Stefan Sebastian, 242

Abstract

TODO abstract

Contents

1 Introduction

1.1 Motivation

Beer is one of the oldest and most popular alcoholic drinks. It is a staple drink in many societies and has been around ever since the formation of human civilizations, with evidence that dates it to more than 10,000 years ago[1]. It's popularity is attributed to having a moderate quantity of alcohol, which leads to more positive drinking experiences, and the fact that it can be drunk at any social event[1].

The global beer market was valued at 593$ billion and is expected to rise to 685$ billion by 2025[2]. The most popular beer style is by far lager, but others are rising in popularity. Craft beer is a type of beer that is produced in microbreweries, in a traditional style and with an emphasis on taste and variety. The global craft beer market accounted for 38$ billion and is growing at an explosive rate of 14.1% each year[3].

It is obvious that there is a huge public interest in this drink, and any insights into what consumers think about beer could have serious economic benefits. This paper presents an experiment on extracting information using unsupervised learning on beer reviews. The main goals are finding natural clusters among beer descriptions and what words are used to describe similar beers. Also the clusters are analyzed by the most common beer styles to see if there is a correlation between reviews and beer types.

1.2 Related work

1.2.1 Text clustering

A comprehensive review on text document clustering similarity measures was published by Anna Huang[4]. The paper describes a complete text clustering process with emphasis on distance measurements. Features are extracted using the tf-idf method and the clustering algorithm is the basic K-means variant. A set of similarity measures are studies such as cosine similarity, euclidean distance, Jaccard coefficient, etc. The datasets considered vary from news to scientific articles.

An implementation of a Scatter-Gather document browsing system was presented by Larsen and Aone[5]. The system applies document clustering on the whole dataset, then when the user selects a group, it clusters the selected group into smaller groups. This operation can be extends until each group contains only one document. The algorithm used by the system is K-means with an improvement in seed selection consisting in updating the initial means after every point assignment. The algorithm uses as a feature extraction method the tf-idf vectorizer.

An interesting approach to text clustering is the one developed by Hu et al.[6]. Their method consists of enriching text information by using Wikipedia concept and category data. A dataset was made from Wikipedia articles related to different concepts. When a document is evaluated the term weight is calculated using the combined tf-idf score from the training and the Wikipedia concept dataset. The category score is built by analyzing the most important concepts of the document. The clustering algorithm used in the experiment was spherical K-means. This method was tested on three datasets and was shown to obtain better results than simple clustering over tf-idf scores.

1.2.2 Review analysis

Most papers on review analysis are focused on extracting information about aspects present in the review. For example, Lu et al.[7] propose an algorithm for inferring the aspects considered in a review and assigning a score to them, and use it on hotel reviews from tripadvisor. The algorithm, called LARA, works in two steps. In the first step, given some keywords for the aspects, it associates each sentence in a review with the aspect for which it has the biggest overlap. In the second step, it computes a score for each aspect based on the sentiments present in the sentences found in previous steps.

Another direction of review analysis is to use linguistic rules in a constrained domain. For instance, Iacob and Rachel[8], created an algorithm for automatically extracting and scoring feature request from mobile application reviews. The algorithm uses a set of 237 rules which were build by manual analysis of review data. An example rule is "(adding) <request>would (<ADVERB>) be <POSITIVE-ADJECTIVE>". The Latent Dirichlet Allocation algorithm is then used on the discovered feature requests in order to find topics.

1.2.3 Experiments on beer datasets

McAuley and Leskovec[9] built a recommender system that adapts to user experience. The motivation was that the users' preferences evolve as they acquire experience. The datasets used for the experiment included the BeerAdvocate review dataset[10] because of the fact that beer is one of the most widespread acquired tastes. The experience label is learned by the model by using rating and timestamp data.

Braun and Timpe[11] scrape data from BeerAdvocate, RateBeer and CellarTracker websites. These contain user text reviews and numerical ratings for different types of beer and wine. The goal of the experiment is to predict scores based on the given review. The features are extracted using the bag-of-words model for single words an bigrams. To obtain more relevant features, the input data is stemmed, cleared of stop words, and the tokens are tagged by part of speech and only adjectives, nouns, verbs and adverbs are kept. The algorithms used for classification are Naive Bayes and Support Vector Machines. The model obtained an accuracy of up to 87%.

1.3 Paper outline

TODO outline per chapters

2 Dataset

The dataset used for this experiment is a collection of beer reviews taken from the BeerAdvocate website. The original dataset was made up out of 1.5 million user reviews, from 33387 different users, collected between 1998 and 2011. It is not available at this time but I found a subset of around 500 thousand reviews on data.world[10].

The data is in csv format, each row containing various information like: beer name, beer style, alcohol content, scores for taste, appearance, aroma and a textual review. The only columns considered for this experiment were the beer style and the text review.

On exploratory data analysis, 407 duplicate and 119 missing reviews were found. Also there are 104 distinct beer styles and the amount of data for each of them is quite

imbalanced as can be see in figure 1, where each bar represents a beer style with a size proportional to the number of reviews for it.

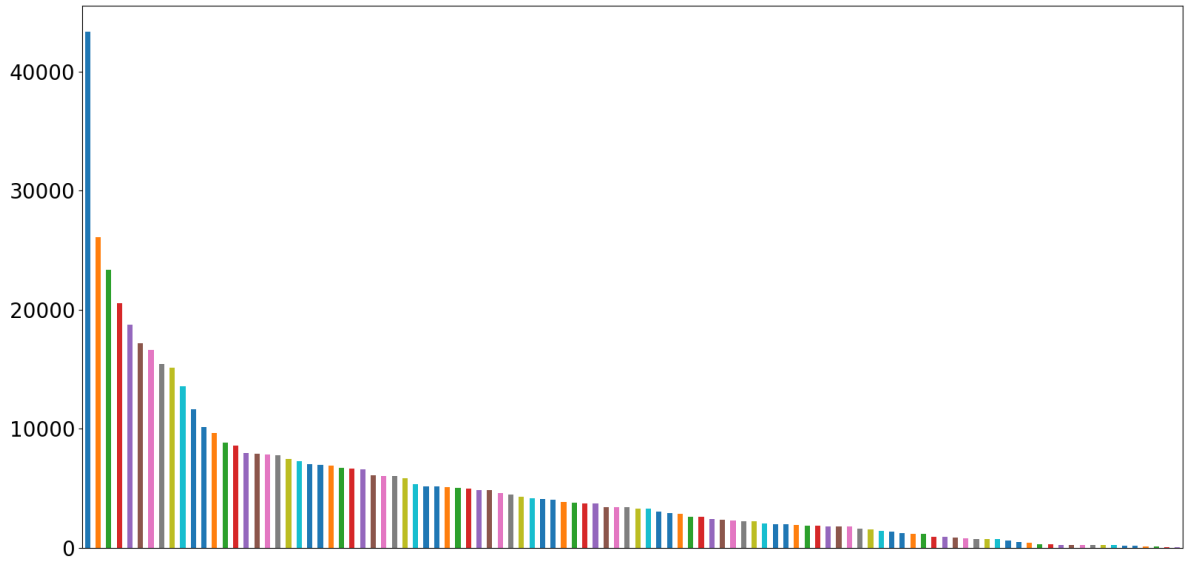

Figure 1: A plot of the number of reviews per beer style

The first step in preprocessing this dataset was to remove rows containing duplicates and missing values in the columns that are relevant for the experiment: beer style, text review. Next, some of the styles for which there wasn't a lot of data available were dropped, as they would be outliers for clustering. The threshold for this cut was chosen arbitrarily at 7000 reviews.

Upon inspection of the beer styles present in the dataset, a fine granularity was observed. For example, the difference between American Pale Ale and American India Pale Ale is minimal and not very rigorous. This and other similar cases would only confuse the classifier. For this reason, a mapping was made for each style to a broader category, using the taxonomy published by BusinessInsider[12].

The final mapping can be seen in figure 2. Finally, a balanced dataset is built by selecting the minimum value for which we have an equal distribution of reviews per beer style, which is around 9500 per style. The analyzed dataset is made through random sampling of 9500 values for each of the remaining beer styles.

3 Solution description

3.1 Model overview

In order to process the textual data, some features need to be extracted by using natural language processing techniques. The method used to extract numerical features about the words in the dataset is the Tf-Idf vectorizer. Tf-Idf[13] stands for term frequency-inverse document frequency and is the most popular method for weighting term. In brief, the importance of a word is proportional to the number of times it appears in a document and inversely proportional to the number of documents it appears in. Before applying Tf-Idf, the review texts are tokenized, the stop words are removed and then each word is stemmed. Stemming is the process of reducing a word to its root, or base form, and the

Figure 2: Mapping of beer styles from the original dataset

algorithm selected for this is the Snowball Stemmer[14], which works by defining a set of rules for replacing word endings. The number of terms considered as features has been determined experimentally. Finally, the dataset contains a list of array that denotes how important is each considered term for every review.

To solve the clustering problem the classic K-Means algorithm is used with two different seed initialization methods: random values and k-d trees. The first method is very simple conceptually, just choose k random points from the dataset. The second method has been described by Redmon and Heneghan[15]. The basic idea is to use this data structure to divide multidimensional features into buckets and the make density estimation depending on the size of the bucket and the volume of the bounding box.

The resulting clusters are evaluated based on initial beer style labels and the dominant terms in each cluster.

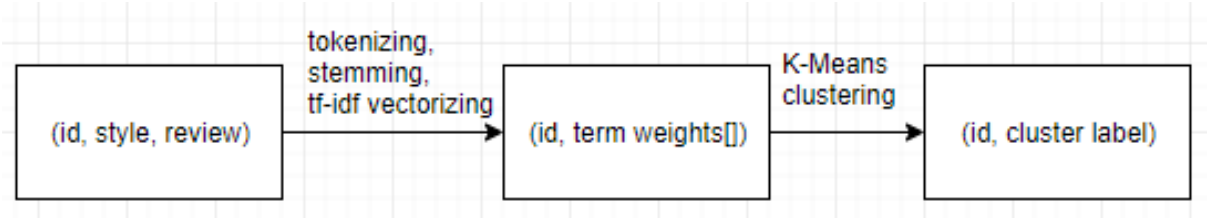

Figure 3: Diagram of the transformations on the dataset

3.2 Technologies used

The project was implemented in the Python programming language on the Anaconda distribution. The clustering and seed initialization algorithms were implemented from scratch.

For data analysis and preprocessing the pandas[16] library was used, which provides some efficient data manipulation and analysis tools. The group by function was particularly useful for operations on all data of a particular beer style.

The algorithms for tokenizing and stemming were taken from the nltk library[17], which is one of the most popular tools for natural language processing. The Tf-Idf

vectorizer implementation was provided by the scikit-learn library[18], which contains a comprehensive collection of data analysis algorithms.

In order to visualize the data and the results the following plotting tools were used. Matplotlib[19] provided a tool to make bar charts in order to observe the initial data distribution and Seaborn[20] was used to generate a heatmap of the resulting clusters.

3.3 Parameters

The most important parameter used in the experiment is k, the number of clusters. This was chosen to be the number of distinct beer styles in the dataset, 9, as the reviews for beers of the same type should be similar to each other.

The number of features for each data point is equivalent to how many word frequencies are considered. This was determined experimentally by running the basic algorithm with some different feature values. The results, shown in table 3.3, indicate that until around 1000 there is the biggest growth in performance.

Nr features	Precision	Recall	F1
200	0.420	0.481	0.449
700	0.501	0.554	0.526
1000	0.537	0.532	0.534
1500	0.515	0.528	0.521

Two methods were tested for seed initialization: random points and kd-tree based. The run with kd-trees obtained a 0.47 F1 score while, random initialization got a 0.53 F1 score. This might be caused by finding good starting points through randomness or that the large number of features is not a good fit for the kd-tree data structure.

4 Evaluation

4.1 Metrics

The direction chosen for evaluation is how well the clusters found separate the initial dataset by beer types. In order to do this, each cluster is labeled with the most common style of beer among its elements. This turns our problem intro a multi-label classification one.

The metrics used are the ones described by Beleites at al.[21] for multi-class problems. First of all, a confusion matrix is computed, which is a lxl matrix where l is the number of features. Each element (x, y) in the confusion matrix has a value meaning how many times a data point with label x has been classified as having label y. Precision and recall are then calculated for each label, simulating the binary evaluation.

Precision, also called positive predictive value, is a fraction that represents how many elements classified with a label have been assigned correctly. Recall, also known as sensitivity, is the fraction of instances correctly assigned to a class and all instances from that class. For the binary case, precision and recall can be calculated as in figure 1, where tp, fp, fn stand for true positives, false positives and false negatives. This can be extended to the confusion matrix, for each row, as follows: precision is the fraction of the value on the diagonal and the sum of values on the column, recall is the value on the diagonal divided by the sum of values on the row. The precision and recall of the

multi-label classifier can then be computed as the means of the values obtained from the previous step.

$$precision = \frac{tp}{tp + fp} \quad recall = \frac{tp}{tp + fn} \tag{1}$$

F1 score is a combined measure of precision and recall, used to provide a single measurement for the system. It represents the harmonic mean of the two values, as in figure 2, where mp and mr are mean precision and mean recall.

$$F1 = \frac{2 * mp * mr}{mp + mr} \tag{2}$$

4.2 Results

4.2.1 Multi-label classification score

The correlation between the beer styles of the data points and the type that is most prevalent in the resulting clusters was chose as a measure of evaluation. For the best run, the result can be seen in table 4.2.1. These were obtained using the random seed initialization method.

Mean precision	0.537
Mean recall	0.532
F1 score	0.534

4.2.2 Heatmap

In order to better visualize the results a heatmap was generated. Visible in figure 4, the diagram helps visualize which style of beer is more common in which cluster. As expected, there is a clear separation for most of the dominant styles in each cluster. This is best observed for the following cluster, style pairs: 0 with Porter, 1 with Stout, 2 with Pale Lager, 4 with Wheat Beer, 5 with Belgian Ale, 6 with Fruit/Vegetable Beer and 7 with Barleywine. For cluster 3 there are two dominant classes : Pale Ale and Amber, which can be explained by the fact that Amber is a particular style of ale. For cluster 8 there is no clear majority, meaning it might be a collection of outliers from other styles.

4.2.3 Common words per cluster

In order to interpret what features these clusters have in common, we can look at the most popular words for each cluster, identified by id. Figure 5 shows the 10 words with the highest tf-idf score for the mean of each cluster. Looking at this figure we can find the reason for the shape of cluster 8, as the most common words are general words used in beer description, like 'beer', 'taste', 'drink', 'flavour', which can be applied to any beer style.

Another thing we can observe from the table of common words is the difference between 'Porter' and 'Stout' styles, which are often put under the same category. According to the diagram, Porter beers tend to have a stronger chocolate, roast, coffee taste while Stouts also have a fruity, citrus taste.

An interesting observation is that while the Wheat Beer style is clearly associated with cluster number 4, the feature words for this cluster: 'chocolate', 'dark', 'roast' are not at all representative of the style.

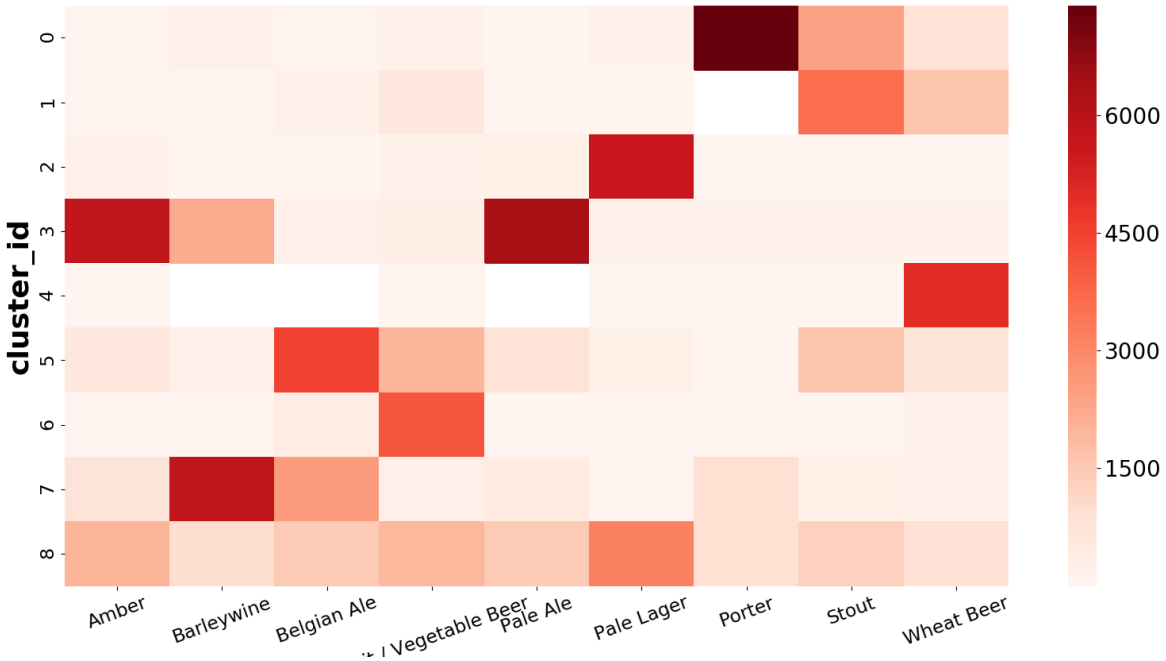

Figure 4: Heatmap of beer styles per cluster

```
0  :  ['chocol', 'coffe', 'roast', 'dark', 'black', 'stout', 'veri', 'beer', 'malt', 'flavor']
1  :  ['wheat', 'banana', 'lemon', 'beer', 'clove', 'light', 'veri', 'tast', 'orang', 'citrus']
2  :  ['lager', 'corn', 'macro', 'light', 'beer', 'adjunct', 'tast', 'yellow', 'grain', 'veri']
3  :  ['hop', 'malt', 'nice', 'bitter', 'veri', 'amber', 'citrus', 'caramel', 'sweet', 'beer']
4  :  ['porter', 'chocol', 'roast', 'coffe', 'dark', 'malt', 'brown', 'nice', 'veri', 'flavor']
5  :  ['appl', 'spice', 'veri', 'yeast', 'light', 'sweet', 'beer', 'white', 'orang', 'flavor']
6  :  ['cherri', 'raspberri', 'tart', 'beer', 'sweet', 'like', 'fruit', 'sour', 'veri', 'red']
7  :  ['alcohol', 'dark', 'sweet', 'caramel', 'brown', 'veri', 'fruit', 'malt', 'hop', 'barleywin']
8  :  ['beer', 'tast', 'like', 'veri', 'smell', 'good', 'drink', 'just', 'realli', 'flavor']
```

Figure 5: Most common words for each cluster

5 Conclusions

References

[1] C. Misfud, "Why beer is the worlds most beloved drink." http://time.com/5407072/why-beer-is-most-popular-drink-world/?#, 2018.

[2] A. M. Research, "Beer market by type - global opportunity analysis and industry forecast 2017-2025." https://www.alliedmarketresearch.com/beer-market, 2017.

[3] "Global craft beer market report 2018." https://www.prnewswire.com/news-releases/global-craft-beer-market-report-2018-300657436.html, 2018.

[4] A. Huang, "Similarity measures for text document clustering," in *Proceedings of the 6th New Zealand Computer Science Research Student Conference*, 2008.

[5] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, (New York, NY, USA), pp. 16–22, ACM, 1999.

[6] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou, "Exploiting wikipedia as external knowledge for document clustering," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, (New York, NY, USA), pp. 389–396, ACM, 2009.

[7] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: A rating regression approach," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, (New York, NY, USA), pp. 783–792, ACM, 2010.

[8] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, (Piscataway, NJ, USA), pp. 41–44, IEEE Press, 2013.

[9] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," *CoRR*, vol. abs/1303.4402, 2013.

[10] "Beer advocate reviews." https://data.world/petergensler/beer-advocate-reviews, 2011.

[11] B. Braun and R. Timpe, "Text based rating predictions from beer and wine reviews." https://pdfs.semanticscholar.org/d88e/4b5c117283b20476ebb7f4bbaac892028cc1.pdf, 2015.

[12] M. Stanger and S. Gould, "Everything you need to know about beer, in one chart." https://www.businessinsider.com/different-types-of-beer-2014-12, 2014.

[13] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, vol. 2. 02 2008.

[14] M. F. Porter, "Readings in information retrieval," ch. An Algorithm for Suffix Stripping, pp. 313–316, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[15] S. J. Redmond and C. Heneghan, "A method for initialising the k-means clustering algorithm using kd-trees," *Pattern Recognition Letters*, vol. 28, no. 8, pp. 965 – 973, 2007.

[16] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference* (S. van der Walt and J. Millman, eds.), pp. 51 – 56, 2010.

[17] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, (Stroudsburg, PA, USA), pp. 63–70, Association for Computational Linguistics, 2002.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and douard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, 2011.

[19] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science and Engineering*, vol. 9, pp. 90 – 95, 2007.

[20] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, and A. Qalieh, "mwaskom/seaborn: v0.8.1 (september 2017)," Sept. 2017.

[21] C. Beleites, R. Salzer, and V. Sergo, "Validation of soft classification models using partial class memberships: An extended concept of sensitivity and co. applied to the grading of astrocytoma tissues.," *Chemometrics and Intelligent Laboratory Systems*, vol. 122, pp. 12–22, 2013.