Stack Exchange Data Tagging

Tarik Buza
Graz University of Technology
tarik.buza@student.tugraz.at

Tarik Softić
Graz University of Technology
tarik.softic@student.tugraz.at

Afan Šečić
Graz University of Technology
asecic@student.tugraz.at

ABSTRACT

Machine learning is one of the most perspective fields in computer science, given its enormous potential in many different areas of science.

Considering that potential, this project has a goal to test if it can be used to tag certain posts given their title. In this report, we will describe our process of work in short word, giving the details about acquiring and filtering our dataset and creating and using each of these three approaches.

KEYWORDS

Machine learning, supervised learning, unsupervised learning, semi-supervised learning, clustering, labeling, classification, dataset

ACM Reference format:

Tarik Buza, Tarik Softić and Afan Šečić. 2018. *Stack Exchange Data Tagging*. June 2018. 5 pages

1 INTRODUCTION

In today's world, everything and everyone is connected via Internet. It enables us to exchange opinions, ask questions, and communicate with people all around the globe. But with that many options of communication, it can be sometimes really difficult to find what you are looking for. But it can be even harder to tell the computer what to look for.

Every one of us wanted to ask other people for their opinions on various questions. One of the best ways to do so is to use forums. Particularly looking at forums, we were very interested into making it easier to filter and find posts. And most common way of filtering posts on forums is – tagging. However, tagging a post is something many, if not all, of us forget or hate to do.

Looking at this problem, we wanted to find out if it could be possible to use potential of machine learning to automatically tag a post given only its title. Therefore, in this project we will focus only on post titles.

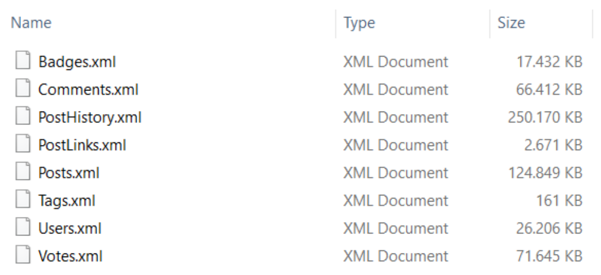
The scope of this project is to explore machine learning approaches and to compare their predictions with real tags and show their accuracy.

2 DATA ACQUISITION

In this section, we will explain the necessary steps needed to acquire data for this machine learning project. Besides that, we will explain the important features that will be extracted for each post.

As dataset for this project, we used stack exchange data dumps. For this particular project, we decided to use data from travel.stackexchange.com, since that data contains simple and straightforward language, which enables us to preprocess data and interpret the results in an easier manner.

Stack exchange data dump is an anonymized collection of posts, tags, comments and other content from a certain stack exchange site. List of all pages can be found on stack exchange website[1]. The dump itself can be downloaded from archive.org[2] website. There, data dumps are stored as archives of XML files zipped via 7-zip compression. It is frequently updated and maintained.



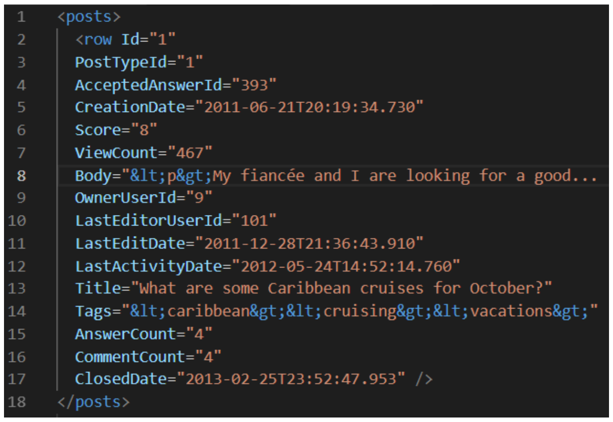
Name	Type	Size
Badges.xml	XML Document	17.432 KB
Comments.xml	XML Document	66.412 KB
PostHistory.xml	XML Document	250.170 KB
PostLinks.xml	XML Document	2.671 KB
Posts.xml	XML Document	124.849 KB
Tags.xml	XML Document	161 KB
Users.xml	XML Document	26.206 KB
Votes.xml	XML Document	71.645 KB

Figure 1: Content of Stack Exchange Data Dump

[1] data.stackexchange.com

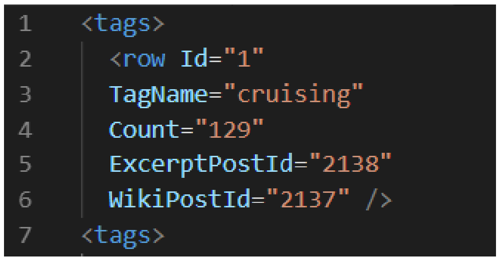
[2] archive.org/details/stackexchange

For this project, we used only two files from dump as our dataset: ‘Posts.xml’ and ‘Tags.xml’.



```
<posts>
  <row Id="1"
  PostTypeId="1"
  AcceptedAnswerId="393"
  CreationDate="2011-06-21T20:19:34.730"
  Score="8"
  ViewCount="467"
  Body="&lt;p&gt;My fiancée and I are looking for a good...
  OwnerUserId="9"
  LastEditorUserId="101"
  LastEditDate="2011-12-28T21:36:43.910"
  LastActivityDate="2012-05-24T14:52:14.760"
  Title="What are some Caribbean cruises for October?"
  Tags="&lt;caribbean&gt;&lt;cruising&gt;&lt;vacations&gt;"
  AnswerCount="4"
  CommentCount="4"
  ClosedDate="2013-02-25T23:52:47.953" />
</posts>
```

Figure 2: Information contained in Posts.xml – This raw dataset contains many information about posts, many of which are irrelevant for this project, and they will have to be removed



```
<tags>
  <row Id="1"
  TagName="cruising"
  Count="129"
  ExcerptPostId="2138"
  WikiPostId="2137" />
<tags>
```

Figure 3: Information contained in Tags.xml

As of June 16, 2019, Posts.xml contains 99064 posts and Tags.xml contains 1811 tags. Of course, this dataset needs to be restricted, filtered and preprocessed.

3 DATA ANALYSIS AND FILTERING

In this section, we will cover the process of restricting and filtering of our stack exchange dataset, to remove unnecessary information.

As we have mentioned in previous chapter, our current dataset contains a large quantity of data that needs to be adapted to our needs. Moreover, many posts do not contain tags at all. We decided to remove them, since we do not have a reference data to test the accuracy of our training. We also understand that this data could be used during training as train dataset but we decided to focus on presenting various machine learning concepts and having certain results in form of accuracy.

To demonstrate this property of our dataset, we implemented a function in Python that plots distribution of posts with and without tags, which can be seen on Figure 10.

As we can clearly see, majority of the posts is without tags, which leaves us with 36421 posts that contain tags.

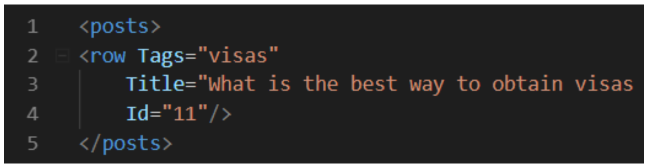
However, amount of 1811 tags is simply too much, since there are many tags that are used only once or not even once. That is the reason we decided to take top ten most common tags. We iterated through Tags.xml and found those tags, which is demonstrated in Figure 11.

From Figure 11, it can be seen that top ten tags are: ‘visas’, ‘usa’, ‘uk’, ‘air-travel’, ‘schengen’, ‘customs-and-immigration’, ‘transit’, ‘passports’, ‘indian-citizens’ and ‘train’. We are going to focus only on those tags while creating the algorithms.

Now we need to remove the posts that do not contain any of top ten tags. Figure 12 shows the ratio of posts with and without top tags.

Finally, we created new XML file ‘PostsCleaned’, which contains only those posts that are tagged with one or more of top ten tags. Of course, we removed unnecessary XML attributes. Creating and writing into XML file is done using *etree* imported from *lxml* Python library.

Therefore, our final dataset is created inside ‘PostsCleaned.xml’ and contains 21138 posts, which we consider to be large enough dataset to demonstrate different machine learning concepts, which will be discussed in next chapters.



```
<posts>
<row Tags="visas"
    Title="What is the best way to obtain visas
    Id="11"/>
</posts>
```

Figure 4: Content of row from final dataset – Dataset is stripped from all unnecessary attributes, as described in this chapter

We also created a graph to show the distribution of tags per post, which shows that the most of posts have 3 tags, while the least are with only one tag.

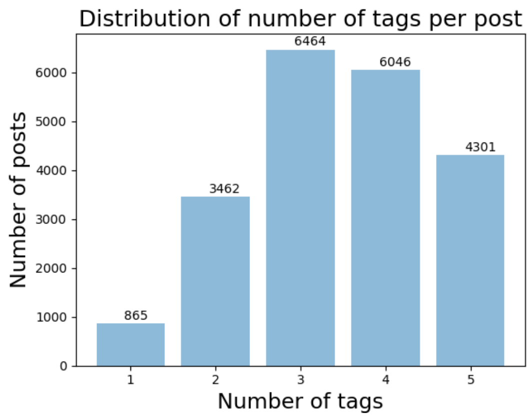


Figure 5: Distribution of number of tags per post

4 LEARNING PHASE

In this section, we will explain three different types of machine learning using previously defined and prepared dataset from travel.stackexchange.com.

Since we filtered unnecessary data, we can consider three approaches in machine learning – supervised, semi-supervised and unsupervised learning. For each approach, we have to appropriately adapt our dataset. Each approach consist of dividing our dataset into training and testing data. However, way of doing so differs between each approach.

4.1 Supervised learning

Supervised learning is the machine learning task of learning a function that maps an input to output based on input-output pairs. This function is formed out of labeled training data.

To create such data, we need to load data rows from previously created XML file into convenient form. We consider pandas Python library to be the easiest to use, so we loaded our data into pandas.DataFrame. Defined columns are 'id' and 'title', which are loaded from XML, and one column for each of top ten columns, which we want to predict. We fill these columns with value '1', if data row attribute 'tags' contains that tag. Otherwise, we set it to '0'. So, for example, if post has tags 'visas' and 'uk', both columns will have value 1, and others will be 0.

Our most important information is post title, because we want to create our tag prediction based on the title. We will create TF-IDF matrix, where each word is a feature, and a document is a post. Since, we do not want unnecessary words from each post to be inside our TF-IDF matrix, we will need to preprocess the titles. Every word will be:

- Transformed to lower case
- Singular
- Split over apostrophe character
- Removed if it is one letter word
- Lemmatized[3] using WordNetLemmatizer

After preparing the dataset for supervised learning, we can start the algorithm. Since our dataset consists of 21138 posts, we decided to randomly split it into 70% (around 14800) for training set and 30% (around 6300) for testing set. We also tried changing the ratio, but we got the best results using these settings.

Splitting training and testing data is done using train_test_split function from sklearn library. We also created Pipeline, which uses TfIdfVectorizer to create TF-IDF matrix from our training data, while also removing stop words, using nltk.corpus. Our pipeline also fits our categories into OneVsRestClassifier, which uses multinomial Naïve Bayes classifier.

[3] Lemmatization – mapping word into their base from the dictionary (*cats* to *cat*, *studying* to *study*, *better* to *good*)

4.2 Semi-supervised learning

Semi-supervised learning is used to maximize usage of unlabeled data for training and typically uses small amounts of labeled data to create input-output function.

To implement semi-supervised learning, our approach was to preprocess titles like in supervised learning approach, and divide dataset into train and test data. Training data was 80% of dataset and test data 20%. From training data, we labeled around 5%. That labeled data was used to run supervised algorithm to predict other 95% of training data. Afterwards, we combined of labeled data and the predictions from the first run of supervised algorithm together, and used it as training data for another supervised learning.

We concentrated on stackexchange.com data.our algorithm never predicted those categories. We finally decided to choose first 4% of data instead, since all categories were included there and our results improved.

In pipeline for the first supervised learning run, we used LinearSVC classifier instead of Multinomial Naïve Bayes.

4.3 Unsupervised learning

Unsupervised approach does not use labeled data. Instead, it creates clusters based on similarity between data itself. Therefore, we did not need to label the data, just to preprocess the titles like mentioned before.

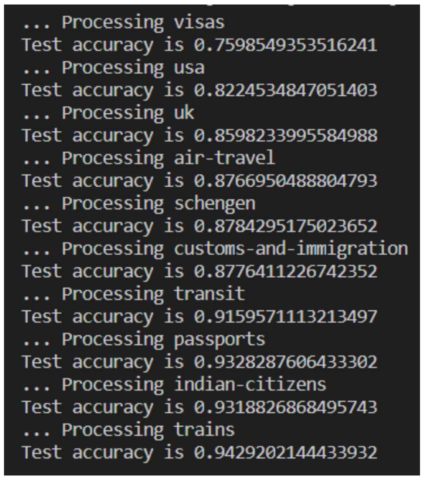
Afterwards, we again used TfidfVectorizer to create TF-IDF matrix (while also removing stop words). To create this matrix, we used 80% of data (17000) as training data. Then we used KMeans to divide this data into 10 clusters (for every tag). We manipulated iteration numbers and after multiple tries, decided to use maximum 400 iterations.

5 PREDICTION PHASE

In this section, we will show the results of training by feeding train data into the created network for each machine learning approach.

5.1 Supervised learning

For each tag, we feed the pipeline with our test data, and compare the prediction with the actual tags of the data for that category. Result for supervised learning are visible on Figure 5.



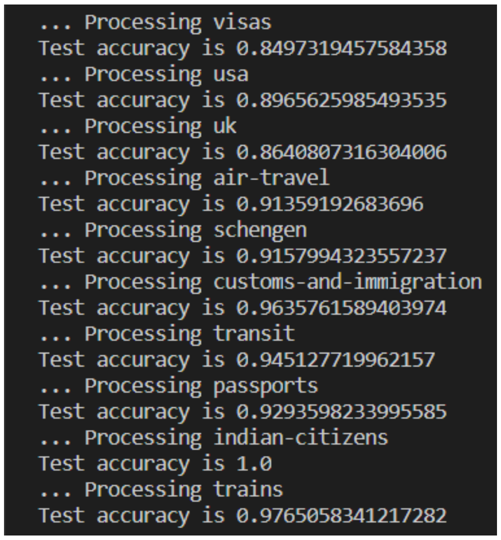
```
... Processing visas
Test accuracy is 0.7598549353516241
... Processing usa
Test accuracy is 0.8224534847051403
... Processing uk
Test accuracy is 0.8598233995584988
... Processing air-travel
Test accuracy is 0.8766950488804793
... Processing schengen
Test accuracy is 0.8784295175023652
... Processing customs-and-immigration
Test accuracy is 0.8776411226742352
... Processing transit
Test accuracy is 0.9159571113213497
... Processing passports
Test accuracy is 0.9328287606433302
... Processing indian-citizens
Test accuracy is 0.9318826868495743
... Processing trains
Test accuracy is 0.9429202144433932
```

Figure 6: Supervised learning test result for each tag

As we can see, supervised learning gives high accuracy scores for each tag. Therefore, we can say that for our dataset, supervised learning seems to be a good approach.

5.2 Semi-supervised learning

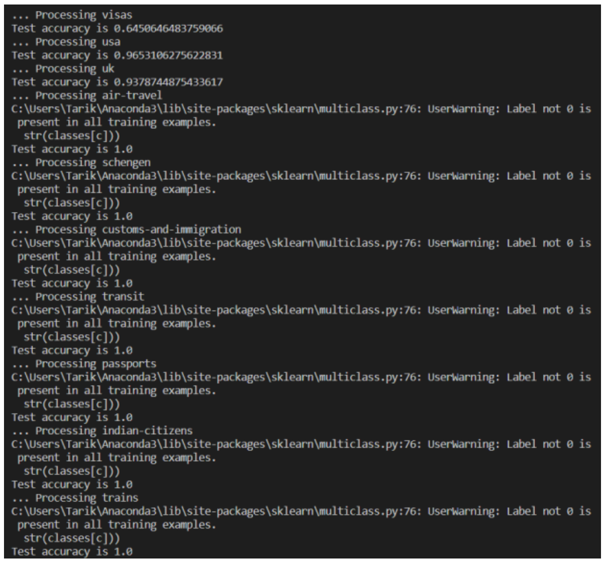
For each tag, we feed data, as in supervised learning approach. When we label first 4% (850) posts, we receive great results from our semi-supervised learning algorithm:



```
... Processing visas
Test accuracy is 0.8497319457584358
... Processing usa
Test accuracy is 0.8965625985493535
... Processing uk
Test accuracy is 0.8640807316304006
... Processing air-travel
Test accuracy is 0.91359192683696
... Processing schengen
Test accuracy is 0.9157994323557237
... Processing customs-and-immigration
Test accuracy is 0.9635761589403974
... Processing transit
Test accuracy is 0.945127719962157
... Processing passports
Test accuracy is 0.9293598233995585
... Processing indian-citizens
Test accuracy is 1.0
... Processing trains
Test accuracy is 0.9765058341217282
```

Figure 7: Semi-supervised learning results with first 4% data labeled

When we tried labeling random 4% of data, many tags were not encountered at all, and therefore were not predicted also. With large enough dataset, this could be avoided.



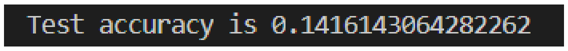
```
... Processing visas
Test accuracy is 0.6450646483759066
... Processing usa
Test accuracy is 0.9653106275622831
... Processing uk
Test accuracy is 0.9378744875433617
... Processing air-travel
C:\Users\Tarik\Anaconda3\lib\site-packages\sklearn\multiclass.py:76: UserWarning: Label not 0 is
 present in all training examples.
  str(classes[c]))
Test accuracy is 1.0
... Processing schengen
C:\Users\Tarik\Anaconda3\lib\site-packages\sklearn\multiclass.py:76: UserWarning: Label not 0 is
 present in all training examples.
  str(classes[c]))
Test accuracy is 1.0
... Processing customs-and-immigration
C:\Users\Tarik\Anaconda3\lib\site-packages\sklearn\multiclass.py:76: UserWarning: Label not 0 is
 present in all training examples.
  str(classes[c]))
Test accuracy is 1.0
... Processing transit
C:\Users\Tarik\Anaconda3\lib\site-packages\sklearn\multiclass.py:76: UserWarning: Label not 0 is
 present in all training examples.
  str(classes[c]))
Test accuracy is 1.0
... Processing passports
C:\Users\Tarik\Anaconda3\lib\site-packages\sklearn\multiclass.py:76: UserWarning: Label not 0 is
 present in all training examples.
  str(classes[c]))
Test accuracy is 1.0
... Processing indian-citizens
C:\Users\Tarik\Anaconda3\lib\site-packages\sklearn\multiclass.py:76: UserWarning: Label not 0 is
 present in all training examples.
  str(classes[c]))
Test accuracy is 1.0
... Processing trains
C:\Users\Tarik\Anaconda3\lib\site-packages\sklearn\multiclass.py:76: UserWarning: Label not 0 is
 present in all training examples.
  str(classes[c]))
Test accuracy is 1.0
```

Figure 8: Semi-supervised learning results with random 4% data labeled

Given the results of our semi-supervised algorithm, we can say that this approach could be used very well to classify this type of data, if given dataset of large enough size.

5.3 Unsupervised learning

After applying KMeans algorithm, we transform test data using TfIdfVectorizer and then use trained model to predict the results. As a result, we get a list of numbers from 0 to 9, which represent cluster (tag) in which our post belongs. We can compare this cluster with tags for that post from XML file. To get the accuracy, we divide the number of same tags with the size of test data and we get figures approximately around what is shown on Figure 9.



```
Test accuracy is 0.1416143064282262
```

Figure 9: Unsupervised learning test results

It is obvious that this accuracy is not comparable with previous approaches, which was expected, because creating TfIdf without labels, from random data is a hard problem. Title itself, without labels does not carry the essence of the post. To improve this type of learning, looking at the body of post should also be applied.

6 CONCLUSION

Using three different approaches on one dataset showed itself to be a challenge. We applied all approaches and showed the results in previous chapters. It is clear that supervised and semi-supervised approaches were by far more successful, because text classification is a complex field and true meaning behind words is hard to be deduced.

It is important to also emphasize that dataset size is a crucial factor. Training quality improves with larger quantities of training dataset, and since our dataset is limited (21138), we think that our results were as expected.

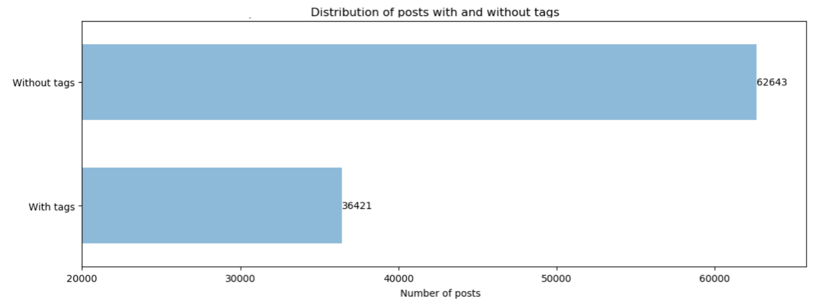


Figure 10: Distribution of posts with and without tags

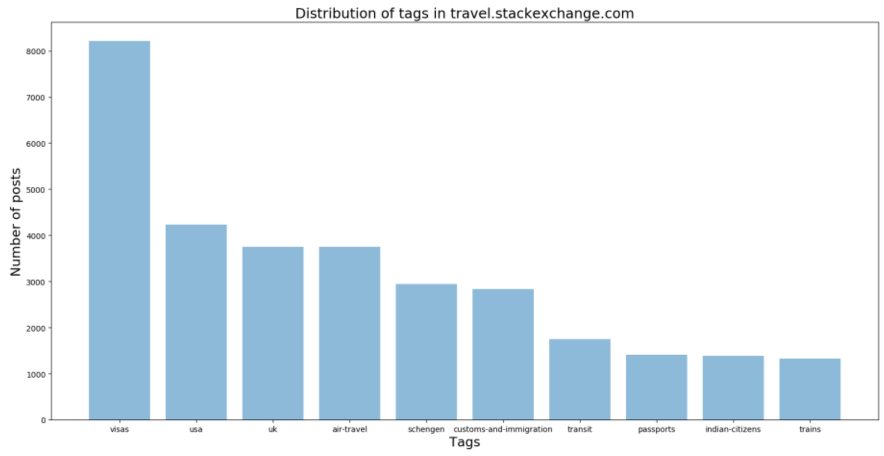


Figure 11: Distribution of tags in travel.stackexchange.com

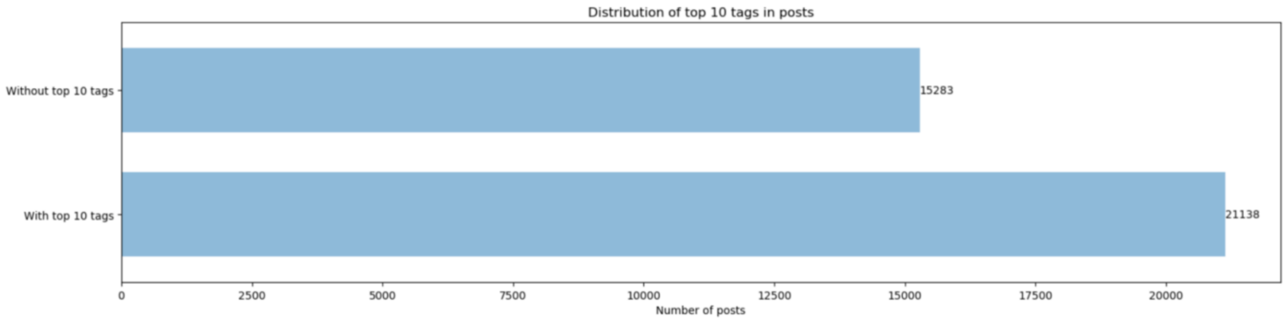


Figure 12: Distribution of top ten tags in posts

REFERENCES

[1] scikit-learn. Retrieved June 20, 2019, https://scikit-learn.org/stable

[2] Towards Data Science. Retrieved June 20, 2019. https://towardsdatascience.com

[3] Stuart J. Russell, Peter Norvig (2010) *Artificial Intelligence: A Modern Approach*, *Third Edition*, Prentice Hall ISBN 9780136042594.