qualifications to do things well" polarity="0.5" subjectivi
<word form="above" cornetto_synset_id="n_a-504850" wordnet_id="a-00125993" pos="JJ" sense="appearing earlier in the same text" polarity="
<word form="abridged" cornetto_synset_id="d_a-9176" wordnet_id="a-00004413" pos="JJ" sense="(used of texts) shortened by condensing or re
<word form="abrupt" cornetto_synset_id="n_a-505100" wordnet_id="a-00640520" pos="JJ" sense="surprisingly and unceremoniously brusque in m
<word form="abrupt" cornetto_synset_id="n_a-529169" wordnet_id="a-01145151" pos="JJ" sense="extremely steep" polarity="0.0" subjectivity=
<word form="abrupt" wordnet_id="a-01143585" pos="JJ" sense="exceedingly sudden and unexpected" polarity="0.0" subjectivity="1.0" intensit
<word form="abrupt" wordnet_id="a-02294122" pos="JJ" sense="marked by sudden changes in subject and sharp transitions" polarity="0.0" sub
```

In a nut shell, TextBlob library has unsupervised methods for finding polarity and subjectivity (and etc) of a text. A lot of linguists, namely Tom De Smedt, have come together and have figured out all different aspect of words manually. Based on this great job that has done recently, we are able to estimate polarity (means sentiment score that is between -1 and +1) even better than KNN or Bayes which have expensive calculation. In our case, we have reached 0.799 percent accuracy for estimating polarity (sentiment) of reviews. We will explain this in “**Evaluation of TextBlob**” part.(More info about textblob for interested reader, perhaps you!, : [link1](#) , [link2](#))

2.1.1) Estimating Sentiments

We help of polarity function we can easily find sentiment score and map it to {Pos , Neg , Neu}.

2.1.2) Extrapolating ratings

For finding ratings we need to extrapolate them with respect to current behavior of users. For this purpose, beside defining a linear map from sentiment score to {1,2,3,4,5}, we define a non-linear function that is trained by ratings distribution of the reviews in data set.

2.2) Deep learning approach

▼ DL method

Constructing X_train and Y_train

```
X_train = np.concatenate((chi2_vectorized.toarray(), np.array([verified]).T, np.array([vote]).T), axis=1)
Y_train = np.array(scores)
```

Importing libraries and initializing parameters

```
from keras.layers import Dense, Embedding, Conv1D, GlobalMaxPooling1D
from keras.models import Sequential
from keras.utils import to_categorical
from sklearn.utils import class_weight
from sklearn.metrics import classification_report

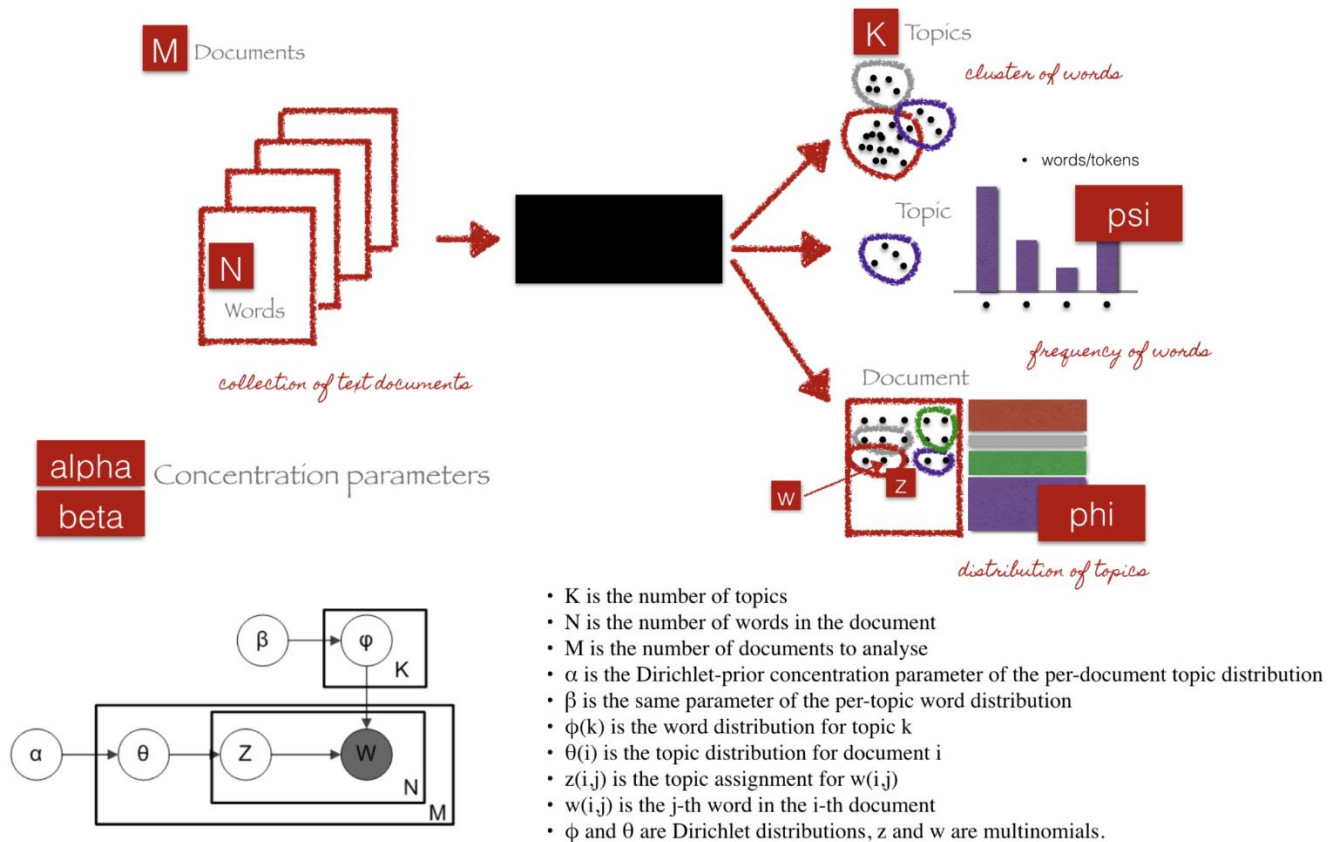
[ ] batch_size = 64
    embedding_dims = 16
    filters = 128
    kernel_size = 3
    epochs = 5
    max_features, maxlen = X_train.shape

    class_weights = class_weight.compute_class_weight('balanced',
                                                    np.unique(scores),
                                                    scores)
    class_weights = {0: 0, 1: class_weights[0], 2: class_weights[1], 3: class_weights[2], 4: class_weights[3], 5: class_weights[4]}

    print(max_features, maxlen)
    print(Y_train.shape[0])
    print(class_weights)
```

part 3) Phase Two (Extracting Topics)

At first, we explain what is exactly the problem?



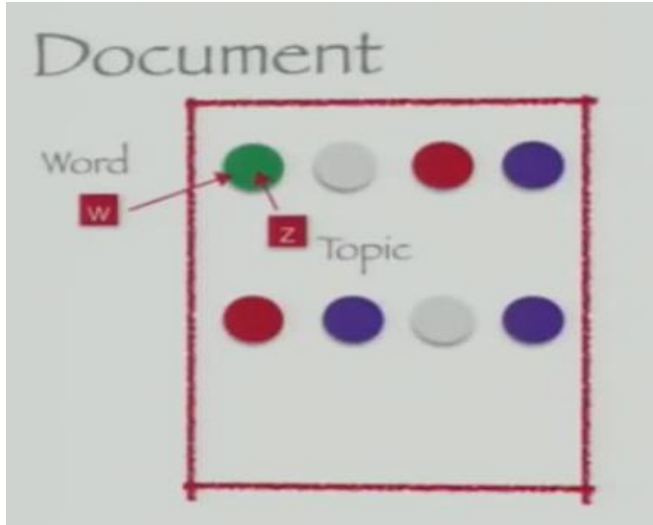
As an input we receive M document. Then we pass them into the black box that is LDA (hopefully it will not remain black box after this explanations) then we do some works (that will be explained in 3.0). As an output we have 3 things:

- 1) soft/fuzzy clustering of words ,
- 2) words distribution for each Topic ,
- 3) Topics-distribution for each document

3.0) Training LDA model

This is an iterative algorithm with the following steps:

- 1) Initialize parameters (M , N , K , iterative steps, α , β , ...)
- 2) Initialize topic assignments randomly



3) Iterate:

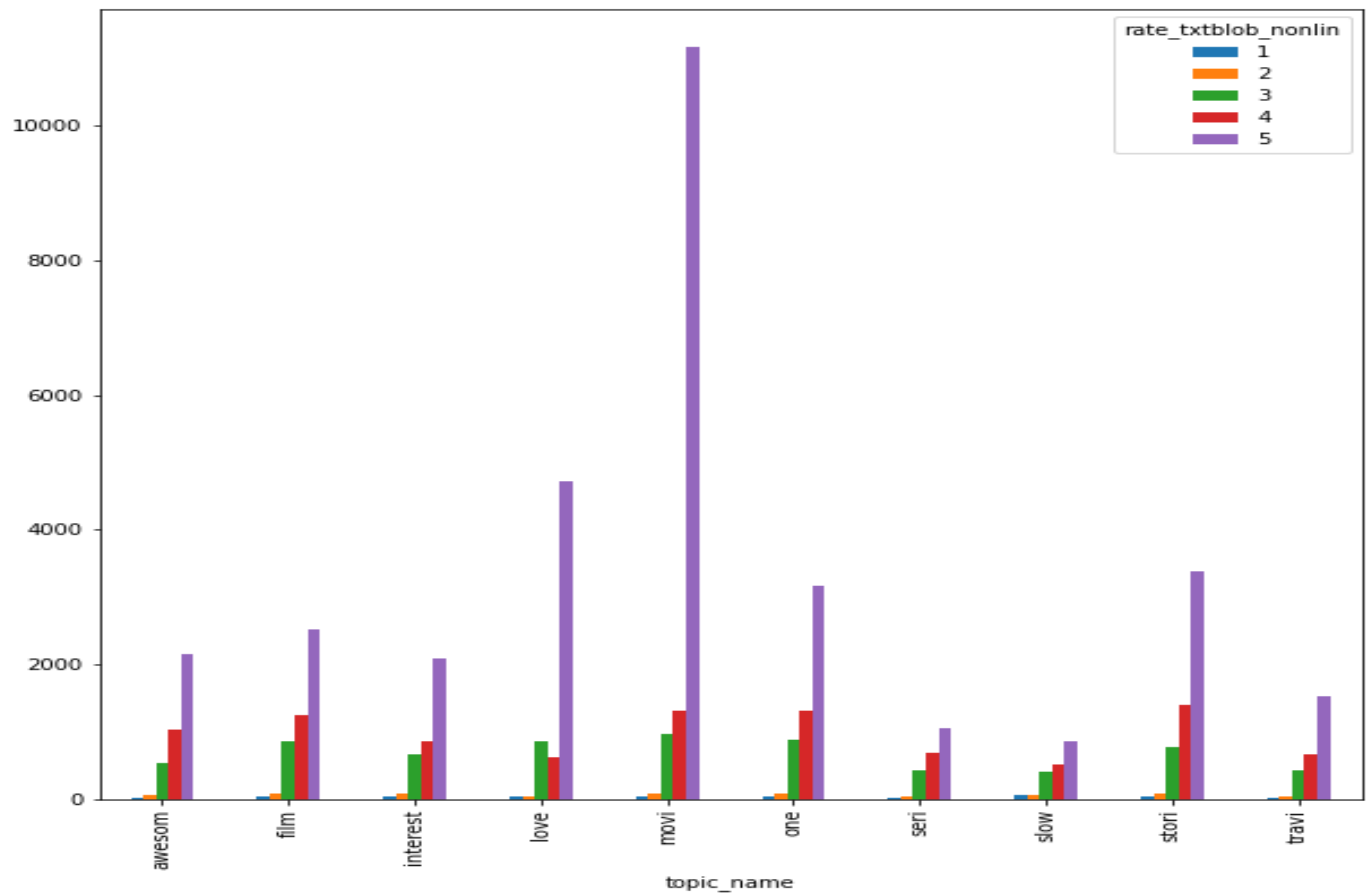
for each word in each document:

Resample topics for words, given all words and their current topic assignments

4) getting the results

part 4) **Evaluations and visualizations**

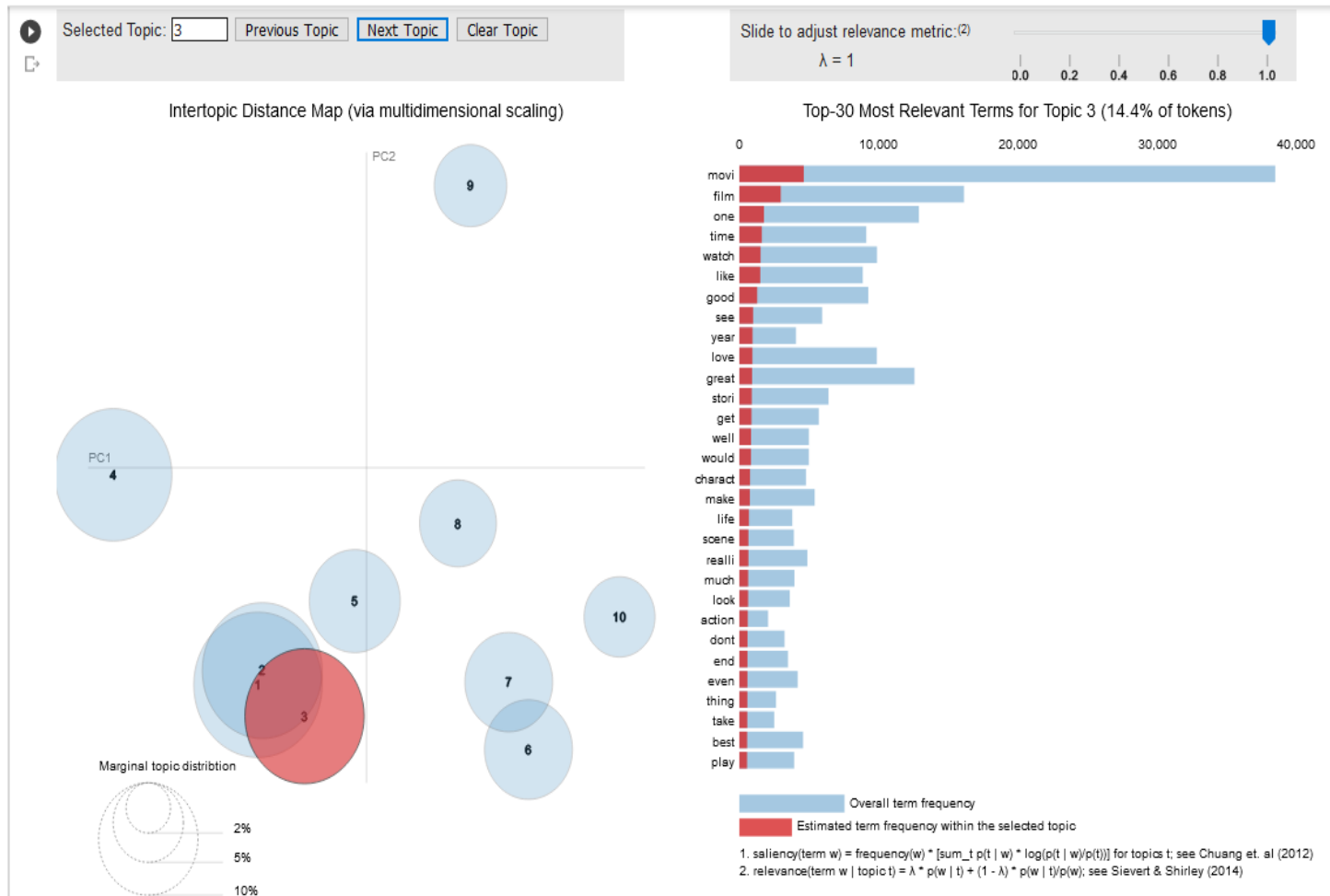
4.1) visualizing topic-sentiment graph of corpus



4.2) Evaluating and visualizing LDA method

We have 3 important factor here:

- 1) area of circles
- 2) distance of their centers
- 3) lambda parameter



4.3) Evaluating and visualizing “textblob” method

Overall accuracy for textblob is 0.79 percent.

```
[84] correctNumber = evaluator(init_ratings, labels)
      OverallAccuracyTextblob = round((correctNumber/N), 5)
      print(OverallAccuracyTextblob)
```

0.79908

4.4) Evaluating and visualizing Deep learning method

```
16/16 [=====] - 1s 51ms/step
          precision    recall  f1-score   support

     1         0.14      0.31      0.19         49
     2         0.05      0.21      0.08         38
     3         0.12      0.20      0.15         60
     4         0.22      0.05      0.08        153
     5         0.76      0.66      0.70        698

 accuracy              0.50         998
 macro avg           0.26      0.28      0.24         998
 weighted avg       0.58      0.50      0.53         998
```

```
[ ] model.summary()
```

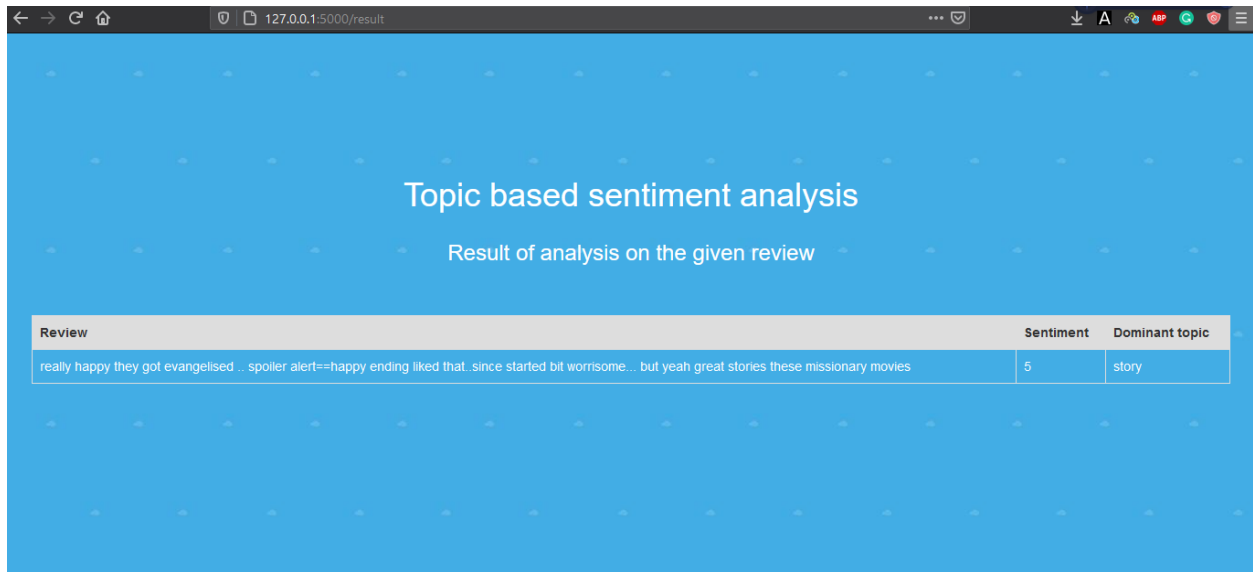
```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 2002, 16)	799216
conv1d_1 (Conv1D)	(None, 2000, 128)	6272
global_max_pooling1d_1 (Glob	(None, 128)	0
dense_3 (Dense)	(None, 128)	16512
dense_4 (Dense)	(None, 64)	8256
dense_5 (Dense)	(None, 6)	390

```
=====
Total params: 830,646
Trainable params: 830,646
Non-trainable params: 0
=====
```

Part 5) **API**

5.1) Offline Website



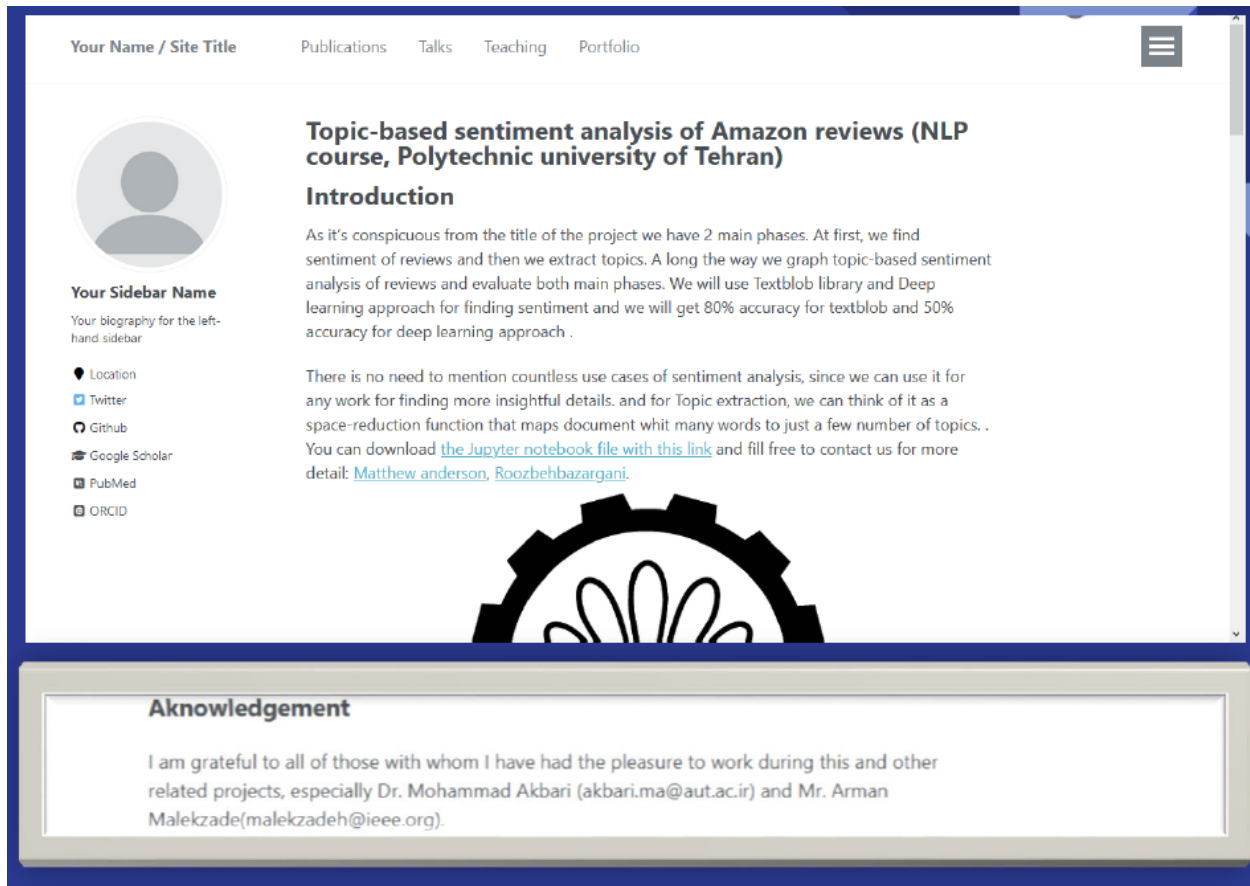
The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/result". The page has a blue background with a pattern of small white dots. The main heading is "Topic based sentiment analysis" in white text. Below it, a subtitle reads "Result of analysis on the given review". A table is displayed with three columns: "Review", "Sentiment", and "Dominant topic". The table contains one row of data.

Review	Sentiment	Dominant topic
really happy they got evangelised .. spoiler alert==happy ending liked that.. since started bit worrisome... but yeah great stories these missionary movies	5	story

5.1) GitHub

“If a tree falls in the forest and no one hears it did it really fall? “

In order to make our work available for anyone who this project is of interest to him, we have created a GitHub repository.



[GitHub repository of the project's link](#)

Part 5) **Contributions chart**

**We both believe we participated equally in this long road.
But in case you need more detail, this chart could be helpful.**

2	Work \ Participation Percent	Roozbeh	Mohammadreza
3	Writing Summary	50	50
4	Presentation	30	70
5	Proposal	70	30
6	Finding Data-Set	100	0
7	preprocessing	30	70
8	phase1(Txtblob implementing)	10	90
9	phase1(Deep learning and Chi2)	100	0
10	Implementation of Part 2	15	85
11	Evaluation and visualization	30	70
12	API (GITHB REPO)	0	100
13	API (off-line site)	100	0
14	Final Presentation	50	50
15	Final report	50	50

If you need more comprehensive detail, check this file: [Link](#) of work sheet

**We sincerely appreciate your time and your insightful comments.
Thanks.**