Topical Web-page Classification of the DMOZ Dataset*

Kahlil Oppenheimer
Brandeis University
415 South Street
Waltham, MA 02453, USA
kahlil@brandeis.edu

Abstract

Multi-class topical web-page classification is a difficult task with widespread application. Throughout this paper, I analyze the performance of well-studied techniques on two different representations of web-pages: hand-written meta-descriptions and on-page text content. I acquired all of the training labels and website descriptions from the DMOZ dataset and all of the on-page content from scraping the actual web-pages. I achieved 74.035% and 79.121% accuracy for on-page content and website descriptions respectively in a 16-way classification task with a 42.032% most frequently tagged baseline accuracy.

1 Introduction

Topical Web-page classification is the task of taking a Web-page and determining what the topic of the Web-page is. For example, a good topical Web-page classifier would be able to take in an ESPN sports review article and classify that web-page topic as sports, or an article about NASA's latest discovery as science.

When approaching such a classification task, you must decide how granular you wish your classification to be. Should Japanese and Korean anime belong to separate categories or should they both belong to the anime category? The more granular your categories are, the more specifically tailored they can be. However, in the extreme case, if every document belonged to its own category, then classification would be a vacuous task.

Instead, we generally want to select categories that highlight the most important differences between documents and that contain a large enough group of documents for meaningful analysis.

The categories I chose are the top levels of the DMOZ directory (arts, business, computers, games, health, home, news, recreation, reference, regional, shopping, sports, science, society, and world).

Topical web-page classification can be challenging because web-pages have a lot of extra noise that traditional text documents dont have. There is a lot of information floating around, such as script and style tags, links to other Web-pages, images, advertisements, etc. Much of this information may turn out to be extremely relevant to topical classification, while much of this information may turn out to be completely extraneous. Sifting through all of this information to determine how to accurately represent a Web-page is non-trivial.

But topical Web-page classification has nice potential benefits. Tools could be built that leveraged the topic of the current web-page. An example would be a browser extension that suggests similar web-pages to the user based on the topic of the web-page they are currently viewing.

There has been a decent amount of research previously in this topic. Many of the techniques I analyze are inspired by the works of (Qi and Davidson, 2009) and (Shen et al., 04). In particular, I experiment with techniques of information-gain and n-gram analysis described in the former, and Luhn and lexical semantic analysis document summarization

*The techniques analyzed in this paper are inspired by the work of previous researchers as well as the teachings of Benjamin Wellner and Te Rutherford.

described in the latter.

I ran several experiments over several different techniques, each of which is described in further detail in Section 3. A brief overview of each is as follows:

1. comparing types of classifiers
2. comparing the results of plain bag-of-words for descriptions versus on-page text
3. observing how different classifier types scale with sample size for both descriptions and on-page text
4. observing how accuracy scales with the number of features in the resultant vector from information-gain
5. comparing n-gram analysis against simple bag-of-words
6. measuring the performance of various document summarization algorithms.

My main goal in this paper is to analyze the effects of the different techniques outlined in existing literature on both description-based and on-page text representations of Web-pages.

Paper layout. The rest of the paper is as follows: Section 2 covers the details of handling the actual dataset, including document featurization. Section 3 covers all experiments and their results. Section 4 concludes the paper.

2 The DMOZ Dataset

For my analysis, I used the directory.mozilla.org dataset (https://www.dmoz.org, 2015), which describes itself as the "largest, most comprehensive human-edited directory of the Web." It was formerly known as the Open Directory Project (ODP), but is now commonly referred to as DMOZ. It contains over 3,600,000 web-pages annotated with short descriptions about the contents of the page, along with fine-grained classification of the topic of the Web-page. I truncated all of the topics to the top level distinctions (i.e. Arts/Anime/Japan became Arts and Science/Space/NASA became science). The dataset does not contain the actual content of the web-pages, which makes sense as these tend to frequently change. However, this meant that I had to scrape the actual content of the Web-pages, which can take a long time. Because of this, I only worked with samples as big as 50,000 Web-pages. Even just 50,000 Web-pages took over 24 hours to scrape because many are hosted on older servers with latency in upwards of 5 seconds per page. To strip the tags and non-text elements from the page, I used an open source Python web scraping framework (http://www.crummy.com/software/BeautifulSoup/bs4/doc/, 2015).

The dataset is distributed as a resource description framework (RDF) file, which is basically just a fancy XML document. I used an open-source parser (Kramr, 2015) to convert this to JSON data. Then, I randomly sampled the desired amount of documents from the entire dataset for each sample size I wanted to analyze. Finally, I scraped the actual web content of those pages and store that as well. All of my website-description analysis used the descriptions in the DMOZ dataset, while the on-page text analysis used the actual scraped source of the Web-pages.

2.1 Document features

Unless explicitly stated otherwise, the features used for each experiment were simple bag-of-words representations of the descriptions, or simple bag-of-words representations of the plain text of the Web-pages (with the scripts, tags, and meta-data stripped away). Common stop-words were filtered all text was put into lower case for all experiments. While seemingly simple, bag-of-words featurization has proven effective in other natural language processing sub-domains and has proved effective in topical web-page classification as well.

3 Experiments and results

All experiments were run using Mallet (http://mallet.cs.umass.edu, 2015), an open source natural language machine learning toolkit. All experiments were evaluated using 10-fold cross validation. The most-frequent-tag baseline for all experiments ranged between 44-46%, depending on the sample. Because the task being evaluated was a 16-way classification, overall accuracy (rather than 16 precision, recall, and f-measure scores) is the metric for each experiment. The overall best accu-

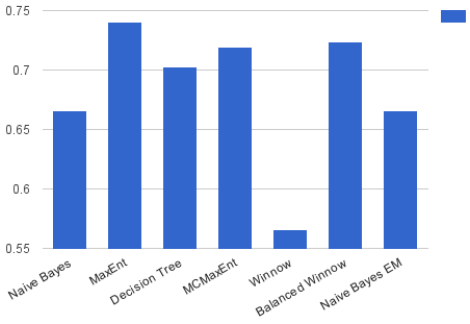

Figure 1: Classifier type vs accuracy

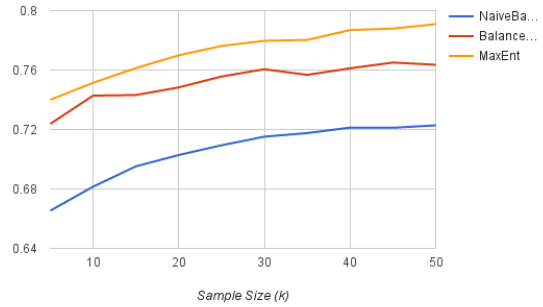

Figure 3: Sample size vs classifier type (description)

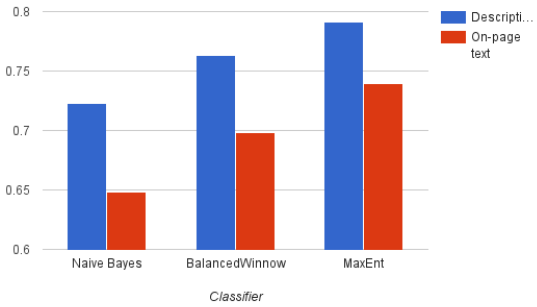

Figure 2: Description vs on-page text

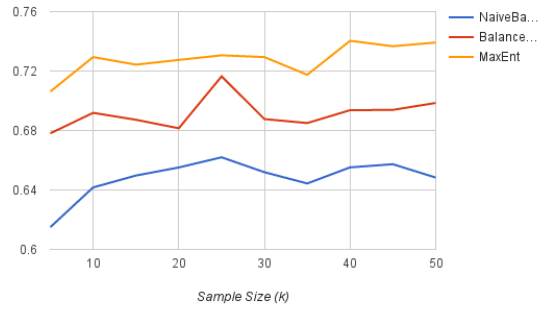

Figure 4: Sample size vs classifier type (on-page text)

racy achieved across all experiments was 79.121% for description-based analysis and 74.035% for on-page text-based analysis. Additionally, I included the results of a Naive Bayes classifier in all results as a baseline to compare against.

3.1 Classifier type vs accuracy

The first experiment determined which classifiers were to be used for the remaining experiments. I took 5,000 random samples of full Web-pages and ran all of the classifiers that Mallet supported and measured their accuracy. The results show maximum entropy classification with a clear lead, but balanced winnow classification in a close second place. The balanced winnow classifier also has the advantage that it takes a fraction of the time as maximum entropy to train. Naive Bayes performed close to the worst among the bunch.

3.2 Description vs on-page text

I ran the top classifiers determined from the previous experiment across 50,000 samples of both web-page descriptions and simple bag-of-word representations of the on-page text of the web-pages. The tests yielded 79.089% accuracy for descriptions and 73.919% accuracy for on-page text with a maximum entropy classifier.

3.3 Sample size vs classifier type

I analyzed how the different classifier types accuracies scaled with sample size for this particular dataset for both descriptions and on-page text. The description accuracy appears to increase logarithmically with the number of samples. This goes to say that more samples helps, with diminishing returns. On-page text, however, shows results that are harder to interpret. There is certainly a clear positive increase in accuracy when moving from 5,000 documents to 25,000 documents. But after the 25,000 document threshold, the accuracy appears to de-

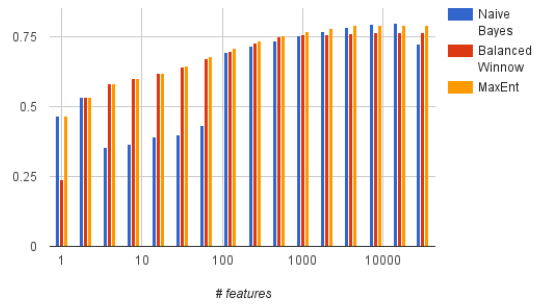

Figure 5: Top-n features vs accuracy (description)

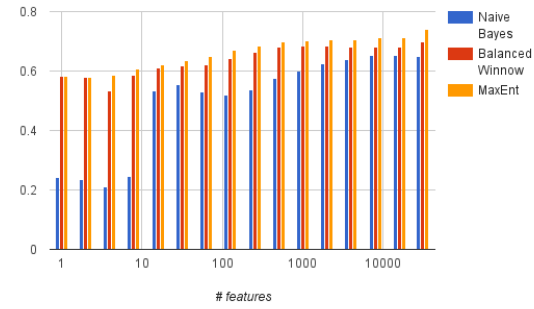

Figure 6: Top-n features vs accuracy (on-page text)

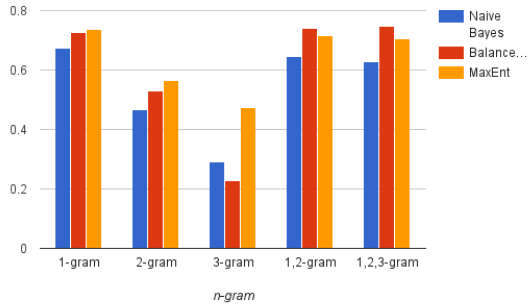

Figure 7: N-gram features vs accuracy (description)

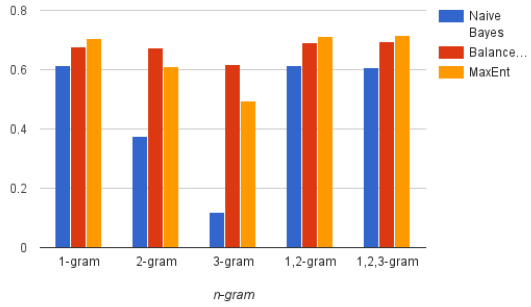

Figure 8: N-gram features vs accuracy (on-page text)

crease then slowly increase again. Without also analyzing larger samples, we cannot conclude too much from this.

3.4 Information Gain

This next experiment exploits Mallet's built in support for information-gain feature pruning. I ran these experiments on 50,000 randomly selected documents for both description and on-page text. The experiment measures how accuracy scales with the target number of features preserved in information-gain. Both descriptions and on-page text show that near-top accuracy can be achieved from preserving just 1,000 features. Both also show that Naive Bayes dramatically suffers from lower feature dimensionality, but catches up to maximum entropy and balanced winnow after around 100 features or so. The description results seem to indicate that accuracy actually decreases after more than 16,000 features or so. In fact, the highest description classification accuracy across all experiments (79.121%) was achieved during this experiment with 16,384 target features.

3.5 N-gram featurization

For n-gram analysis, I experimented with unigrams, bigrams, and trigrams on a sample of 5,000 randomly selected documents. I tried each individually, as well as cumulatively (i.e. unigrams + bigrams and unigrams + bigrams + trigrams). No result performed higher than plain unigrams, though some performed equally. I tried running the results that contained unigrams, bigrams, and trigrams through the information-gain algorithms described in Section 3.4 because I assumed they would have at least as rich a feature space, if not more, than simple unigrams. However, accuracy was lower than from applying information-gain to plain unigrams. Thus, I determined that n-gram analysis was not useful for this particular classification task.

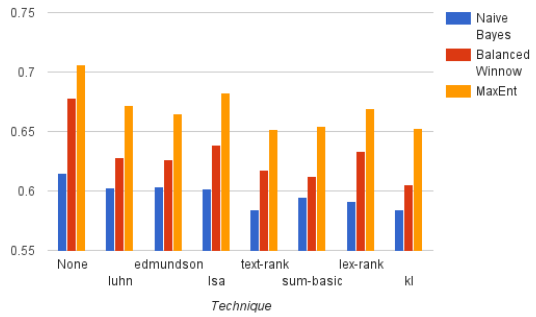

Figure 9: Summarization techniques vs accuracy

3.6 Document summarization techniques

My final experiment was to try out different document summarization algorithms on the on-page text of web-pages to see if I could approximate the performance of descriptions. I used an open-source document summarization toolkit (Belica, 2015). This toolkit offered 7 different document summarization techniques, so I tested each. I ran each summarization on a single set of 10,000 randomly selected on-page text representations of documents, then trained and evaluated classifiers on this new corpus of summarized documents. I used summaries that were 20% of the original document lengths. The results show that Luhn, Edmundson, and lexical semantic analysis perform the highest, which is consistent with (Shen et al., 04). Training time was immensely reduced, but I did not gain any accuracy from using these summaries, as (04) suggests. This is likely because I used vanilla text summarization algorithms, while (04) modified the algorithms specifically for web-pages.

4 Conclusion

I found that a maximum entropy classifier achieved the highest classification accuracy, while a balanced winnow classifier achieved almost-as-good results in a fraction of the training time. Description classification accuracy outperformed on-page text accuracy by at least 5% in all cases, signifying the value in generating or approximating Web-page descriptions, through summarization techniques or otherwise. Increasing sample size increased accuracy, but with diminishing returns, and training time can easily climb from minutes to hours. N-gram analysis resulted in accuracy less than or equal to plain unigrams, but dramatically increased the feature count and training time. Information gain resulted in accuracy less than or equal to full feature vectors, but cut down training time by several orders of magnitudes. Finally, summarization techniques resulted in lower accuracy, but cut down on training time by upwards of 80%.

A big takeaway from my analysis is that the results stated in prior research are not always easily reproducible, and not all techniques will be as performant as stated (at least out of the box). I did learn, however, that certain techniques like information gain and summarization can dramatically decrease training time while maintaining near-equivalent accuracy. I also learned that certain classifiers will have higher accuracy but will take dramatically longer to train. I suppose there is no free lunch in natural language processing.

Acknowledgments

All of my work is inspired by the work of the cited authors, my professor Benjamin Wellner, and my teaching assistant Te Rutherford.

References

Mio Belica. 2015. https://github.com/miso-belica/sumy. *Github*.

http://mallet.cs.umass.edu. 2015. Mallet: machine learning for language toolkit.

https://www.dmoz.org. 2015. https://www.dmoz.org.

http://www.crummy.com/software/BeautifulSoup/bs4/doc/. 2015. Beautiful soup.

Tom Kramr. 2015. https://github.com/kremso/dmoz-parser. *Github*.

Xiaguong Qi and Brian D. Davidson. 2009. Web page classification: Features and algorithms. *ACM Computing Surveys*, 41(2).

Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zhang, Benyu Zheng, Yuchang Lu, and Wei-Ying Ma. 04. Web-page classification through summarization. *SIGIR*.