Classification of Reddit Text Posts

Pradeep Kumar Srinivasan
Purdue University
sriniv68@purdue.edu

Mohammad Haseeb
Purdue University
mhaseeb@purdue.edu

ABSTRACT

We plan to work on a large text classification problem, specifically the Reddit Self-Post Classification Task (RSPCT). RSPCT is a dataset with around 1000 classes ("subreddits") with around 1000 examples per class, which is unique because most text classification datasets have sparse labels. We plan to use two traditional machine learning algorithms and one deep learning algorithm to learn to predict the class given the title and body of a post.

Keywords

NLP; large text classification; multi-class classification

1. INTRODUCTION

Text classification with few classes, such as in sentiment analysis, has been well-studied [2] with state-of-the-art techniques like LSTM. However, those techniques do not always work as well in scenarios with many classes [4]. Another issue with training on many-class datasets is that they have a very large number of labels (such as WikiLSHTC-325K dataset [7], which has 325K labels) and are sparse - most labels in the above dataset have less than 100 examples [5].

Reddit Self-Post Classification Task (RSPCT) is a text corpus containing self-posts (i.e., text posts) from Reddit. RSPCT was collected to help spur research on models that could tackle a large number of classes by ensuring a large population of examples for each class (currently, around 1000 examples for each class) and a large number of classes (around 1000). The aim is to allow for a situation comparable to that in the computer vision community, which was helped by the famous ILSRVC competition (ImageNet [10]) with 1000 classes and around 1400 examples per class. This potential is what made us find this project interesting.

We think this is a useful problem to solve because it has not been adequately studied, to the best of our knowledge, and because this could pave the way to future work on such many-class datasets. We hope to try both traditional and deep learning models so that we get a good sense of their relative performance on such datasets.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 ACM. ISBN 123-4567-24-567/08/06…$15.00
DOI: 10.475/123_4

2. PLANS FOR EXPLORATORY ANALYSIS

In order to explore the data and get a feel of it, we intend on using different methods. We will be plotting several histograms to get to know the median length of the subreddits in terms of both words and characters in the subreddits. We hope to get a deeper understanding of the lengths of the subreddits in general using this method. Because the dataset contains thousands of different words and each word is a possible feature, we will be using t-Distributed Stochastic Neighbor Embedding (t-SNE)[11] for visualizing this high-dimensional dataset.

As part of pre-processing, we will remove stopwords and any other undesirable tokens from the dataset.

We will compute sentiment ratings for different posts to see if there are any meaningful correlations across subreddits. If so, we will add that as a new feature to our dataset. We want to use TextBlob[1], or a better alternative if we find one, to get the median sentiment value of each subreddit. We will then be able to use this to determine which subreddit a new post can belong to given its sentiment value. Another potentially useful feature might be the readability level. We plan to compute the readability of the text [8] and check for meaningful correlations.

3. ALGORITHMS WE AIM TO IMPLEMENT

We aim to implement two traditional classification algorithms and one deep learning algorithm.

First, we plan to use Naive Bayes classification because of its simplicity and past track record in multi-class text classification [9]. This will provide us with a good baseline against which to test the other models. Just in case, we will also use a trivial classifier (one that predicts classes at random) as another baseline.

Second, we plan to use logistic regression with a bag-of-words model based on word n-grams and character n-grams [3]. Since we are not sure if logistic regression or SVM will be better suited to this problem, we plan to answer this question using our literature review. If prior research suggests SVM performs better on large text classification, we may switch to it.

Lastly, we plan to use Convolutional Neural Networks (CNN) to test whether deep learning models can outperform the traditional models on text classification problems with many classes. Again, we will check if the existing literature recommends other deep learning algorithms for text

classification and may switch to something like RNN.

4. PROPOSED EXPERIMENTS

On top of the previous three algorithms, we will experiment with adding a new feature - the sentiment of the text (which we will compute using a pre-trained, off-the-shelf model [1]) - to see if it improves performance.

5. EVALUATION OF OUR PROJECT

The key metric for the models will be the Precision-at-K metric [6], since this is a problem with a very large number of classes and we don't expect the correct label to be predicted as the top option. We will measure the precision on the top-1, top-3, and top-5 labels. As mentioned before, we will also experiment with adding a new feature for sentiment rating of the text to see if that improves performance. To understand the impact of the title on classification, we will measure performance by training and testing on just the titles of the posts, just the bodies of the posts, and then with both title and body.

In addition, for the deep learning model, we will plot the loss and accuracy curves vs number of epochs. For the traditional models, we will plot the learning curve vs training set size.

6. TIMELINE

Week 1: We will do exploratory data analysis and study the literature to learn the strengths and weaknesses of past approaches towards large text classification.

Week 2 and Week 3: We will implement and train the different models.

Week 4: We will tune our hyperparameters to get the best performance on the validation sets. We hope to fine-tune variables like learning rate, batch size, and number of epochs.

Week 5: We will spend this week writing our report and presentation (and finishing any work that is left from the previous weeks).

7. CONCLUSIONS

In conclusion, we think this would be a interesting attack on the problem of large text classification with many classes. The fact that we are covering both traditional and deep learning models will help us judge which models are likely to work well on these kinds of datasets. The models we train could find use in judging posts that are off-topic to a particular subreddit and might suggest more suitable subreddits. With more data, we could transfer this approach to classifying news articles by subgenres or blog posts by topic.

8. REFERENCES

[1] TextBlob: Simplified Text Processing. https://textblob.readthedocs.io/en/dev/.

[2] A. Agarwa. Sentiment analysis of twitter data. *Association for Computational Linguistics*, 2011.

[3] A. Besbes. Overview and benchmark of traditional and deep learning models in text classification. https://www.kdnuggets.com/2018/07/overview-benchmark-deep-learning-models-text-classification.html, 2018.

[4] R. Ghani. Combining labeled and unlabeled data for multiclass text. 2001.

[5] M. S. Jones. The reddit self-post classification task (rspct): a highly multiclass dataset for text classification (preprint).

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] I. Partalas, A. Kosmopoulos, N. Baskiotis, T. Artieres, G. Paliouras, E. Gaussier, I. Androutsopoulos, M.-R. Amini, and P. Galinari. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581*, 2015.

[8] G. C. Pastor. Translation universals: do they exist? A corpus-based NLP study of convergence and simplification. https://www.researchgate.net/profile/Viktor_Pekar/publication/254434062_Translation_universals_do_they_exist_A_corpus-based_NLP_study_of_convergence_and_simplification/links/0046353b6bfcaa134b000000/Translation-universals-do-they-exist-A-corpus-based-NLP-study-of-c pdf, 2014.

[9] J. D. Rennie. Improving multi-class text classification with naive bayes. 2001.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F.-F. Li. Imagenet large scale visual recognition challenge. *CoRR abs/1409.0575*, 2014.

[11] L. van der Maaten. Visualizing Data using t-SNE. https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf, 2008.