Text Analysis of Online Dating Profiles

James Fung & Christo Sims
{jgf@eecs, christo@sims}.berkeley.edu
12.13.06

I256: Applied Natural Language Processing
Instructor: Marti Hearst
UC: Berkeley

Introduction

When a user participates in an online matchmaking website, they must create a profile to describe themselves. The profile provides a way for others to assess whether or not they may be a good match for their dating goals. Profiles typically consist of one or more photos of the individual, their answers to a series of multiple choice questions, and a section of text where they describe themselves and what they're looking for.

For our final project, we wanted to investigate patterns of language use in this latter textual portion of these personal profiles. We were curious to see whether NLP techniques would be able to meaningfully classify profiles based on textual information alone. Instead of relying on unsupervised clustering, we used the users' self-reported information from the multiple choice questions to create classes of profiles for training. These classes include, but are not limited to: gender, education level, ethnicity, religious practice, and their desire to have children. This approach allows us to explore the degree to which text profiles could reveal other characteristics about the individuals. What follows is a description of our approach and then a report on our findings. We end with a discussion of why we believe we got the results we did.

Approach

We began by collecting 400 profiles from Yahoo! Personals. We sampled 200 male profiles and 200 female profiles, all between the ages of 25 and 35 within 50 miles of San Francisco.

Christo then preprocessed these profiles to extract the text descriptions as well as the user's answers to the predetermined questions required by Yahoo! To accomplish this preprocessing, he modified a Python toolkit for parsing HTML and XML called Beautiful Soup.[1]

This parsed data was then passed to a version of the weka.py file distributed for Assignment 4 modified by James. This file extracts features from parsed data files and outputs an .arff file that can be read by the Weka toolkit. Our initial version extracted unigram frequency (TF and TF.IDF) and bigram frequency features from each profile with headline tokens given extra weight.[2] In our first pass at feature selection, the top 200 most frequent words and 200 most frequent bigrams were selected as features. The goal was to see if a user's word choice was related to their personality and background.

Additionally, James coded seven "readability" measures describing the complexity of the writing and ease of comprehension.[3] Noting that all of these measures were based off of ratios of the certain quantities – the number of characters, syllables, words, complex words, and sentences – he also implemented each of these quantities and selected ratios as features. The reason for doing so was to see if writing style provided any additional useful information to the learning algorithm.

---

[1] http://www.crummy.com/software/BeautifulSoup/

[2] In Yahoo! Personals, the "headline" is the short line that appears next a users photo in search result screens and as the title for the user's page.

[3] These seven measures were: Automated Readability Index, the Coleman-Liau Index, the Flesch-Kincaid Grade Level, the Flesch-Kincaid Reading Ease, the Gunning-Fog Index, the Linsear Write, and the Simple Measure of Gobbledygook Index.

A few profiles foiled these readability measures with unexpected formatting. Primarily, they refused to use periods – either because they were writing a list of interests and personality traits or because declined to use any punctuation – and so it became very difficult segment the text into sentences, which threw off many of the readability measures. For these outliers, James' modified code simply omitted their readability measures in the features data files.

We used this Python code to produce an .arff file for each of the possible relations we explored (e.g. gender, education level). We then used Weka to select features and apply classification algorithms. We primarily used Chi-Squared and Information Gain measures for filtering. We also tried Subset Evaluation for some of the .arff files, which searches for a relevant feature subset while minimizing redundancy. In terms of classifiers, we tried a variety, including Naïve Bayes, Multinomial Naïve Bayes, K-Nearest Neighbors (KNN), Decision Trees (J48), and a Support Vector Machines (SVM).

This first exploratory pass produced poor results (see the Results section) causing us to reevaluate our approach. Since we felt some classes had too few instances in many of our experiments, we decided to expand our corpus and combine classes to address the data sparsity problem.

To extend the corpus, we scraped an additional 200 male profiles and 200 female profiles, between the ages of 25 and 35, bringing our total to 800 profiles. To avoid duplication, we gathered our second batch of profiles from users who live within 50 miles of Los Angeles. We are aware that this approach could affect our data and tried to take measures to compensate for any potential bias in the feature selection phase (e.g. removing features such as "san," "francisco," and so forth).

To combine classes, we looked at the distribution of instances and tried to reduce the number of classes while making the distribution more even. For the Ethnicity classification, we created three classes: Caucasian, Asian and everyone else. For Education Level, we grouped them into two classes as follows:

Previous Classes	*New Classes*
Post-Graduate College Grad	College Grad or above
Some College High School Grad Some High School	Some College or below

For the Attends Services classification, we created two classes:

Previous Classes	*New classes*
More than once a week Weekly Monthly	At least once a month
Only on holidays Rarely Never	Rarely if at all

We also created one new classification challenge, trying to differentiate between users from Los Angeles and those from San Francisco.

We also increased the number of features included in the .arff file. Instead of only extracting the top 200 most frequent words and bigrams, we extracted all words and bigrams with a frequency count of two or higher. Our strategy was to feed many features to Weka and then to rely on Weka's more robust feature selection algorithms to reduce our number of features. This approach yielded an .arff file with over 20,000 features. While such a large number of features made for lengthy file loading into Weka, lengthier feature selection, and pushed Weka's memory limitation, it provided noticeably better results for some classification challenges.

Lastly, James implemented some more features to capture the users' writing style. We noticed that, although the textual portion allowed the user much flexibility in expressing themselves in terms of word choice, length, etc., they were limited in formatting options. Thus, some users put some words in all caps for added emphasis. We handled this by weighting them in token frequency, must like we did with headline tokens.

Furthermore, some users wrote the entire text portion of their profile in all caps, presumably to make it stand out in a search. We believe this stylistic choice reflects on their persona, so we created another feature that measured what percentage of their words was all caps. If this value was over a 50%, a flag would be set signaling this profile was in all caps.

Results

While our exploration consisted of many frequent adjustments in methodology, we have grouped our efforts into two main passes.

First Pass

In our first pass, we used the 400 San Francisco profiles. We passed Weka the top 200 words, the top 200 bigrams, and a series of readability measures as features. The total set was about 600 features. What follows is a discussion of our results for each classification problem.

Education Level [James]

During our first pass, education level consisted of five classes. The classes and their distributions were as follows:

Class	*Instances*	
Post-Graduate	94	
College Grad	175	47.2%
Some College	86	
High School Grad	13	
Some High School	3	

A naïve baseline system can get 47.2% accuracy by always guessing the most common class, "College Grad." Weka's feature selection tools Chi-Squared and Information Gain identified only a few relevant features. Using these features with a Multinomial Naïve Bayes classifer, we were just barely able to beat our naïve baseline, scoring 47.4% accuracy.

Gender [Christo]

Obviously, the gender distribution was more evenly balanced between categories since we sampled our profiles to ensure it. Additionally, only having two classes for classification seemed to help given the small number of total instances, N = 400. Some of the features that Weka identified as relevant included:

"man" – 2 times as likely in a female profile
"sense" – over 2 times as likely in a female profile
"independent" – over 3 times as likely in a female profile
"loving" – over 3 times as likely in a female profile
"crazy" – over 3 as likely in a male profile
"me laugh" – over 4 times as likely in a female profile
"great sense" – over 6 as likely in a female profile

Based on this, it appears that women are more likely than men to seek someone with a great sense of humor who makes them laugh, which is not surprising. When used with various classification algorithms, these features were able to accurately classify about 69% of profiles. While these results beat our naïve baseline, they do not approach the levels we had hoped for.

Want (more) kids [Christo]

Instances in this collection could be classified into one of three possible categories:

Class	*Instances*	
Yes	202	62.3%
Not sure	105	
No	17	

The naïve baseline in this case would always guess "Yes" with 62.3% accuracy. Note that there are very few instances of "No" but quite a bit more of "Not sure." We believe this is because some users selected the non-committal response in order to not scare off

potential matches, and this is an example of users doctoring their profiles to appear closer to social expectations. Applying Weka's Subset Evaluation algorithm to our feature set only yielded 11 relevant features, including:

"games" – almost 2 times as likely in "not sure" than "yes"
"caring" – over 3 times as likely in "yes" than "not sure"
"far" – over 4 times as likely in "yes" than "not sure"

When this feature set was used with the K-Nearest Neighbor algorithm we were just able to beat our baseline, scoring 64.8% accuracy.

Other Classification Problems

None of the other potential classifications approached anything close to beating the naïve baseline. James' experiments on "Employment Status" and "Political Views" contained classes which dominated the others, namely "Full-time" for the former (75% of all instances) and "Liberal" and "Middle of the road" for the latter. "Employment Status" was even worse in that 61% of users declined to give their income range.

Christo had a similar experience with his experiments. In the case of "Attend Services," presumably a measure of how religious someone is, the categories of "Never" and "Rarely" account for nearly 200 of the 290 of the people who answered the question:

Class	*Instances*
More than once a week	5
Weekly	40
Monthly	22
Only on holidays	26
Rarely	109
Never	90

"Ethinicity" was dominated by 51% Caucasian, 19% Asian, with the next highest category being under 10% and several categories with only a few instances.

In all of these cases, we could not get a classification algorithm to score a higher accuracy than could be achieved than guessing the dominant class for each instance.

Second Pass

As explained earlier, in our second pass we increased the number of instances to about 800 and increased the number of features fed to Weka. What follows is a description of our results for each classification problem.

Education Level [James]

We addressed the data sparsity problem by collapsing the number of classes down to two: "College Grad" and above and "Some College" and below. This still produced a strongly biased distribution:

Class	*Instances*	
College Grad+	494	65.7%
Some College-	258	

From the full 20,000+ features, we applied filtering to remove the irrelevant features. Using the Information Gain measure, only 325 were deemed relevant. Most of the highly ranked ones came from the readability measures, both complexity of writing (e.g. characters per word) and length (e.g. syllable count). This indicates that a user's education does have an impact on how they portray themselves online.

From this reduced feature set, we could then apply a Greedy Stepwise forward search wrapper feature selection algorithm. When used in conjunction with the Multinomial Naïve Bayes learning algorithm, we received 69.1% accuracy with over 80% F-measure

on the "College Grad+" class. However, F-measure for "Some College-" was only 22.7% due to the large number misclassified as "College Grad+." The confusion matrix for this experiment:

Classified as		
Some College-	College Grad+	
34	224	Some College-
8	486	College Grad+

This table illustrates the final feature subset of 11 features and their preferred class:

Some College-	*College Grad+*
Characters per word	"usually"
Emphasis	"enjoy"
"princess"	"current"
"honest person"	"notice"
"ask me"	
"J"	
"queen"	

Some interesting remarks can be made about these results: First, users with "Some College" education or less are more likely to use all caps to emphasize words and are more likely to use the dating meme of wanting to be "treated like a princess/queen." Second, these features are not highly discriminative since they have difficulty overcoming the large *a priori* probability of "College Grad+." Third, which sounded counterintuitive, was that users who use longer words were slightly more likely to be classified as "Some College" or lower despite the mean being lower in that class. We believe this is because the Multinomial Naïve Bayes learning algorithm used does a binary binning to determine the presence or absence of a feature and loses information in the process.

For a comparison, we used the same features with a linear kernel SVM and improved performance to 69.7%. This is startling because the wrapper algorithm selected features to optimize the performance of Multinomial Naïve Bayes. We attempted to perform a similar wrapper feature selection around the SVM but could not due to computational limitations. Looking at the feature usage in the SVM, however, a high character-to-word ratio was now more likely to be classified as "College Grad+."

Gender [Christo]

Our gender classification improved dramatically when we increased the number of instances as well as the number of features. The Information Gain feature selection measure in Weka identified 670 attributes as relevant. We then passed these features to a Multinomial Naïve Bayes classifier and got the following results:

Accuracy:	85.2%
Men F-Measure:	0.857
Women F-Measure:	0.848

Classified as		
Man seeking a Woman	Woman seeking a Man	
348	53	Man seeking a Woman
63	324	Woman seeking a Man

Interestingly, some of the most important tokens were: "children" (almost 7 times more likely in female profiles), "loving," "me laugh," "man," "honest," "guy," "original," "ladies," "women," "sweet," "herself," "older," "values," and "gal." Most of these were more common in female profiles than in male.

In comparison, when the number of features was cut to the top 1800 unigrams and bigrams, the same attribute selection and classification algorithms yielded only 65.7% accuracy. This implies that there are a large number of features relevant to the classification problem outside just the most common ones.

Want (more) kids [Christo]

For the "Want (more) kids" classification, we wanted to beat a naïve baseline of about 62%. When all 20,000+ features were fed to Weka's Subset Evaluation feature selection algorithm, a subset of 34 features was returned. This feature set was then passed to a variety of classification algorithms. The best performing was KNN, with very similar results were achieved using the SVM classifier:

Accuracy: 70.5%

Classified as			
Yes	Not sure	No	
408	1	0	Yes
160	46	0	Not sure
31	0	5	No

Some of the relevant features included: "positive person," "affection," "caring," "realize," and "lasting relationship," which are not surprising coming from someone who desires a family.

Ethnicity [Christo]

In addition to passing more instances and a larger number of features into Weka, we also collapsed several of the ethnic categories into a single category, resulting in three classes total: Caucasian, Asian, and everything else. Doing so gave us the following distribution of instances:

Class	*Instances*	
Caucasian	357	45.9%
Asian	117	
Everyone else	303	

The Information Gain feature selection measure identified 226 relevant attributes. When passed to a Multinomial Naïve Bayes classifier, we achieved 67.5% accuracy. Using SVM we achieved 63.19% accuracy.

Some of the most important features, according to the Information Gain measure, were: "lived," word count, "camping" (more popular for "Caucasians"), number of characters, number of syllables, number of complex words, "weekend," "wine" (3 times more likely in "Caucasian" than "Asian" and over 4 times more probable in "Caucasian" than the "Everyone else" category), "Chinese," "coast" (much higher for "Caucasian"), "decided," and "job" (more likely in Caucasian).

Los Angeles vs. San Francisco [Christo]

For fun, we also tried to see if NLP techniques could distinguish between LA and SF profiles. Obviously, certain features would be dead giveaways (such as "san," "francisco," "bay," "los," "angeles," etc.) so these were removed. Using Information Gain measure (220 attributes identified as relevant) and a Multinomial Naïve Bayes classifier, we were able to achieve 78% accuracy:

Classified as		
SF	LA	
258	137	San Francisco
36	357	Los Angeles

Attends Services [Chirsto]

Since most profiles attend services either "rarely" or "never" we decided to group the six categories into two: those who attend at least once a month, and those who attend at most only for holidays. Doing so still produced a skewed distribution of instances with 73.7% being in the category that rarely attends services. Information Gain identified 338 relevant features. We then passed these features to a variety of classification algorithms. KNN was unable to beat the naïve baseline so it was discarded. Both Multinomial Naïve Bayes and SVM were able to beat the naïve baseline, but not by much:

Supper Vector Machine
Accuracy: 80.2%

Classified as		
At least monthly	Rarely if ever	
50	108	At least monthly
11	432	Rarely if ever

Multinomial Naïve Bayes
Accuracy: 74.5%

Classified as		
At least monthly	Rarely if ever	
146	12	At least monthly
141	302	Rarely if ever

The SVM achieved higher accuracy ratings but was biased towards the dominant class, those who attend service rarely if ever. The Multinomial Naïve Bayes classifier had lower overall accuracy but scored a higher F-Measure for those who attend service at least monthly.

Political Views [James]

These experiments were scrubbed because Weka consistently ran out of memory when trying to perform them. Although it appears that our experiments benefited from a larger number of token features, our tools had a hard time handling all that data.

Discussion

As with our results section, we break our discussion into the two passes we made.

First Pass

We attribute our disappointing results in our first pass to one or more of five possibilities:

Explanation One: Too Small Of A Training Corpus
One likely reason for our poor results is that our training corpus (N=400) is too small. This is particularly true when the instances are split over a large number of classes (e.g. Ethnicity) or when the distribution is skewed towards one or two categories. Without a larger training set, patterns within a minority class may not emerge. This was our motivation for scraping an additional 400 profiles for the second pass.

Explanation Two: Not Enough Token-Based Features Provided to Weka
Related to our small sample size explanation is the possibility that we didn't pass Weka enough token-based features. We selected our unigram and bigram tokens based on their frequency of occurrence. In problems where the distribution of instances is biased towards one category, it is also likely that the 200 most frequent unigram and bigram features reflect that category. If this were true, then minority categories could have few, if any, token-based features in the .arff file. We saw this in the outlier case of a profile that attends services more than once a week:

> "I go to church I am very sincere in my faith and my striving to become more Like Jesus. By know means am I perfect, however, NEW MERCIES EVERYDAY! … I am a very real and straight forward person, however, HUMBLE to God's word and voice in my life. Always looking to HIM for my direction and HE is my SOURCE"

Why couldn't Weka pick up on the use of tokens such as "faith," "Jesus," "mercies," and "God?" None of these tokens made the cut initially imposed by our Python code, so for the second pass we cast a wider net and then used filtering to reduce the feature set.

Explanation Three: The Features Provided Were Only Weakly Relevant

From among the features we tried, none were highly discriminative. The classes we are trying to predict have great variety and flexibility in how the express themselves within their profile. At best, a feature may isolate a handful of instances characteristic of a subset within a particular class. However, we are hard-pressed to think of what features would perform better. Thus, it appears that, to achieve performance significantly better than the naïve baseline, we need a large number of these weakly relevant features. In the case of the readability measures, they are fairly correlated and using many of these leads to redundancy. Therefore, for our second pass, we implemented additional stylistic features to provide a different source of information to the learning algorithm.

Explanation Four: People All Write Similarly in Online Profiles

Related to the above, based on our own inductive observations of the profile text, it appears that users tend to write about the same topics and at a similar level of complexity to not deviant too far from the expected social norm. This would also explain why many features are only weakly relevant. In terms of readability measures, it is possible that profile text is generally presented at the same complexity, regardless of the educational background (or any other characteristic) of the author. In the context of dating profiles, there might be a normative level of complexity at which everyone strives to write. Similarly, different classes of people might not trend towards different word choice in their profile text. Again, there might be normative expectations to talk about certain subjects in certain ways – norms that cut across things like gender, ethnicity, and education level.

Explanation Five: Weka's Feature Selection is Not Accurate

One issue while running information theoretic measures, such as Information Gain, on continuous-valued features is how to estimate its probability distribution function from

samples. The usual method is to quantize the instances into bins and carry out the analysis as if it were a discrete-valued variable. However, we have reason to believe that Weka uses a very simplistic equal-sized binning, which is not very robust in the data sparse scenario that we have. For instance, term frequency features often only have a handful of profiles with one instance of a token and the vast majority having zero. However, Weka fails to treat this as two integer bins and appears to create many empty floating point bins in between. Similarly, for classes with few instances, many bins are also empty. This would explain why many features registered zero information gain during the first pass experiments when an eyeball analysis showed many features were relevant, if only weakly.

Second Pass

In our second pass we achieved higher accuracy ratings in many of the classification scenarios. When the number of features inputted to Weka was reduced, our classification algorithms suffered. This suggests that relevant tokens are not necessarily the ones that occur most commonly in the corpus. Using Term-Frequency or even TF.IDF to rank features in Python was not always helpful. Instead, we found it necessary to feed Weka any possible token feature and then rely on its feature selection tools to determine relevance, though this is computationally more intensive.

Increasing the number of instances also likely helped our results, especially in cases where the minority categories were too small to provide adequate training data. Similarly, collapsing categories seemed to help, although the boundaries between certain categories are not necessarily so rigid. For example, the boundary for “Attends services” could be drawn in many places – should those who only attend monthly be grouped with those who rarely, if ever, attend? When those who attend more than once a week were kept as their own category, the list of relevant features contained terms like “God,” “spirituality,” and so forth. These features were not as discriminative when the “More than once a week” class was watered down with instances from other classes.

Despite our better performance with more instances and more features, our results were not always ideal. Why couldn't we score higher accuracy ratings? We have couple theories:

First, the classification problem is not an easy one. We believe many people tend to portray themselves as one or more canonical, desirable personas, such as an active, vibrant twenty-something. As such, appearing too educated may be a handicap, which may be why the mean of the readability measures is only slightly correlated with the class. Furthermore, the textual portion of the profile gives the user much flexibility in terms of word choice in expressing themselves. Thus, specific words appear infrequently, leading to data sparsity and limiting the discriminative ability of individual tokens.

Second, the number of training examples in the corpus is small, especially for classification problems with more than two classes. Minority classes, especially, have too few instances to reliably model. Furthermore, even in the case of classification problems with only two classes, if one class dominates the other, the weakly relevant features will have a difficult time overcoming the strong prior probability. Thus, it is not too surprising that the problem where we made the most improvement relative to the naïve baseline was gender, where we had two classes of almost equal size.

Conclusion

The results of our exploratory experiments show that it is possible to extract information out of a user's textual portion of their profile with regard to their personality. After all, this is precisely the profile author's goal. However, it appears that humans are better adapted to interpret this information than computers.

The frequency of certain terms has been shown to be related to the user's class, even to the degree that it reinforces certain social stereotypes. Likewise, unsurprisingly the readability of a user's text is slightly correlated with their education level. However, individual features are only weakly relevant, either because they are so sparse (as in the case of token frequency) or their intraclass variance is large relative to the intraclass variance (as in the case of readability measures). This actually should not be too surprising as humans are complex and highly diverse creatures, and to capture that diversity with only a handful of variables would be astonishing.

What this means for natural language processing of personal profiles is that, since there are no strongly relevant features, to produce an accurate model we need many weakly relevant features, but with minimal redundancy. For token frequency features, we noted that it is not necessarily the most common tokens which are more relevant, so in future we suggest using a large battery of potential tokens and filtering out irrelevant ones. Furthermore, though the readability measures gave a different perspective of the user than token frequency features, between themselves they were fairly correlated so that only one or two sufficiently represented all the discriminative information between them. Thus one avenue of future work could be to increase the diversity of features by breadth rather than depth in an attempt to bring in different aspects of the user.

A side effect of the low discriminative ability of all features is that it is difficult to overcome issues with the corpus, namely the sparsity of training data for minority classes and the inability of weakly relevant feature to overcome large a priori probability. We showed that performance improves when the same features and methodology are applied

to binary classification problems with more balanced distributions, but that is expected. Future experiments will benefit from a larger corpus.

Finally, although Weka was a convenient tool in that already implemented many feature selection and learning algorithms, we have come to suspect what goes on under the hood, especially how and when it bins continuously-valued variables.