Wikipedia Articles Topic Dataset

Vladislav Gusev

Dec 2021

Abstract

This report contains details about building topic classification dataset from Wikipedia top level section names and related text. Also, few common NLP models was build to check they perfomance on this dataset.

1 Introduction

In this work we take new approach to use Wikipedia data to produce automaticaly labeled dataset for topic classification task. It important because provide low human labor apporach to building topic classification datasets.

1.1 Team

Vladislav Gusev

2 Related Work

Wikipedia a data source for the variety of data science models and tasks. For example it used to train BERT [Devlin et al., 2019] and RoBERTa [Liu et al., 2019]. BERT was pretrained on two tasks: language modelling (15% of tokens were masked and BERT was trained to predict them from context) and next sentence prediction (BERT was trained to predict if a chosen next sentence was probable or not given the first sentence). Also, Wikipedia discussions used in toxic classification dataset [Thain, 2017]. The Stanford Question Answering Dataset (SQuAD) [Rajpurkar et al., 2016], a reading comprehension dataset consisting of 100000+ questions posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage.

3 Dataset

Wikipedia articles contain few levels of headers, level 1 - article name, level 2 - top level article section headers, level 3 - subsectons. Each section has some number of paragraphs related to section name. Most section headers have standardized names, like "Geography", "History", "Location", etc. At this work, top level 2 section names was used as topic labels, and section text as topic content. As source of data, english Wikipedia xml dump was used, this dump produced regulary and can be downloader from `https://dumps.wikimedia.org/enwiki/`. Then WikiExtractor utility was used, to extract artilces text from dump, it was slightly patched to output specially marked level 2 section headers. Sections with topic name and it text content was extracted each to separate file, then it cleaned up to delete topics smaller then predefined size (384 bytes). Statistic was caclulated, how much samples we have for each section name, top N selected as topic names, at this work N=65, it gives approximately 8 000 samples for smallest topic, obviously N is number of classes when we use this dataset for classification task. Number of samples per topic heavily imbalanced, to tackle this problem undersampling was used, with number of samples per class approximately 12000, if class has fewer samples than it truncated to actual size. Totally 600858 samples was produced for final dataset. At last step, splitting into train and test part was done, with train fraction 0.85. All scripts and more detailed info how download and reproduce dataset located at GitHub repo `https://github.com/xmvlad/nlp_wiki_topic`

4 Models

To benchmark created dataset three common models was used. Logistic regression with Tf-Idf vectors as baseline, BERT [Devlin et al., 2019] and RoBERTa [Liu et al., 2019]. For BERT and RoBERTa pretrained models used with top classification layer was reinitialized to make fine tuning.

5 Experiments

5.1 Metrics

Two widely known metrics was used accuracy and f1 score, they calculated on top 1 result, it means exact match for most probable predicted class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

5.2 Experiment Setup

Full dataset contains 600858 samples with 65 classes, it was stratified splitted into 0.85 train and 0.15 test part. For BERT and RoBERTa pretrained model with last classification level removed and reinitialized to handle 65 classes, then model was fine tuned with gradients propagated over full model. All models was trained for 4 epochs, then epoch with best result on test dataset was selected, all other hyperparameters was fixed initially and doesn't change. For optimization Adam optimizer was used with following parameters β_1=0.9 and β_2=0.999, learning rate=2e-5, effective batch size=24.

5.3 Baselines

Logistic regression with TF-IDF embedding vectors was used as baseline. Vocabulary size was truncated to most common 5000 words. Experiments with increasing vocabulary size or using stemming for text tokens, doesn't change model perfomance sagnificantly or produce worse results due overfitting.

6 Results

Model	Accuracy	F1 score
LogReg TF-IDF	0.626	0.621
BERT	0.778	0.777
RoBERTa	0.784	0.784

Table 1: Model results

Achieved results Tab. 1. overly consistent with model perfomance on other datasets. TF-IDF logistic regression provide strong baseline because most topics have unique words that distinguish them from each other. Expectedly BERT model sagnificantly improve results over baseline, and RoBERTa improve few percent over BERT. Results consistent over two used metrics: accuracy and f1 score.

7 Conclusion

Novel approach to generate topic classification datasets from Wikipedia was present. Few decent models was fine-tuned to achieve SOTA results for this dataset.

References

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *CoRR*, abs/1907.11692.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

[Thain, 2017] Thain, Nithum, D. L. W. E. (2017). Wikipedia talk labels: Toxicity.