Age Classification for Chatroom Messages

Maribeth Rauh
Jennifer Long

CSE 40657
May 5, 2015

Goals

Our project will attempt to identify the age of a person based on their writing and potentially their speech. We are aiming to do so on the assumption that vocabulary and sentence structure evolve as a person ages and that this evolution is similar for most people. For example, the vocabulary and sentence structure of a ten year old differs from that of a forty year old, and we assume that there is a common difference between all ten year olds and all forty year olds. This difference may be in the topics discussed, types of words used, complexity and structure of their sentences, and other indicators we will test. The inputs will be text from a chatroom conversation written by people of varying ages, and the output will be a classification of the users involved in the conversation by age.

If successful, our project has multiple applications. One of the more direct results is that since we are analyzing chatroom text, this could be used as a safety feature to prevent significantly older people from claiming they are young on online chatrooms. Internet safety is an important topic, and when the only information available about a person in a chatroom is their written text, it can be very valuable to analyze it. If these methods were used on a different set of data, such as student essays, it could be used as a new benchmark to measure a person's development, in addition to more typical measures such as education level or literacy. This may apply to schools that are seeking to measure their performance or their students' progress as well as to nations attempting to measure population's development. The government may also use our project to better sift through the massive amounts of data they collect to gain deeper insights. Finally, companies that rely on ads for revenue could use this as another way of more accurately targeting their ads. Since each user may have a body of text associated with them, as on Facebook for example, it could be used to show specific users better ads. However, media such as blogs or articles could also be analyzed to determine what audience they are targeting so better ads can be displayed alongside them.

Methods

In order to accomplish our goal of age classification, we chose to look separately at the vocabulary used and the grammar in which these words are used, since that distinction is made in most of the existing methods. We provide below a brief overview of each method and the rationale for testing this method.

Vocabulary Methods

In order to analyze the vocabulary of different age groups, we treated each post as a bag of words and used both naive Bayes and logistic regression models to analyze this. The intuition behind using a bag of words model is that we would expect different age groups to use different sets of words, even if we ignore the order of words and how these words are used. For example, we generally assume that as a person gets older, their vocabulary expands, so we might expect older people to use longer and more complicated words. These patterns may be different because we are using chatroom data, but either way, our bag of words model will pick up on these trends. We may also assume that people of different age groups have different interests and therefore talk about different things, and our bag of words models can use this information as well.

We chose to test both the naive Bayes model and the logistic regression model. Naive Bayes uses a somewhat simple routine of counting and dividing, whereas logistic regression works through multiple iterations of the training data to find optimal weights for each word. Our hope is that logistic regression with therefore be slightly more accurate on our testing data than naive Bayes.

Sentence Structure Methods

Just as we'd expect word choice to evolve as a person ages, we'd also expect that their grammar and sentence structure would change. We would intuitively expect that older people have more elaborate sentence structures, for example. In order to test this, we looked at the sequence of part-of-speech tags for each post and placed n-grams of this tag sequence into a naive Bayes model. This is therefore designed to pick up on common part-of-speech patterns for each age group.

Since our goal is to classify unannotated text, this also required that we create a basic part-of-speech tagger, which we treated as a sequence labeling problem. We used a finite automaton and the Viterbi algorithm to construct the most likely tag sequence for a given post. Then, the trigrams and 4-grams of this tag sequence are fed into a naive Bayes model in order to classify the post into an age group.

Methods for Combining Vocabulary and Sentence Structure Scores

Since vocabulary and sentence structure can both be useful predictors of age, we needed a way to combine these two scores to come up with an overall classification. The first method we tried was to combine the two separate bag-of-words models (vocabulary and tag n-grams) into one model, such that we treated both the words and the part-of-speech tag n-grams as features. We also combined the weights using a linear regression model and then classified based on the predicted age.

Existing Approaches

This is a hot topic in NLP research, and there have been a few recent papers using natural language processing techniques to categorize writers into age groups, focusing on different types of text. One paper from Antwerp University analyzed Dutch chat conversations and obtained an accuracy of up to 85% by using a bag of words model with unigrams, bigrams, and trigrams, and they noted that adding more linguistic analysis, such as part of speech tagging, could be beneficial. A 2006 study by Schler et al studied age classification of blogs, and they found that language is closely tied to age; according to this study, older writers tend to use more prepositions and articles, while younger writers tend to use more pronouns and “assent/negation” words. This study also noted that younger writers tend to use more “blog words,” which intuitively makes sense and will probably be seen in our chatroom data set, as well. This study also studied the effects of gender on writing style, and interestingly enough, they found that many of the same words and writing styles that distinguished male writers from female writers also distinguished old from young writers. This study and many others draw a distinction between style-based features (writing style) and content-based features (vocabulary).

Argamon et al performed a similar study on authorship profiling, using Bayesian Multinomial Regression as the learning algorithm. This study found that determiners and prepositions were indicative of older writers, and contractions without apostrophes was the strongest signal of younger writers. They also found that teenagers had a distinctive writing style in comparison to 20s and 30s, and thus it was much more difficult to tell the difference between 20s and 30s.

A thesis from the Naval Postgraduate School by Jenny K. Tam tested age prediction using Naive Bayes and Support Vector Machines. The study found that Support Vector Machines performed slightly better, and both models performed best when using trigrams. This study, along with others we surveyed, limited the number of features to 1000 so as to only pinpoint the most distinctive features.

These previous papers offer us a good starting point. Most of them considered the sentence structure and parts of speech separately from the vocabulary, which is currently our intended method. Most of these studies focused on both age and other demographic features, especially gender, so by narrowing our view to age, we can hopefully get better results. Our project is distinct from other studies done previously because the data set is only chatroom conversations, which is generally more concise than most other forms of writing used in NLP research and has different characteristics.

Experiments

Data

We used the data from NPS Internet Chatroom Conversations (https://catalog.ldc.upenn.edu/LDC2010T05, downloaded from http://www.nltk.org/nltk_data/packages/corpora/nps_chat.zip). This data is organized into XML files of chatroom conversations from different days in different age groups. We determined that the important fields were the text of the actual post, part of speech tags for each word, and the part of conversation (eg. statement, emotion, greet). We pulled the words of the post from the XML, then appended the part of speech tag and part of conversation tag for each word, separated by slashes. We grouped the files into different age groups and then labeled each line with the age, then combined the labeled lines into one document with all of our data.

We chose to ignore all posts with “System” labeled as their conversation part because these were actions by the chatroom, such as new users joining, that did not contain actual text from users. We also chose to replace the user names with a generic “USER,” since new users appear all the time and therefore the user names are very rarely repeated, which doesn’t give us very useful information. We thought it would potentially be useful to pick up on how often different age groups use usernames at all. We chose to remove the “adult” chatroom posts because there wasn’t a specific age for this chatroom. The other four classifications each represent a period of 10 years (teens, 20s, 30s, and 40s) and having data that was less specific would be very difficult to deal with. Our final document of data contains lines that represent each chatroom post labeled 10, 20, 30, or 40, followed by the words of the post with the part of speech and part of conversation labeled.

Success Metric

We will measure the success of our model by how well it classifies the text by age group. The model classifies on a line by line basis. Because all of our data is labeled, we will check the classes chosen by the model against the correct classes and use this as our measure of accuracy.

Baseline

Our original baseline method uses a random number generator to guess the class of each line of text. This is the simplest and least accurate baseline possible without purposely biasing the model away from certain classes. Because our original baseline randomly selects classes, its accuracy averaged 25.11% across seven runs. The highest accuracy seen was 27.63%, and the lowest was 23.22%. This meets expectations since the model could classify lines of text from dataset into one of four age groups.

Our new baseline method performs only slightly better than random because the maximum word length of a line does not seem to be a very strong predictor of the age group. That said, the fact that it gave any sort of improvement tells us that the maximum word length may be a valuable characteristic to include in our final model.

Results and Discussion

Naive Bayes Model for Vocabulary

The first method we decided to implement was a basic bag of words model using Naive Bayes. For each category (teens, 20s, 30s, and 40s), each line from the training data is read and processed, and each word is parsed. When a word is processed, we increment the count for that word in that particular category, increment the number of total words for that category, and add the word to the vocabulary set. Once all of the words have been counted, we calculate the conditional probability p(k|d) by dividing the word count for a particular word and category by the total count of words for that category. I implemented add-one smoothing in order to account for unknown words, so a delta value of 1 is added to the numerator, and the vocabulary size is added to the denominator. I also calculated the probability that an unknown word would receive in each category, which is the same as the formula above, given that the word count for that word is zero.

Then, the testing data is read in line by line. For each line, we calculate the logarithmic probability of generating that sentence for each category and then choose the category that gives the greatest probability. The probability of generating the input sentence for the given category is the sum of the logarithmic probabilities for each word. The category that our model chooses is then compared to the actual category, and we keep track of the number of correct predictions versus the number of incorrect predictions.

This method represented our initial test for how a bag of words model performs in our age classification task. We reached an accuracy of 59.6% for unigrams and 60.6% for bigrams when classifying among four age groups, which is a huge improvement from random guessing (25%) and our word length-based baseline (29.8%). This indicates that it is worth using a bag of words model as part of our final composite model. We were very pleasantly surprised by how much improvement this simple model provided.

Logistic Regression Model for Vocabulary

In addition to naive Bayes, we implemented logistic regression to try another method of analyzing vocabulary. Logistic regression iterates over the training data, tweaking the weights for each word repeatedly, fine tuning the model in a way that naive Bayes does not. It also provides greater sensitivity since the scores it generates do not have to sum to one.

The model uses a weight of 0.08 to increment and decrement the scores for each word for each class and runs through training 25 times. We experimented with both of these parameters to optimize them. When using cross-validation, these were ideal, with an accuracy of about 52.5%. However, when we used a fixed set of training and testing data, the ideal weight was 0.09 and obtained an accuracy of 60.2%. We also implemented regularization to ensure that we were not overfitting the data, with a factor of 0.015. With a fairly small dataset, we felt that it was important to account for overfitting. This accuracy was comparable to the naive Bayes vocabulary classifier discussed above.

Part-of-Speech Tagger

In order to analyze the grammar patterns for each age group, we first needed a way to find the parts of speech for our unannotated input string. Luckily, our data is annotated with parts of speech, so we used this to train our tagger. We treated this as a sequence labeling problem, so we counted the occurrence of each word/tag pair and each tag bigram in order to construct our bigram hidden Markov model. Then, we found the highest probability route through the finite state automaton by using the Viterbi algorithm. The resulting output is a sequence of part-of-speech tags for the input line of text.

We tested our part-of-speech tagger by inputting our testing data and checking the output tags with the annotated tags provided in the data. Our tagger correctly predicted 84.11% of tags, which is fairly accurate, given that there are over 70 part-of-speech tags used in our corpus.

Naive Bayes Model for Tag n-Grams

Once we had built our part-of-speech tagger, we needed a way to analyze the tag sequences as they relate to the different age groups. Our solution was to place tag n-grams into a bag-of-words model, specifically a naive Bayes model for simplicity. The idea is that the model will pick up on which grammar patterns are common in each age group. We tested bigrams, trigrams, and 4-grams, and each one provided a slightly better classification accuracy. We chose to use both trigrams and 4-grams. Since many posts are only one word (and thus three tags: the starting symbol <s>, the part of speech of the single word, and the ending symbol </s>), 4-grams alone were not sufficient, which is why we chose to use both trigrams and 4-grams. Adding bigrams did not improve the classification accuracy, so we did not include tag bigrams in the final model.

Training the model follows the same count and divide procedure given above except that the inputs to the model are the various trigrams and 4-grams for the inputted tag sequence. Then, the decoding routine takes a tag sequence as its input, and it calculates the logarithmic probability that the set of trigrams and 4-grams of the inputted sequence are generated. The age classification that provides the highest probability is then chosen.

This model paired with the annotated tags (not using our own part-of-speech tagger) gave a classification accuracy of 49.8%, and the model paired with the tags we generated using the part-of-speech tagger gave an accuracy of 48.0%. It makes sense that the accuracy would be slightly lower when the inputs are the tag sequences we generated, since our part-of-speech tagger isn't perfect. Though our 48.0% classification accuracy is significantly lower than the roughly 60% accuracy given by our vocabulary models, it is still significantly higher than our baseline accuracy of 29.8%. This tells us that analyzing grammar patterns is a useful predictor of age, though it is not as useful as vocabulary.

Removing Single-Word Posts

During our final presentation, we received a suggestion to check how many single-word messages our dataset contained and to see if this could be bringing our accuracy down since they are most likely not indicative messages. 16.889% of the whole dataset was single line messages. In our fixed training data file, all_posts_train, 16.895% were single messages. The testing file, all_posts_test, was slightly less at 15.400%.

Although our dataset is somewhat small, we ran logistic regression on the data without single-word messages to test if removing them increased our accuracy. Without them, our accuracy dropped from 60.2% to 58.16%. Because of this, we decided not to test it further on our other models. This tells us that single-word posts do have some useful age information contained in them and that despite their short length, they were still being classified correctly at least part of the time.

Combined Bag-of-Words for Vocabulary and Tag n-Grams

Once we had built our separate vocabulary and sentence structure classification models, we needed a way to combine them. Both models improved upon the baseline and therefore contained valuable information. The most straightforward way to combine them was to merge the two bags-of-words (for vocabulary and part-of-speech n-grams) into one. Determining the smoothing parameters proved to be a challenge, since adding so many features seemed to dilute the information. Without smoothing or adjusting the weights, the performance decreased from our vocabulary bag-of-words model. Once we experimented with the relative weights for each word and tag n-gram, we were able to get the classification accuracy to 61.6%, which is slightly better than our best classification accuracy with only one of the models (60.6% for naive bayes on the word choice alone, with unigrams and bigrams). This slim improvement indicates that either we need a better way to combine the models, or the two models typically agree and very little new information is gained by combining them.

Linear Regression Model

Our second method for combining the vocabulary and grammar models was to mathematically combine the probabilities using linear regression. We modified each model such that they output four probabilities representing the probability that the post is of each of the four classes, and then we collected this data from each model for each line the training set. Then, we inputted these scores into R and formed various types of linear regression models.

Interestingly, this decreased the performance significantly. After testing several models (basic linear regression model, ridge regression, using LASSO to eliminate variables), the highest classification accuracy we obtained was 43.2%, which is worse than either of the models alone. We ran into some issues initially because our variables were all probabilities that summed to one for each of the two models, so our columns were not linearly independent.

After looking more closely at the data, we noticed something interesting. We had been working under the assumption that the features that our models had been learning about were linearly tied to age. For example, we assumed that if a post were written by a 40-year-old, it would be most similar to other posts by 40-year-olds and also similar to posts written by 30-year-olds. We assumed that this post would be more similar to 30s posts than to teen posts. However, after looking at the data more closely, we discovered that there are many cases for which this assumption does not hold. Table 1 provides the probabilities for the first five lines of our training data, for each age category according to the two models.

Table 1. The actual age classifications and the probabilities given by our vocabulary and part-of-speech tag n-gram models for each of the first five lines of our training data.

	Vocabulary Model Probabilities				POS Tag n-grams Model Probabilities			
Age	10	20	30	40	10	20	30	40
20	0.0612	0.2717	0.1638	0.5032	0.2491	0.2537	0.2490	0.2482
10	0.3922	0.1867	0.2845	0.1367	0.2500	0.2500	0.2500	0.2500
40	0.0793	0.2120	0.1631	0.5456	0.0918	0.1679	0.1340	0.6062
20	0.1176	0.2525	0.2442	0.3857	0.2500	0.2500	0.2500	0.2500
40	0.0172	0.2204	0.1281	0.6343	0.2293	0.2328	0.2497	0.2883

For example, for the first post, the vocabulary model indicates that there is a 50% chance that it was written by a 40-year-old and a 27% chance that it was written by a 20-year-old, but only a 16% chance that it was written by a 30-year-old. We'd expect the ordering to be 40s, then 30s, then 20s. This has some interesting implications: is it possible that each age group has different and completely separate characteristics? This could also be a product of the lack of ordering in our models.

Because of these trends, linear regression fails. We tried making a linear regression model for vocabulary alone, which also performed poorly. Linear regression fails given the trends above because it essentially gives a weighted average among the ages. For the line described above, a weighted average between the top two age categories (40 and 20) would give a number somewhere in the 30s, which is incorrect. Thus, we determined that linear regression was unsuited to this task.

Cross Validation

Most of our models were run on a fixed training and testing set. However, we chose to implement k-fold cross validation as a proof of concept. Were we to continue this project to the future, we would implement this for every model as this is a more robust way of of training and testing, especially with a smaller dataset.

The cross validation script shuffles the entire dataset and divides it into 10 equally sized files. Logistic regression is then run 10 times, using one of these files as the testing data and the rest of the files as training data. Because the dataset is shuffled randomly each time the script is run, the accuracy varies some, but it tends to fall around 52.5%. This is lower than the accuracy we obtained with the fixed training and testing data because we were able to optimize the model's parameters specifically to that division of the data, essentially overfitting. Cross validation gives us a more accurate representation of how our model performs.

Conclusion

One important conclusion that we came to is that this 4-way classification is a very difficult task. Since the posts are often very short, there may be very little data for our models to use when classifying. We hypothesize that using our methods on other types of writing, such as classroom essays or blog posts, might give more reliable results because there is more text to work off of. One future modification that we may make is to change our task to classify the age of each user, given each of their posts. This would provide more lines of text per classification decision and may be significantly easier.

Given that this is a difficult task, we were very pleasantly surprised by how well the basic bag of words model for vocabulary worked. Our results show that different age groups definitely use different sets of words, and even using a simple framework like naive Bayes gave us dramatic improvements over the baseline. Analyzing the grammar patterns of each age group revealed some useful information, but it did not improve the classification accuracy as much as using only vocabulary did. Combining these two models proved to be a challenge: we gained a modest 1% in accuracy when combining the two bags of words into one, which indicates that the information provided by the two models may overlap. We attempted to combine these models using linear regression, but the characteristics of each age group turned out to be less linear than we were expecting. It turns out that we can't treat age as a spectrum; we are better off treating our age bins as separate and unordered.

Roles

We split up the methods as follows:

Maribeth: Baseline, logistic regression for vocabulary, removing single word posts, cross validation
Jen: Naive Bayes vocabulary model, part of speech tagger, tag n-grams model, combined bag of words, linear regression

Each of us wrote the experiments section for the methods we created, and we collaborated on the other sections of the report.