Clustering Blogs Based on Stylistic Characteristics

Matin Movassate MATINM@CS.STANFORD.EDU
Christopher Lin TOPHER@CS.STANFORD.EDU

Computer Science Department, Stanford University, Stanford, CA 94305

Abstract

Given a sufficiently broad genre such as technology or politics, Internet users typically have straightforward means of determining which websites and blogs offer content of the desired category. Unfortunately, users with more specific preferences or tastes that relate more closely to style of writing rather than the subject matter itself (for instance, observational humor or opinionated prose) will have limited means of discovering content suited to their tastes. In order to generate a more natural and intuitive grouping of web pages, we applied k-means clustering and principal component analysis across a diverse set of textual features to a large corpus of blog data. Our results showed that our training corpus contained between $k = 10$ and $k = 20$ natural style clusters, and that appearance of word bigrams and use of punctuation are the best indicators of a blogger's writing style.

1 Motivation

While many content aggregation systems on the Web (such as Technorati) aim to group websites and blogs based entirely on genre (e.g. grouping gardening blogs or film review sites together), few have attempted to track similarities that stem from literary considerations, such as the writing style used by these sites' authors (e.g. sentence forms and employment of particular figures of speech), the discursive tone of the writing (e.g. comedic, authoritative, or conversational) and employment of certain literary genre conventions (e.g. use of narrative and dialogue in short stories in contrast to a prose essay).

Although ignored thus far, such characteristics strongly influence a reader's perception of blog material. These language features, though difficult for a human being to precisely classify, have a dramatic effect on how much enjoyment a reader is able to derive from a given blog. Because blog text contains both structural and semantic signals that provide insight as to the tone, style and content of the writing, we can develop learning algorithms that aggregate on these features and develop a sensible grouping for our training data.

We therefore hope to use machine learning techniques to find clusters of blogs that share common attributes in stylistic characteristics, where groupings can produce sensible blog recommendations by simply suggesting blogs in the same cluster.

Successful clustering can have profound implications on recommendation systems. While most recommendation systems are based on predefined genre listings or explicit user-indicated preferences, very few recommendation systems actually base their groupings on the structure and content of the text. Perhaps this system could provide a more intuitive means of determining blog suggestions, and thereby increase the utility and enrich the experience of online self-expression.

Because we are employing unsupervised learning algorithms, it will be difficult for us to quantitatively gauge the effectiveness of our methods. The whole impetus for our application is to discover a more natural grouping of websites that can not be easily quantified by a human being. Though this fact admittedly makes the prospect of "success" difficult to quantify, it makes the application as a whole much more unique and intriguing.

2 Data Collection

For obtaining a sufficiently large amount of user-generated content spanning a wide variety of different blogs, we identified major blogging platform websites such as Blogger and Wordpress as the most valuable online resources. For amassing a sufficient corpus of training data, we ultimately took advantage of Blogger's "random blog" feature to extract the 25 latest posts from 2,155 randomly-chosen blogs, resulting in a total corpus of roughly 50,000 blog entries. HTML markup was directly downloaded with a scraper script using a combination of Python's `urllib2` standard libraries and Google's Data Feed API. A parser was built with the Python library BeautifulSoup in order to parse the downloaded XML markup associated with each blog feed and ultimately extract the desired blog content. Finally, once we attained enough

blog text, we stored the associated data (along with some relevant metadata, such as date of posting) in a `MySQL` database. Blog text was preprocessed and standardized to be all lower case, with extraneous formatting markup stripped out, and images, videos and audio replaced by the tokens `<<IMAGE>>`, `<<VID>>` and `<<AUDIO>>`.

3 Feature Selection

By expanding upon previous work in computational stylistics, we were able to leverage an expansive list of linguistic features aimed to highlight stylistic similarities amongst blogs. For each blog in our data set, we obtained the following values and aggregated them all into a single float-valued vector. Each feature (or set of features) describes the type of writing style we were hoping to infer.

3.1 **Average Length for Posts, Sentences and Words:** Longer entries suggest a more involved treatment of the subject matter, longer sentences suggest a more thoughtful (and perhaps stream-of-consciousness) level of prose, and longer words suggest a more verbose vocabulary.

3.2 **Word and Character Frequencies:** Examining the appearance of particular words can be indicative of a document's overall style, as evidenced by previous work on authorship attribution used to track correlations in word usage [1]. For word frequencies, we collected the top 50 most common words in our entire dataset and simply calculated the frequency of each of the top 50 words for each individual blog.

3.3 **Function Word Frequencies:** Function words are defined as tokens that have little lexical meaning but express grammatical relationships between other words in a sentence (`a`, `the`, `his`, `to`). Frequencies of function words are an attractive feature because they are insensitive to a particular subject matter, yet can be indicative of certain types of sentence constructions and phrases [2].

3.4 **Word-Level and Character-Level Bigrams:** Word combinations such as fixed phrases and collocations up to lengths of seven have been previously used in authorship studies of Shakespearean works and can be indicative of a particular idiolect (or individual style of writing) [3]. Our bigrams involved every possible two-word combination of the top 50 words in our entire dataset (note that tokens such as `<<IMAGE>>` are considered words by our model).

3.5 **Parts of Speech:** A relatively high frequency of adjectives and adverbs could indicate colorful, descriptive text, while a relatively high frequency of nouns could indicate heavy use of lists.

3.6 **Punctuation Use:** Frequency of punctuation characters are traditionally a successful indicator of authorship, mostly because of the opportunity of variation in usage [4].

3.7 **List and Quotation Use:** Frequent use of quotations and dialogue may indicate a narrative style and storytelling approach to writing rather than an expository style more typical of an essay.

3.8 **Vocabulary Size:** Blogs employing more verbose diction suggest a higher level of intellectualism in the text and target audience.

3.9 **Readability Metrics:** For each blog, we calculated readability via the Flesch Reading Ease test:

$$206.835 - 1.015\frac{\text{total words}}{\text{total sentences}} - 84.6\frac{\text{total syllables}}{\text{total words}}$$

The ease with which a reader comprehends a document can greatly affect his/her enjoyment of the document, and may also give indicators as to the document's formality and target audience.

3.10 **Web-Specific Features:** As Web texts, blogs can integrate rich media directly into their content. Blogs that do so frequently contrast with blogs that focus on a style more consistent with traditional text prose. We tracked occurrences of embedded images, videos and audio in a blog (by replacing the content of HTML `<img>` tags with a catch-all `<<IMG>>` token, for instance), as well as the frequency of emoticons (`:-D`) and common web acronyms (`LOL`).

4 Algorithms and Model

We took advantage of a variety of unsupervised learning algorithms in order to generating blog clusters. Once we generated our feature vectors of length n=5,692 for each of our m=2,155 blogs, we fed (at least some portion of) the feature set matrix $\in \mathbb{R}^{m \times n}$ into the following algorithms.

4.1 k-means Clustering

Our implementation of k-means clustering generated groupings based on three input values: a feature vector $f = \{f_1, f_2, \ldots, f_p\}$, $p \leq n, f_i \in F$ (where F is the original feature vector of size n), number of clusters k and number of replications r. The number of replications is the number of times the k-means clustering algorithm was run, with different initial values generated at the start of each run to overcome local optima.

4.2 Principal Component Analysis

Because the dimensions of our feature matrix are skewed such that $n \gg m$, it makes sense to apply some level of dimensionality reduction, so that we may reduce noise and superfluousness in our feature values. In particular, the high number of word and character frequencies in our original feature vector F, as well as word and character bigram frequencies, suggest the possibility of a great deal of noise in the data. We use principal component analysis to attempt to correlate features with one another and reduce the dimensions of our feature vector.

4.3 k-nearest Neighbors

Another algorithm that deserves mention is k-nearest neighbors (KNN). Though easy to implement, this algorithm turned out to live up to its notoriety for being slow: since our training set was rather large, many distance computations needed to be made, which was simply impractical given the size m, n of our training set and feature vectors. Preliminary results were no better than k-means clustering, and were produced at a substantially slower pace, so this algorithm was mostly ignored for the purposes of this project.

5 Experimental Results

In an effort to assess the density of our k-means clustering result, we examined how the tightness of each cluster produced correlates with the value for k. the means for each produced cluster to assess how distinct and densely-grouped our k clusters are. The intuition is that, the more dense our generated clusters are, the more successful they were in finding similarly styled blogs. The equation used to calculate the average distance of an example from a cluster's centroid was calculated by:

$$\frac{\sum_{j=1}^{k}\left(\frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\}\sqrt[2]{\sum_{f=1}^{n}(x_f^{(i)}-\mu_f^{(j)})}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}\right)}{k}$$

Where we let $\mu^{(i)} \ldots \mu^{(k)}$ be the generated centroids $\in \mathbb{R}^n$, and $c^{(i)}$ is the cluster index of example i. The results we obtained for each are in Figure 1. Also, refer to Figure 2 for the results obtained from our run of principal component analysis.

6 Analysis and Errors

In discussing the success of unsupervised learning algorithms, we will need to take a more qualitative approach in assessing the results of our methods and algorithms. While it is relatively straightforward to measure success with supervised learning algorithms, unsupervised learning algorithms lack any clear structure *a priori*, and so need to be justified not just with our quantitative metrics, but also with some qualitative insights.

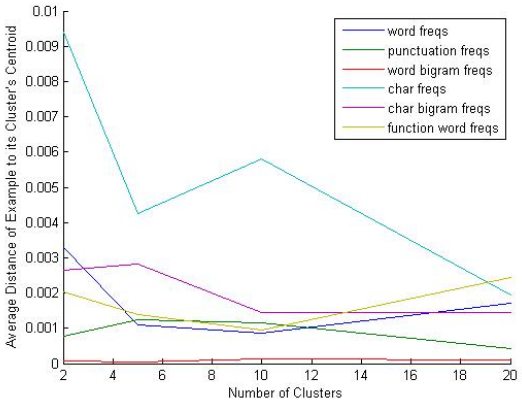

Figure 1: Plot of k-values to average distance from examples to associated cluster centroids for each subset feature vector. Word bigram frequencies perform best, followed by punctuation frequencies. Results for feature vectors including average length, readability scores, and part-of-speech frequencies are excluded because of extremely poor performance.

For trials involving the entire feature set with $k = 5$, $k = 10$ and $k = 20$, the algorithm was capable of matching up a blog involving introspective, melancholy analysis of vintage comic books with another blog involving a teenaged girl discussing – at a deep and emotional level – the angst of being a high school student. Though both blogs concern entirely different subject matters, they share striking stylistic similarities in how they forlornly and contemplatively address their respective topics. Additionally, it managed to cluster together a how-to blog on knitting alongside a how-to blog on computer modding. Again, despite concerning wildly different domains, these blogs presented similar treatments of their respective subjects. This suggests that our features are often successful in picking out the hidden literary details that make one blog stylistically similar to another.

For k-means clustering, a high value of k (such as 100 or 200, which then corresponds to a small cluster size) had mostly meaningless results. This is because there simply was not a diverse enough array of blog types within our data set to justify k values of 100 or 200. Groupings in this case seemed mostly arbitrary.

From observing our blog clusterings, it becomes clear that a nontrivial portion of our dataset was rather homogeneous. Many of our clusterings seemed to collect a skewed set of user blogs; specifically, a large portion of our collected blogs were (strangely enough) image galleries showcasing the latest fashion trends. This could be because of the audience Blogger

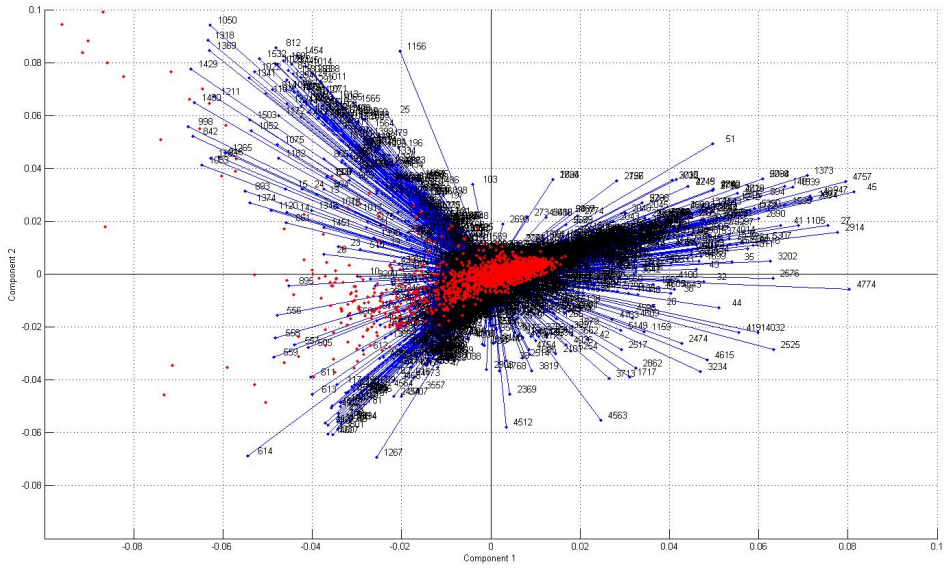

Figure 2: Plot of training examples and feature vectors against first two principal components. Red points represent training examples, and blue vectors represent individual features. Blue vectors are labeled with numbers encoding features. The first principal component is most correlated with features that measure the frequencies of word and character bigrams, while the second principal component is most correlated with character frequencies.

attracts. Perhaps, other than a few niche purposes, the site is becoming an increasingly-unpopular destination for online expression, meaning many serious users are turning to competing blogging platforms like Wordpress.

In addition, since our word bigram model was built over a language of the top 50 words in our dataset, there were $50 \times 50 = 2,500$ feature values devoted to word bigram frequencies. And because very few of these bigrams actually appear in the blog text, most of these values end up being 0. Perhaps a better construction for the bigram models would have been to build the word bigram features directly from the most frequently occurring bigrams, instead of constructing bigrams from the most-frequently occurring unigrams. Also, implementing some form of Laplace smoothing or Good-Turing smoothing could help alleviate some of the sparsity in our bigram model.

Finally, for our k-means clustering trials, we ran into the possibility of encountering optimization errors, where the algorithm would converge onto a local optima instead of a global optima. This issue was slightly alleviated by running multiple iterations of k-means with randomized initial values.

7 Conclusion and Future Work

Overall, both k-means clustering and principal component analysis produced many startlingly intuitive blog clusterings. For instance, k-means clustering using a combination of word bigram frequencies and punctuation frequencies managed to link together fashion blogs and family photo blogs that, despite focusing on different subjects, featured casual yet grammatical prose and a personally reflective tone. These results correspond to authorship attribution literature that highlights the usefulness of word bigrams and punctuation in computational stylistic analysis.

Principal component analysis suggests that the dimensionality of the feature vector can be reduced by combining features measuring character frequencies and features measuring frequencies of some word and character bigrams.

In the future, our method of measuring clustering success will most certainly need to be improved. Clustering text – especially online text – is a difficult problem, as content and style can vary dramatically from one blog to another. Thus, given a cluster, it can be difficult for a human being (or computer program, for that matter) to hastily scan through each blog in the cluster and determine whether success was achieved or not. But this is as expected: the entire purpose of the project was to produce groupings that rely on the *deeply-rooted* structure of the language itself, not simply some surface-level attribute that can easily be skimmed. To obtain as accurate a gauge on success as possible, in the future, we would consider conducting surveys and human interaction studies, asking par-

ticipants to provide their opinion on the legitimacy of produced clusterings. Such an endeavor turned out to be too costly to execute in the timeframe given for the project.

Our feature vectors also had the issue of being extremely sparse, mostly due to our word-bigram frequency features. When building our feature vector in the future, it will be useful to run feature selection algorithms leveraging attributes like mutual information. This will allow us to determine which features are the most fruitful in our large feature set, enabling us to reduce the dimensionality of our trial matrix and allow for more tractable computations on our entire data set (instead of getting bottlenecked by our very large, mostly meaningless feature vectors).

Much computational power and time was also wasted calculating blog clusterings for different cluster sizes. Since the cluster number k is a nuisance parameter of the k-means clustering algorithm, in the future, we plan to leverage techniques like v-Fold Cross-validation to determine the optimal size of k before running our clustering algorithm several times for different values of k.

We would also need to consider obtaining data from a variety of sources, instead of just a single source. In our case, all our blog data was extracted from a single website: Blogger.com. While the resulting consistency in HTML and XML formats alleviated the arduous task of data collection and preprocessing, it also introduced an unwanted degree of homogeneity into our training data. Though Blogger.com is an all-purpose blogging platform, it seemed to attract a surprisingly narrow set of opinions, as explained in our Error Analysis. In further work, we will extract data from other sources such as Wordpress and Technorati.

In the future, we could experiment with more sophisticated clustering algorithms such as Fast Genetic k-means Clustering Algorithm. This method provides promising benefits over normal k-means, since it's guaranteed to converge to a global optimum and claims to run considerably quickly [5].

8 Acknowledgements

We would like to thank Matthew Jockers, from the department of Humanities Computing, for his advice on feature selection related to computational stylistics.

References

[1] Craig, H (2004). Stylistic analysis and authorship studies. A Companion to Digital Humanities, ed. Susan Schreibman, Ray Siemens, John Unsworth. Oxford: Blackwell. http://www.digitalhumanities.org/companion/

[2] Koppel, M., Argamon, S., and Shimoni, A. (2002). Automatically categorizing written texts by gender. *Literary and Linguistic Computing* 17(4): 401-12.

[3] Lancashire, I. (1997). Empirically Determining Shakespeare's Idiolect. *Shakespeare Studies* 25: 171 p85.

[4] Grieve, J. (2007). Quantitative Authorship Attribution: An Evaluation of Techniques. *Literary and Linguistic Computing* 22: 3.

[5] Lu, Yi et al. (2004). FGKA: a Fast Genetic K-means Clustering Algorithm. *Symposium on Applied Computing.*