Classifying Blogs Using NLP: Challenges and Pitfalls

Hong Qu, Andrea La Pietra, Sarah Poon

University of California at Berkeley
School of Information Management & Systems
314 South Hall, Berkeley, CA, USA 94720-4600
{hqu, lapietra, sspoon}@sims.berkeley.edu

Abstract

Blogs are difficult to categorize by humans and by machines alike. In the early days of the web, directories maintain by human labor could not keep up the growth of the web; likewise, blog directories cannot keep up with the explosive growth of blogs. Yet machines learning methods cannot accurately categorize blogs because blogs are written in an informal, capricious style. We design a natural language processing (NLP) experiment for categorizing one hundred and twenty blogs into four topic groups: personal diary, politics, news, and sports. The baseline measure is term document frequency. For comparison, we extract two linguistic features: the title of blog entries and the anchor text from inbound links. While these features perform worst than the baseline, they illustrate the challenges involved in designing a blog classification algorithm. We analyze the corpus, features, and result data to formulate a more robust system and explore the benefits and limitations of NLP in organizing blog content.

1. Introduction

The number of blogs in the blogsphere is growing at an exponential rate. The Pew Institute found that 2-7% of Internet users have a blog and that 11% read blogs [1]. Technorati’s web crawlers indicate that there are about 12,000 new weblogs created each day; put another way, a new weblog is created every 7.4 seconds [2]. Given the explosive popularity of blogs, it would be useful if we could devise a natural language processing (NLP) system to automatically classify blogs. It is difficult to group blogs into categories because of the freestyle nature of the discourse. Bloggers write whatever is on their mind, sometimes inventing new vocabulary and grammar. Some blog intentionally deviate from rules of language and decorum to create a spectacle for the sake of attracting an audience [3].

While standard statistical measures such as term document frequency perform reasonably well in classifying blogs, we posit that weighing specific linguistic features such as the title of individual blog entries and the anchor text from incoming links will make the classification algorithm even more accurate. Our hypothesis is that these features can better differentiate between blog categories. In practice, however, blogs do not fit neatly in one category: they can fall in multiple categories because bloggers write about whatever topic strikes their fancy. Nonetheless, we experiment with a small corpus of blogs using NLP techniques to explore the feasibility of automatic classification.

2. Previous Work

The practice of blogging is in its nascent stages. There are many definitions for what constitutes a blog. For our purposes blogs are: “Webpages that are constantly updated with new commentary and links about a particular topic. Often very personal [4].” It may be too early to try to fit blogs into genres or topics because “our collective conceptions of weblogs are changing too quickly to realistically capture them in such frameworks[5].” One farsighted researcher proposed a classification system along two dimensions: personal vs. topical, and individual vs. community [6].

Yahoo started out by creating a directory of websites by manual labor. But the growth of the web soon overwhelmed human indexers. Online blog directories such as Blogarama and Blogexplosion are taking this same approach. We believe that the growth of blogs will require automation. Moreover, these services classify over topics but ignore other dimensions such as genre. The challenges of automatic classification of blogs are enormous. As a modest beginning, we focus on classifying blogs into four topics: personal diary, politics, news, and sports.

3. Feature Selection

Picking good linguistic features to classify blogs is challenging, because blog content is so diverse and convoluted. We came up with two features: the title of each post and the anchor text over inbound links to the blog. We ran variations on both features to find ways to improve them. The baseline we tested against is unigrams, or individual word tokens in the corpus minus stop words such as the, a, and of.

The first feature is the title text of every blog post. It summarizes the post’s content. We predict that the words in
the title co-occur within a particular category but do not co-occur in other categories. We weigh the words in the title ten times greater than the rest of the corpus. That did not boost accuracy. Then we used a sentence splitter and weighed the first sentence of the blog entry 10 times. The second linguistic feature is inbound links. We believe that the text over inbound links to the blog describes the content within the blog.

4. Preparing the Corpus

First, we limit the scope of topics to four categories (personal diary, politics, news, and sports) and manually harvested 30 blogs for each category. Next, we parse the RSS feeds from these blogs. We used Magpie [7] an open source PHP tool, to extract the title and body text of the blog. We used NLTK [8] to process the text.

Using NLTK we tokenized and annotated the blog entry titles. We obtained the other feature, anchor text from inlinks, by querying Technorati’s Cosmos search [10]. To get incoming anchor text, a script sent queries to Technorati searching for incoming links to the blogs and save the results into text files. We then scraped the Technorati results to extract the text around the incoming links. After gathering the blog text and the anchor text, we normalize the corpus and ran various classification algorithms using Weka [9].

5. Evaluation

The results indicate that both linguistic features performed worst than the baseline. We used the Naïve Bayes Multinomial classification algorithm. We found that baseline (unigram tokens) classified blogs with the highest accuracy [Appendix A]. The reason the title and the anchor text under-performed may be due to the small corpus. We only had one hundred and twenty blogs, some of which might have been categorized incorrectly by the human annotator.

The most unexpected finding is that anchor text decreased the accuracy. We ran the training dataset and the testing dataset using the baseline unigram feature and discovered a very peculiar result. All the blogs hand-labeled as news blogs were incorrectly classified as political blogs [Appendix B].

The classifier misclassified news blogs as political blogs multiple times. In the test set, all the blogs were classified correctly except for the 5 news blogs. Even in the training set, 12 news blogs were correctly classified as news, but 7 were incorrectly classified as political. This result makes sense because the category news blogs is very difficult to pin down. News blogs often talk about political news; even human classifiers would have trouble reaching a consensus on whether a blog that talks about politics is a political or a news blog. We believe that the source of poor performance stem from a number of flaws in the experimental design. First, the topics boundaries are blurry. Second, the corpus is too small. These shortcomings distort the effect of the two features in the classification algorithm.

6. Discussion

Blogs are inherently difficult to group into categories. There are no definitive taxonomy that satisfies the judgment of the blogger, the classifier, and the end user. We investigated feasibility of applying NLP to prototype a blog classifier. The major stumbling block lies in defining the semantic topics for grouping blogs. At first glance, news would be a good candidate as a semantic topic, but it turns out that news blogs contain mostly political commentary. Nonetheless, the findings suggest that machine learning algorithms performed reasonably well (accuracy of about 80 percent) even on a small corpus. A more mature conceptual framework for grouping blogs would help guide our experiment design.

Future studies can also look into the role that individual blog entries play in classification. We started by classifying at the blog entry level and found the title of the entry to be helpful. But we switched to categorizing entire blogs because Technorati provided inbound links on a per blog basis. It would interesting to first classify individual entries and then apply a voting algorithm classifier or a percentage threshold to determine in which category a blog belongs.

Finally, modifying the features to utilize bi-grams and n-gram analysis might prove fruitful. Because popular vernacular phrases spread through blogs as memes, these multi-word tokens appear throughout particular categories of blogs. A good feature is a linguistic token that occurs not too frequently but, also, not too infrequently in a corpus.

The nascent state of blogs as a genre limits the effectiveness of NLP techniques. In addition, using supervised machine learning methods requires a tremendous amount of human annotation to prepare the training dataset. In spite of these difficulties, the results show that automatic classification blogs using machine learning is feasible.

Acknowledgements

The authors would like to thank Professor Marti Hearst for her tutelage.
References

http://www.pewinternet.org/pdfs/PIP_Content_Creation_Report.pdf
http://www.access-egov.info/glossary.cfm?sid=PA
http://magpierss.sourceforge.net/
[8] Natural Language Toolkit (NLTK)
http://nltk.sourceforge.net/index.html
[9] Weka: Data Mining Software in Java
http://www.cs.waikato.ac.nz/ml/weka/
[10] Technorati Cosmos API
http://technorati.com/developers/

Appendix A

<table>
<thead>
<tr>
<th>Linguistic Feature</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigrams (baseline)</td>
<td>84%</td>
</tr>
<tr>
<td>Title + 1st sentence</td>
<td>76%</td>
</tr>
<tr>
<td>Anchor text</td>
<td>80%</td>
</tr>
<tr>
<td>Title + 1st sentence and Anchor text</td>
<td>73%</td>
</tr>
</tbody>
</table>

Table 1: Classification accuracy for 120 blogs

Appendix B

| Train Dataset Results
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>classified as</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----------------</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>personal</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>7</td>
<td>0</td>
<td>news</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>22</td>
<td>0</td>
<td>political</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
<td>17</td>
<td>sports</td>
</tr>
</tbody>
</table>

| Test Dataset Results
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>classified as</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----------------</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>personal</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>news</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>political</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>sports</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrix for baseline data sets.