Detection of Humor in Online Product Questions

Implications of humor in written language in online product questions

Mattison Hineline
Masters in Data Science Program
Colorado University Boulder
Malaga, Spain
mattisonhineline@gmail.com

ABSTRACT

As the internet and online shopping become more prominent in today's world, it is imperative that we develop ways to identify and understand written language and its tone so that the social and economic impacts are positive. Specifically in this study, we will look at ways to analyze online product questions and whether or not they are meant to be humorous, ironic or funny, or if they are an honest question (both positive or negative). We use two techniques to try to identify the questions: KMeans clustering and Random Forest Regression (ensemble). We find that the Random Forest Regression is more apt at correctly identifying the labels.

INTRODUCTION

This study aims to progress the technical field such that computers are able to identify funny and honest written text about products.

In verbal language, humans use inflections in the voice, word emphasis, hand gestures, and facial expressions, among other techniques, to convey the emotion, feeling and tone. Further, when communicating without visual cues it can be much more challenging to understand the implications of one's words.

Most people have some sort of experience with these challenges when it comes to common actions such as texting, emailing or posting on social media. Because of such challenges regarding understanding emotions via short written language exchanges, the famous "emoji" was created to help facilitate more natural communication [1]. Henceforth, people are generally more able to express emotions such as happiness, sadness, and anger, among others. With such useful context information given by emojis, it is easier to know how the intended meaning of the phrase is to be interpreted. For example, a short text of "Come home now" with a smiling emoji versus an angry emoji versus a party emoji will all have significantly different interpretations.

In online product reviews and questions, generally people do not use emojis. Therefore, the question we put forth in this project is whether or not it is possible to identify humor in online product questions or not, and how accurate those results are. Without helpful guidance, such as the emoji, humans often struggle understanding text because it "cannot accurately convey tone, emotion, facial expressions, gestures, body language, eye contact, oral speech, or face-to-face conversation, it is likely messages will be misinterpreted or misunderstood. The real meaning of your message gets lost through the medium." [2]

Although natural language processing (NLP) has been developing for many years, our means of communicating continue to evolve. We started with things such as email and now we post quick daily tweets. Online reviews and questions are quick and easy to do and are just another, fairly new, way to communicate our thoughts and opinions to others. However, humans commonly give honest reviews as well as humorous ones, just like one would in real-life situations person-to-person. For this reason, we propose further exploration in NLP by specializing in online product questions.

Many works and studies have been published in the theme of NLP because it is such a broad subject as well as complex. Prior work has focused on expressions or idioms and their implications [3] as well as more related topics such as hotel reviews [4].

Through this project, we hope to find another option of processing text which could then be applied further to new studies regarding NLP.

RELATED WORK

Prior work has shed significant light on natural language processing and the complexities it holds. There are so many different routes to take with regards to NLP which is exciting but also can seem overwhelming.

Take for example "Multiword Expressions: A Pain in the Neck for NLP" [3]. This study looked at many multiword expressions and how they can pose challenges to today's language processing technology. Spoken language has many unwritten rules as well as clearly defined ones. A computer may struggle to understand complex expressions and when text implies one thing versus the literal written words.

More related to our proposed study, the work of Ghorpade and Ragha [4] focuses on online reviews and categorizing

them into their sentiments by using well-trained sets. Although for our study this sounds appealing, we have decided that categorizing into two groups (humor versus serious) is a step that should be taken first. The reasoning behind this is partly because of the data set we are using but also because logically, companies looking at product questions may first want to know what is meant to be a serious review so that they can fix or better their product, then secondly analysis of the humorous reviews may prove helpful in marketing, customer engagement or other aspects.

Continuing, Barbieri and Saggion and their study "Automatic Detection of Irony and Humour in Twitter" [7] follows along similar ideas as this project. They tried to identify humor and irony in online Twitter posts using cross-domain classification experiments. Their study inspired us to evaluate our proposed work and procedures and to improve upon them. In Barbieri and Saggion's study, they were able to identify many types of grammatical (structure, frequency, etc) and lexical (sentiments, ambiguity, etc) which helped them to identify which tweets were ironic and which were not. In our project, we will also look at sentiment analysis to try to classify the online questions, but we will focus on the overall sentiment score.

Lastly, Ziser, Kravi and Carmel [5] used the same dataset as we will be using to analyze with which they used a deep-learning framework. They used two properties of the product questions: incongruity and subjectivity. They were able to find an accuracy of 90.8% of detection. They claim their model "is the first to detect humor in [product question answering]" [5]. In our proposed study, we strive to achieve such metrics and build upon their predictive model.

PROPOSED WORK

For this project, as stated, we propose to identify humorous versus non-humorous online product questions using real data collected from online product question answering platforms. The data was gathered from the Open Data registry on AWS [6].

The datasets are all presented as CSV files with 9571 entries each where each entry is the product question and are recorded in rows. Each dataset has the following columns: 'question' (the text), 'product_description' (short description of the product), 'image_url' (url for the product image), and 'label' (1: humorous, 0: non-humorous). The three datasets are: humorous_unbiased (where all entries are pre-labeled as humorous: 1), non_humorous_unbiases (where all entries are pre-labeled as non-humorous: 0), and non_humorous_biased (all entries are pre-labeled as non-humorous: 0).

The datasets 'humorous_unbiased' and 'non_humorous_unbiased' will be used to train the models and do most of the analysis. This is because they are drawn from the same set of products. This will make training the model and analyzing the data easier. The 'non_humorous_biased' dataset contains product questions from randomly selected products. For this project, we will not be using this third dataset, however in the discussion section future report suggestions include it.

After taking a preliminary look at the data, we find the 'humorous_unbiased' (which will be referred to from here on as just 'humorous') to have some missing values.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9571 entries, 0 to 9570
Data columns (total 4 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   question             9571 non-null   object
 1   product_description  9547 non-null   object
 2   image_url            9230 non-null   object
 3   label                9571 non-null   int64 
dtypes: int64(1), object(3)
memory usage: 299.2+ KB
```

We can also take a look at the first few rows to get a clearer understanding of how the information looks.

	question	product_description	image_url	label
0	Will the volca sample get me a girlfriend?	Korg Amplifier Part VOLCASAMPLE	http://ecx.images-amazon.com/images/I/8111XZea...	1
1	Can u communicate with spirits even on Saturday?	Winning Moves Games Classic Ouija	http://ecx.images-amazon.com/images/I/81tkcYEG5...	1
2	I won't get hunted right?	Winning Moves Games Classic Ouija	http://ecx.images-amazon.com/images/I/81tkcYEG5...	1
3	I have a few questions.. Can you get possessed...	Winning Moves Games Classic Ouija	http://ecx.images-amazon.com/images/I/81tkcYEG5...	1
4	Has anyone asked where the treasure is? What w...	Winning Moves Games Classic Ouija	http://ecx.images-amazon.com/images/I/81tkcYEG5...	1

We further find that 'non_humorous_unbiased' produces similar initial information.

After importing the data and taking a first look at what we are working with, we must clean and preprocess the data so that it is usable.

First we will look at the missing values by simply deleting those rows. We have no missing values for the question and label columns, which is what we will be focusing on in this project. We have 24 total (0.25%) missing values for product_description and 317 (3.31%) missing values for image_url. Since we are unsure if we will use these columns for the model and the columns we need for modeling are 'question' and 'label', we will leave the missing values in the dataset for now. If needed later, we may simply delete the missing data rows because it shouldn't affect the final result significantly (only 0.25% and 3.31% of the data is missing).

Next, we will drop the 'image_url' column. For this project, we are more focused on the word choice of the texts than we are of the photo associated. For this reason, we will simply delete the 'image_url' column.

In any text driven study, you must preprocess the text so that it is workable and you are able to run models on the data. One goal is to find the common words (words that appear at least twice out of all of the reviews given) then take those words and set them into rows in a new dataframe. This is because we will be able to then count how many times, if at all, a word appears with each label. We will find the total occurrences, and ratios of humorous/non-humorous of each word. It was decided to delete the misspelled and rare words because there were so many randomly spread out which occurred just once out of all the questions and it caused very skewed data visualization and analysis later on.

Ultimately, the new data frame will look like this:

	word	humorous	non_humorous	total occurances	percent humorous	percent non_humorous
0	will	1445	405	1850	0.781081	0.218919
1	sampl	6	15	21	0.285714	0.714286
2	commun	17	5	22	0.772727	0.227273

Secondly, we will work with the original data frames and combine them. In addition, find the sentiment score, length of cleaned phrase and the average length of each word. The thought behind this is that it is possible humor isn't determined by traditional routes such as grammar and word form, but perhaps instead as simple as identifying sentiment, length of phrases and common words.

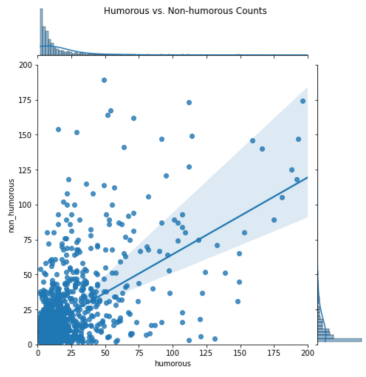
	question	product_description	label	q_no_punctuation	clean	clean > 3	tokenized	lemmatized	stemmed	number of words	all sentiment scores	sentiment compound score	avg word length
0	Will the volca sample get me a girlfriend?	Korg Amplifier Part VOLCASAMPLE	1	Will the volca sample get me a girlfriend	will volca sample get girlfriend	will volca sample girlfriend	[will, volca, sample, girlfriend]	[will, volca, sample, girlfriend]	[will, volca, sampl, girlfriend]	4	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	6.25
1	Can u communicate with spirits even on Saturday?	Winning Moves Games Classic Ouija	1	Can u communicate with spirits even on Saturday	can u communicate spirits even saturday	communicate spirits even saturday	[communicate, spirits, even, saturday]	[communicate, spirit, even, saturday]	[commun, spirit, even, saturday]	4	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	7.5
2	I won't get hunted right?	Winning Moves Games Classic Ouija	1	I won t get hunted right	i get hunted right	hunted right	[hunted, right]	[hunted, right]	[hunt, right]	2	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0,	0.0000	5.5

To have either such table the following measures must be performed:

1. Remove small words. For example: I, am, he, is, it, etc. To remove small words we will use the stopwords library from nltk.corpus. We remove these words because they are so common they will create noise in our data.
2. Convert all letters to lowercase and remove punctuation. Although uppercase letters, lowercase letters and punctuation could be useful in natural language processing, we chose to make the dataset as neutral as possible to really push the word choice models we will do further on.
3. Remove words that are 3 characters or less. Similar to stopwords, small words are removed to reduce noise.
4. Tokenize. Tokenizing the data will split the set so that we can lemmatize, stem and split the words into columns.
5. Lemmatizer. Using WordNetLemmatizer, we lemmatize the questions. Lemmatization takes a word and reduces it to its base form. For example: is → be, populated → populated, united → united.
6. As a comparison to lemmatization, we will also stem the data. Stemming gets to the core of each word compared to lemmatize. For example, stemming will result with: is → is, populated → popul, united → unite. After comparing the results of lemmatization and stemming, we have decided to use stemming because with so many different product questions, we really want to focus on the key concepts and core words people are using.
7. Now we see we have many misspelled words and words that only appear once in the whole dataset. We will address this in the Challenges section below.
8. Once the data is cleaned and processed as such, we need to create the two data frames. We will merge the humorous and non_humorous dataframes.
9. Then in the first new dataframe ('merged_df') we will convert each word to a row, count the number of humorous and non_humorous occurrences, the total occurrences, and ratios of each.
10. The second data frame ('combined') will take the original combined dataframe, leaving the sentences in place. This data frame will also contain the word count for each sentence and average word length. Here we are only focusing on the cleaned question strings from the 'clean > 3' column as to have continuity across. This dataframe will also have sentiment scoring by using the NLTK library's Sentiment Intensity Analyzer.
11. Then we will take the top 20 humorous and the top 20 non-humorous words and turn those is to columns in the 'combined' data frame. Iteratively, if a word appears, then we increment that row and column by one.
12. Exploratory visualizations will be performed to get a better grasp on the data. (See below)
13. We will run KMeans clustering from the Sklearn library on the 'combined' data frame with two clusters.
14. We will run a Random Forest Regression Ensemble from the Sklearn library on 'combined' data frame using the boolean label column as the predictive column.
15. After running the models, we will compare their efficiencies and effectiveness which will be discussed in the evaluation and conclusion sections.

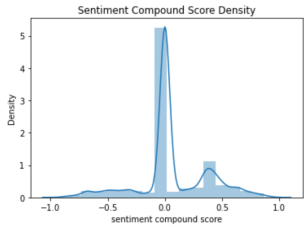
With the cleaned data, we can quickly visualize it to have a better idea of what data we are working with.

From this first scatter plot, we can see that humorous and non-humorous words have roughly the same occurrences. We notice that most words do not appear more than 75 times which makes sense because the English language has so many words to choose from but a smaller number of commonly used words. This is about what we expected given the data set.

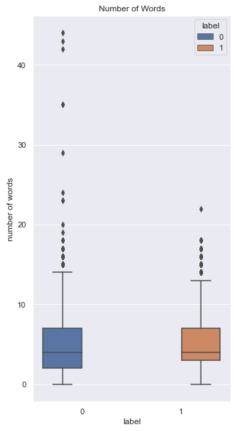


Next we look at the sentiment scores. The sentiment score tells us, based on the sentence, if the sentence tends toward positive or negative sentiment. Here we used the compound score to get a better view of the sentiment

overall per each sentence. We see that for this data set, most of the sentiment analysis tends towards 0.0, or neutral. We see a small spike in the positive score around 0.4 suggesting there could be some connection between sentiment score and outcome.



In the box plot, we can see how many humorous (denoted by “1”) and non-humorous (“0”) questions there are in the data set compared with the number of words in the question. It is interesting that the non-humorous label has many more outliers with significantly more words in the question but overall humorous and non-humorous questions tend to have about the same number of words.



In this project our focus is to know if we are able to correctly identify humorous and non-humorous product questions so that companies may better their customer service and products appropriately. To do this, we will ask two questions: “What, if any, words play a role in determining the outcome of humor or not?” and “Does the number of incorrectly typed or slang words play a role in the outcome?”

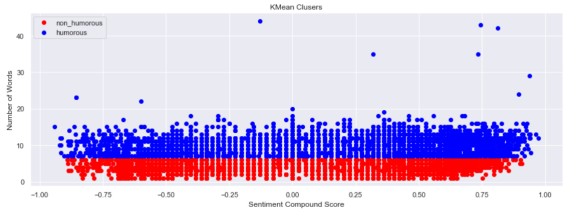
EVALUATION

After visualizing the data we have to work with, we are then able to start evaluating and running some models. For this project, we will start with two different models: linear regression and KMeans clustering. For the models, we will use sklearn libraries.

The next model performed was KMeans clustering which is an unsupervised model from the sklearn library. For this model we used the dataframe which contained sentences and sentiments (“combined” dataframe). We used 2 clusters and the following columns: label, number of words, sentiment compound score, avg word length. These columns we put into a new dataframe to make analysis faster and easier to understand (dataframe “kmeans_df”).. Two clusters were decided because we knew already that we have just two labels, 0 or 1 (non_humorous or humorous). Since we had the labels, we can do a check to see how well our model does given the data set.

From the confusion matrix, we found that 7014 non-humorous questions and 2487 humorous questions were correctly identified. However, we also see that about the same amount was misclassified. This information about the model is further shown by the precision for both humorous and non_humorous as 0.50. Through recall, we see that 74% of non_humorous questions were correctly identified while only 26% of humorous questions were correctly identified. Further, the accuracy is at just 50% overall, suggesting a poor model.

	precision	recall	f1-score	support
0	0.50	0.74	0.59	9532
1	0.50	0.26	0.34	9545



These two cluster plots for the Kmeans cluster model demonstrates that although the model is able to distinguish the clusters well when given sentiment and number of words, it contrasts that when given sentiment and average word length. This model is a good start but further modeling must be done to get better results.

	precision	recall	f1-score	support
0	0.50	0.74	0.59	9532
1	0.50	0.26	0.34	9545
accuracy			0.50	19077
macro avg	0.50	0.50	0.47	19077
weighted avg	0.50	0.50	0.47	19077

This model is not very dependable with only 50% accuracy and 50% precision for both labels.

We did the KMeans model twice, first (above) not including the most common word columns, hoping for a simple model for the problem. After the results we decided to rerun the model with the word columns.

	precision	recall	f1-score	support
0	0.50	0.26	0.35	9532
1	0.50	0.74	0.60	9545
accuracy			0.50	19077
macro avg	0.50	0.50	0.47	19077
weighted avg	0.50	0.50	0.47	19077

Although the f1-score and recall changed, we still see just a 50% accuracy and precision for all labels. Therefore we will move on to our next model.

The next model we performed is the sklearn Random Forest Regression ensemble. Using the same data frame, we ran a 80/20 train-test split to run this model. We also ran a baseline using a constant of 0.5 in a new column which represents the average of the label column (because every question is either humorous 1, or non-humorous 0, we can create a baseline by using 0.5). Our baseline error is 0.499. If our Mean Absolute Error (MAE) is less than this, we can accept our model as sufficiently good. Indeed, after running the model, we reach a MAE of 0.3782 degrees. This suggests that the model is acceptable at predicting the outcomes.

Because this model includes a random forest, it is immensely large. However, we still wanted to visualize it to get an idea of how it is represented. Here is one tree:

We can see the tree is very deep, which is due to the number of columns presented from the data frame and this image clarifies its complexity.

From our two models presented, we quickly find that the random forest regression is a stronger model for this dataset. However, although the random forest regression ensemble may be more effective, it is much less efficient. Running a complex forest on large datasets could decrease efficiency, to which it is dependent upon the company and their limitations as to whether or not this is the best model for them.

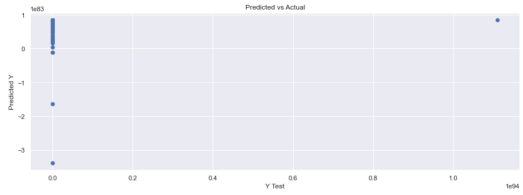
DISCUSSION

Timeline: The project is on schedule and has completed all steps of the timeline. However, as discussed below, it is important to note that the work is not done yet as more research is suggested and more models should be run.

Challenges: This project presents many challenges. There are many misspelled words (‘cradl’, ‘fone’), combined words (‘handshake’), poorly typed words (‘thisisoneword’), fantasy words (‘Ouija’, ‘zozo’), and slang (‘lmao’, ‘omg’). Misspelled words could quickly be taken care of with some NLP libraries but for now we will leave the misspellings to see if perhaps that is part of the connection between label and word. The same thinking goes towards poorly typed words, fantasy words, and slang. We didn’t want to incorrectly “correct” any words or return no value, which is an error in NLP libraries. For example, “ur” could be corrected to “or”, instead of the correct “your” version. For this reason we decide to leave these errors in place.

We initially tried training the data with a 80-20 train-test split for the linear regression model using the “merged_df” dataset (the data frame that contains each word as row). Here, we were trying to predict the word based on the number of humorous and non_humorous occurrences. However, because a string form word cannot be run in a model such as this, we decided to first convert each word to its bit-form. For example the string “hey” would be converted to 110100001100101011111001. After running the model we take a look at the results.

	Coefficient
humorous	-3.803821e+80
non_humorous	-1.378596e+80



After running the coefficient and plotting the results, it is noted that perhaps using bits and words in this fashion with a linear regression model is not actually the best way to model this data. This is because the values do not actually connect to usable and understandable data. Therefore this part of the study was removed.

Another challenge was that we saw no change from the two KMeans models. Surprisingly, using frequent word data did not significantly impact the model.

Lastly, the efficiency of the random forest is not ideal in all situations. Hopefully the random forest regression model can be improved to run faster or else a different model entirely may be necessary.

Foreseeable steps: Further research is needed. We suggest more research projects to expand upon this study. Perhaps using a SGD classifier or Naive Bayes models would work better. In addition, the third dataset, ‘non_humorous_biased’ was not used in this study due to time constraints. It would be interesting to run the best model on this dataset to test the effectiveness of it. Or even train the dataset with the ‘non_humorous_biased’ and compare the outcome of the model.

Questions: There are many questions that arise in this project. Specifically,

1. Does upper or lowercase letters affect the model?

2. Does punctuation affect the model?
3. Does word frequency impact the result?
4. Does the type of word effect the outcome (ie, noun, verb, adjective, etc)?

Although these are great questions, our project will not be able to reach all of them. Outside projects are recommended to be able to touch upon all these questions and more.

CONCLUSION

This study is one step out of many future steps in language processing for online reviews and product questions. Humans often struggle to determine tone and emotions via online messaging or text. For this reason, we can expect that computers also struggle to learn these sentiments as well and be able to determine feelings based solely on typed words. The project aims to bridge that gap by taking online product questions and designing a model that can determine humor or sincerity in text.

Our study has found that the best model for this dataset is a Random Forest Regression ensemble. It had a mean absolute error lower than the baseline suggesting a viable model. We suggest that other models are tried in future studies as well such as SGD classifier or Naive Bayes models. We also found that a multiple linear regression model and Kmeans cluster model were not apt for this data set.

Possible work related to this project are as such: NLP of texts, product reviews, social media posts, and emails. Being able to comb through and identify what your clients, followers and/or customers truly are trying to convey is powerful to any company and being. This area of study is just beginning and as humans become more dependent on online communication, it is essential that we develop these preliminary models now and plan ahead for future developments.

REFERENCES

[1] Arielle Padres. 2018. Emoji: The complete history | wired. (February 2018). Retrieved January 02, 2022 from https://www.wired.com/story/guide-emoji/

[2] Scribendi Inc. Miscommunication: The problem with texting. Retrieved January 28, 2022 from https://www.scribendi.com/academy/articles/miscommunication_and_texting.en.html

[3] Sag I.A., Baldwin T., Bond F., Copestake A., Flickinger D. (2002) Multiword Expressions: A Pain in the Neck for NLP. In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2002. Lecture Notes in Computer Science, vol 2276. Springer, Berlin, Heidelberg. Retrieved January 02, 2022 from https://doi.org/10.1007/3-540-45715-1_1

[4] T. Ghorpade and L. Ragha, "Featured based sentiment classification for hotel reviews using NLP and Bayesian classification," *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, 2012, pp. 1-5, doi: 10.1109/ICCICT.2012.6398136.

[5] Yftah Ziser, Elad Kravi, and David Carmel. 2020. Humor Detection in Product Question Answering Systems. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (July 2020), 519–528. DOI:http://dx.doi.org/10.1145/3397271

[6] Yftah Ziser, Elad Kravi, and David Carmel. 2020. Humor Detection from Product Question Answering Systems. (2020). Retrieved January 05, 2022 from https://registry.opendata.aws/humor-detection/

[7] Francesco Barbieri and Horacio Saggion. 2014. Automatic Detection of Irony and Humour in Twitter. *ICCC* (2014), 155–162. Retrieved January 03, 2022 from http://computationalcreativity.net/iccc2014/wp-content/uploads/2014/06/9.2_Barbieri.pdf