Classification of websites

Junhua Chen – University Of Melbourne

***Abstract*—We employ various models from Natural language processing (NLP) to classify websites into 1 of 10 categories: Arts, History, Geography, Everyday life, Social sciences, Biological and health sciences, Physical Sciences, Technology and engineering as well as Mathematics. The algorithm is also provisioned to detect irrelevant websites.**

I. Introduction

While many websites are easy to classify as non-academic, websites of academic value still frequently form distractions for students. For instance, Wikipedia surfing, news articles as well as irrelevant topic pages pose major distractions for the modern student. The aim of the following article is three-fold; To establish an experimental benchmark for detecting non-academic websites as well as to devise a method for classifying academic websites into categories.

II. Benchmark and task description

A. Informal Task Description

We define 10 different semantic categories:

- History
- Geography
- Arts
- Religion and Philosophy
- Everyday life (note this is counted as irrelevant)
- Social sciences (including commerce)
- Biological and health sciences
- Physical sciences (including chemistry)
- Technology and Engineering
- Mathematical sciences

These categories are labelled 0 to 9 in that order. We need to construct a function $f: Text \rightarrow R^{10}$ that takes a piece of text as input and as output returns a likelihood value for each of the 10 categories. Ideally, this function should be able to detect if a page belongs to none of these categories (e.g a fantasy article) and thus also class it as irrelevant.

B. Training data and Benchmarking methods

The Limitations on time mean that human labelled data was ruled out immediately. Instead, Wikipedia, a common data source of NLP problems was used. In this case, Wikipedia has a collection of 8000 feature-quality articles [1] of total length 500 MB categorised into the categories in part A. These were split in a 85-15 ratio into training and cross validation. Thus, these were scraped and cleaned using the beautiful soup library in python. It was found in testing that models were much more accurate on Wikipedia pages that other articles, and thus a more realistic set of websites were used for benchmarking. 30 random Wikipedia keywords were chosen from each category and fed into google via a python API, and the top 15 results were returned. These websites were scraped too, and despite some having scrape blockers, 1822 websites were successfully obtained. During the scraping process, only <p> tags with more than 50 words were kept.

C. Formal Scoring Process

Many websites my discuss content from multiple disciplines, thus despite all input websites having one label, we cannot merely consider the top prediction per website. Thus, it was decided to consider a website correctly predicted if the correct label was among the top 2 labels predicted. Further to this, we will tune a threshold such that if the confidence level of the intended category(s) is overly low, that the page will be rejected from all categories. A statistical approach will be used for this and the level will be chosen to be the k such that the probability that a correctly predicted website will have an confidence interval below k is at most 0.025:

$$Pr(Conf < k | top2) < 0.025$$

This will evidently reduce the accuracy by ~2.5%, but will not be reported in the results table. As detailed in later sections, this approach is supported by the experimental observation that the conditional random variable $Conf|top2$ is often lognormally or normally distributed.

D. Linguistic Intuition

The problem can best be described linguistically as classifying the semantic field of a text. [2] Semantic fields are often categorised as groups of words with lots of sense relations between them. Thus, the problem is best looked at from a lexical perspective, and so sequence models and associated models like LSTMs were immediately discounted.

III. Models Used

Due to the limitations of time, only two representations of data were tried: The Glove50 [3] word embedding and the ubiquitous Bag of Words model.

A. Bag of Words (BOW) Based Models

The models tried were very standard, namely Naïve Bayes, Logstic Regression as well as Linear kernel SVMs. Naïve Bayes (Laplace smoothed) was coded from scratch, while Linear Regression employed the Pytorch library and SVMs utilised sklearn. Only SVMs required hyperparameter tuning, of which C=30 and 2000 iterations was found to suffice. To improve the accuracy of the model, A Porter Stemmer from the NLTK library was used to stem all the words. Further, common stop words such as "a, an, the" were removed as well as 1 letter words. Due to the need to allow for a text to belong to multiple classes, sigmoid activation was used in lieu of the more common softmax.

B. Embedding based models

Glove embeddings were trained on an extremely large corpus of Wikipedia articles, and is trained such that the cosine similarity of two articles would represent their semantic similarity. This makes it an attractive tool for semantic classification. The first model is a standard approach. The word embeddings for all words in the text were uniformly averaged, then fed into a 10-way classifier, of which linear regression and a 3-layer neural network was tried. The second was far more complicated, and was based on a paper [4], and involved 10 separate binary (yes/no) classifiers for each class, with each of the 10 classifiers was trained using an undersampling method: Every positive example for each domain was sampled along with the same sized random sample of negative examples. The input to each binary classifier was a Naïve-Bayes log probability weighted averaged embedding as described in the paper.

Regrettably, there was not enough time to pursue ideas surrounding arranging binary classifiers in the shape of decision trees or gradient boosting or to try SVMS with the Naïve Bayes averaging. As with previously, Stop Words were removed and sigmoid activation was used for training, although a coding error which could not be rectified in time meant they weren't removed for the NB-weighting method.

C. Sentence Voting

A new approach as an accuracy booster for aforementioned models. With the intuition that individual sentences will cover individual semantic fields and thus that the frequency of sentences on a particular semantic field will determine the overall semantic field of a text, we employ 40-word sliding windows (with 30 word shifts) on the text, with a classifier determining the field of each window. Each window then votes on a single topic. However, windows with maximum confidence levels lower than the 95% threshold determined for each classifier are thrown away as irrelevant. These parameters were determined on the whim based on the average length of an academic English sentence according to a single Quora answer [5], and a small overlap was chosen as intuitively having the end of the previous sentence as context felt correct.

IV. Results And Analysis

In all tables in this section Test set accuracy refers to the top-2 accuracy of the returned confidence values.

A. Bag Of Words Based Models

Model	*Wikipedia Validation*	*Testset*
Naïve Bayes	*51%*	*740/1820 (40.7%)*
Logstic regression	*96%*	*1376/1822 (75.5%)*
SVMs	*93%*	*1380/1822 (75.6%)*

It is clear from this table that Models perform much better on wikipedia articles than articles in the wild. It is suspected that this is less due to overfitting, than it is to wikipedia articles being much more substantial and well behaved. SVMs and Logistic regression also had top3 accuracies exceeding 85%.

B. Bag of Words Models

Model	Wikipedia validation	Testset
Naïve averaging, Logistic regression	85%	1207/1822 (66.2%)
Naïve Averaging, 3 Layer network (50-20-10 architecture)	92%	922/1822 (50.6%)
Naive-Bayes weighted Logistic regression (10-way 1 vs all)	Not tested	523/1822 (28.7%)

The accuracy of the supposedly smart-averaging model was extremely poor, and from examining the activations of each Logistic regression, it seems that despite extremely high training accuracy (high 90s), the activations of each of the 10 binary classifiers were consistently low (<0.3). This suggests overfitting. Similarly, the 3-layer network appears to overfits too. However, a lack of time meant that regularization measures could not be undertaken. While it should be noted that the performance of all three models is worse than the simpler models, the naïve averaging method is still very useful. Despite being worse in the top2 category, this model has the property that its neuron activation is usually (~80% of time) not too far below the top (less than 0.1) and has high activation.

C. Sentence Voting

Three models were selected for the sentence averaging approach: Linear regression, Naïve embedding linear regression and Bag of Words SVMs:

Model	95% activation threshold	Boosted accuracy on testset
Naïve Embedding LogisticRegression	Not calculated, but 0.3 was used	1365/1822 (74.9%)
Logistic regression on BOW	**-2**	**1481/1822 (81.3%)**
Linear BOW SVM	-0.75	1378/1822 (75.5%)

It is clear that sentence voting provides a marked improvement over text-global models, with the exception of the BOW SVM for unknown reasons. While time was not available to optimise SVMs, the coefficients of each word could be thought of as a mini size 10 embedding, and averaging these embeddings provide a very fast method of evaluating logistic regression. As such, while both logistic regression tests ran in under a minute, the SVM took more than an hour to benchmark.

V. Model Analysis

Seaborne was used to analyse the mistakes of each model:

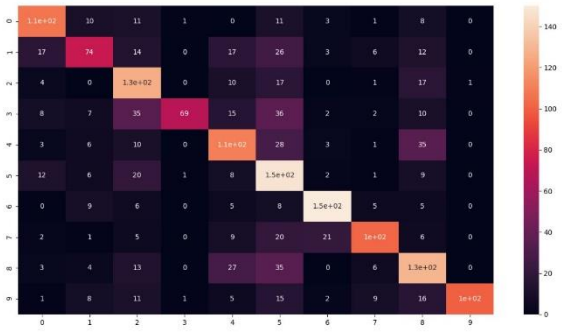

Figure 1: Confusion Matrix for sentence voting BOW Logistic regression

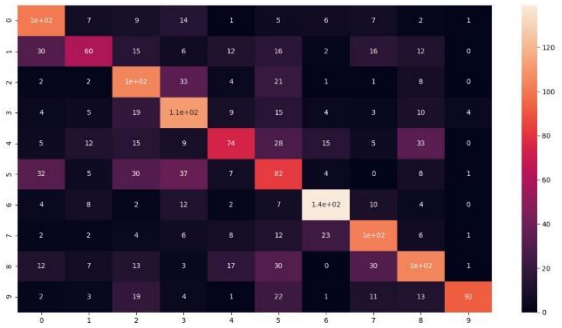

Figure 2: Confusion Matrix for sentence averaged Embedding regression

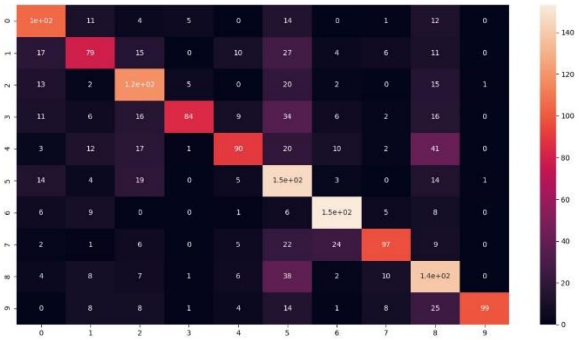

Figure 3: Confusion matrix for BOW SVM

The confusion matrices indicate mistakes in the top-1 predictions. From this diagram the main observation is that the main points of confusion are in the social sciences category, as well as technology with everyday life. It is believed that this is unavoidable due to the large amount of overlap between the topics. Despite this, each of the Sentence sliding window models {BOW SVM, Glove50 regression, logistic BOW regression} have performed well, with the best model having a 81.3% accuracy. Seaborne plots indicate that the distribution of output weights for logistic regression (before sigmoid applied) were approximately normal with $\mu = 1.458, \sigma = 1.758$, so a Z level of $Z = 1.96$ ($i.e.$ ~97.5%) was chosen to get a threshold of -2 for irrelevance cutoff for the text.

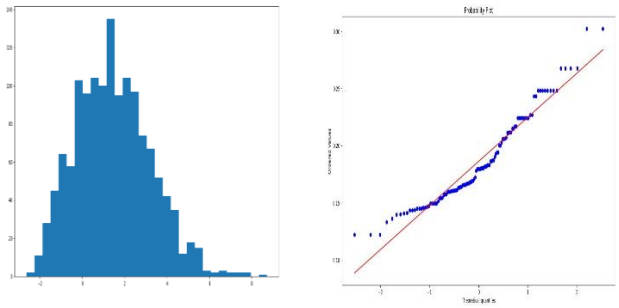

Figure 4: Distribution of confidence thresholds of model on correct category. It roughly obeys the 68-95-99.7 rule and the QQplot shows good linearity with the normal distribution.

Conclusions

A viable model with top2 accuracy exceeding 81% is developed on the benchmark test set. Further, the normally distributed properties of the confidence values enabled an irrelevance cutoff of -2 to be chosen.

Acknowledgment

The Author is grateful for productive discussions with Jerry Mao about pytorch implementations and undersampling as well as Bowen Feng for rubber ducking my code.

References

[1] Vital articles, Wikipedia https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/4

[2] Stanford University CS224 Course materials 2019 https://web.stanford.edu/class/cs224n/readings/cs224n-2019-notes01-wordvecs1.pdf

[3] Mikolov et al. 2013 https://arxiv.org/pdf/1301.3781.pdf

[4] Elsaadawy, Abdallah & Torki, Marwan & El-Makky, Nagwa. (2018). A Text Classifier Using Weighted Average Word Embedding. 10.1109/JEC-ECC.2018.8679539.

[5] Number of Words in common sentence, Quora, https://www.quora.com/What-is-the-average-number-of-words-per-sentence-in-common-writings-such-as-news-articles-and-college-essays