Text Classification of College Attendance

Eliza Evans
CS224N Final Project

Abstract
In this study, I explore the natural language characteristics of college-going choice among high school students. Having asked the students to write short answers in response to questions about college, I then 1) use various classification algorithms to predict their college attendance (or not) and 2) use latent dirichlet allocation (LDA) to explore the topics in the essays and seek patterns of topic usage among the two groups. I find that the unigram model with an added feature for essay length performs best in classifying the essays. The LDA analysis shows no discernable pattern of topic usage among the two groups of students, perhaps providing insight into the unexpectedly lower performance of some of the content-based classifiers.

Research Question
Can we predict college attendance based on the language high school students use when writing about college? In this study, I approach this question as a classification problem, in which I test different classification algorithms' ability to classify students as 'attending college' ('positive') or 'not attending college' ('negative'), based on answers they wrote in response to prompts we provided. Additionally, I ask, are there common topics used in the responses of students in each group? I investigate these topics through latent dirichlet allocation (LDA), hoping to identify word clusters and topics that are prevalent in the written responses of each group of students.

Both of these lines of questions and investigative approaches are informed by a research partnership: I am working with a college advising program that places counselors in schools with low college attendance. The counselors administered our prompt questions and sent us the student responses. We hope to identify certain language features and topics in the students' answers that are particularly predictive of attending college (or not), so we can create more targeted education plans and interventions that will help more students eventually go to college.

Literature
I follow in the methodological tradition represented by scholars such as Pang, Lee, and Vaithyanathan (2002) and Wang and Manning (2012), who demonstrate that simple, statistical algorithms can reliably perform sentiment analysis and text classification. In this project, I use the same algorithms tested by these scholars (e.g., Naïve Bayes, maximum entropy, and support vector machines) to model college-going decisions in the corpus of student essays. To date, application of these methods in the social science literature is rare. Grimmer and Stewart (2009) advocate for the use of computational linguistic models in the study of social science research questions, but there are no notable instances of this happening yet, especially in the education literature. The benefits of a computational linguistic approach are clear in this study. A single counselor may not have time or resources to meet and interview hundreds of students in a single school, and so reliable quantitative text classification of student responses may help her identify the students most in need of guidance and allow her to target her conversations and finite time.
LDA (as described in Blei, Ng, and Jordan 2003) is more often applied in the social

sciences (e.g., Ramage et al. 2009), but often in the area of borrowing and influence across academic disciplines or political groups. Topic modelling in service to educational interventions is unique to this project.

Data

The data for this analysis are a corpus of short answers, written by high school students and collected during spring semester of the 2012-2013 school year. All of the students in three different high schools wrote short answers to four prompt questions about college.[1] In total, there are 1,494 responses in the corpus. 412 of the responses are written by students not planning to attend college, while 1,082 are written by students who do plan to attend college. I remove from the corpus the responses of students (N=282) who did not write anything in response to the prompts or did not indicate their plans for college.

Dataset	(N+,N-)	Avg. Length	|V|
Student Essays	(1082, 412)	18	2434

Table 1. Descriptive Statistics of Essay Corpus

Methodology

Cleaning, Tokenization, Stemming

Following the bag of words assumption inherent in Naïve Bayes and other text classification algorithms, I group all four of a student's question responses into a single text file for the analysis, so that there is one file per student. I then clean these files by removing all punctuation, tokenizing on white space, removing stop words drawn from a list of English stop words, and then stemming the text using the Porter Stemmer algorithm[2] (Porter 1980). I preserve all spelling errors, under the hypothesis that the unique words that appear as spelling errors might be more likely to occur in college-going or non-college-going essays and provide additional predictive value for my classification algorithms.

Part I. Text Classification

For all of the baseline and n-gram classifiers, I use the nltk for Python implementation of the Naïve Bayes classifier, described as:

$$\underset{c_j \in C}{argmax}\ P(c_j) \prod_{i=1}^{n} P(f_n|c_j)$$

where $P(c_j)$ is the prior probability of a given class and the set $f_1, f_2, \ldots, f_n$ describes a feature set for a given document.

Baselines

As baseline classification algorithms, I run three different Naïve Bayes classifiers, each based on a single descriptive feature. The first is based on essay length; the second, on the essay vocabulary (i.e, the number of unique tokens in the essay); the third is a measure of 'lexical diversity' borrowed from Bird et al.'s (2007, p.9) nltk textbook. 'Lexical diversity' conditions the vocabulary size on the length of the essay. Each of these are simple baseline measures that might provide some relevant classification knowledge to the algorithm, but they provide no particular lexical content about *which* words are in a given essay.

N-gram Naïve Bayes

The n-gram models build upon the baseline models by adding features to the algorithm that represent which words occur in which essays. I implement uni-, bi-, and trigram

[1] 1) What resources are available to help you learn about college at school? At home? In other places?; 2) What are some of the difficult things you must do if you want to go to college?; 3) If you go to college, how will it affect you and your family?; 4) How would you pay for college? Discuss your plans to pay for college and how you will access the funds you plan to use.

[2] Location of source file for the Porter Stemmer: http://tartarus.org/~martin/PorterStemmer/index.html

models that have features for specific words or combinations of words.

Maximum Entropy

Maximum entropy classification (MaxEnt) is a classification algorithm that has proven to be competitive with Naïve Bayes, especially in corpuses with a smaller vocabulary (Nigam et al. 1999), which I have in my dataset. MaxEnt also does not have the same conditional independence assumptions as Naïve Bayes, which may lead it to perform better in some situations. Each feature has a weight for each class, and the probability of a given class for a given document is described as,

$$P(c|d) = \frac{exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} exp \sum_i \lambda_i f_i(c',d)}$$

In my analysis, I use the nltk for Python implementation of the MaxEnt classifier.

Support Vector Machines

Support vector machines (SVM) are another model used for text classification. Rather than acting as a probabilistic model, the SVM works in hyperspace. In the case of a two-category classification problem such as mine, SVM identifies the hyperplane that divides the each class's document vectors in from each other in a way that maximizes the distance between the two classes. In my analysis, I use the scikit-learn Python implementation of SVM with all of its default settings.

Model	All	Even
Doc Length	0.727	0.548
Vocab Size	0.720	0.548
Richness	0.713	0.488
Unigram*	**0.740**	--
Bigram*	0.733	--
Trigram*	0.720	--
MaxEnt**	0.713	--
SVM	0.677	--
* N-gram models include a feature for length **1000 iterations, unigram features + vocab size 'All' dataset: Training N = 1.045; P(pos) ~ 0.72 'Even' dataset: Training N = 439; P(pos) = P(neg)		

Table 2. Classification Algorithm Accuracies on Test Set of Essays (N = 150 for 'All'; N = 63 for 'Even')

Results and Error Analysis

The classification model accuracies on the test set are listed in Table 2. Below, I go into detail about the implementation and error analysis for each model.

When I ran the baselines on the 'All' dataset containing all 1,494 essays, it seemed that the high prior probability of the 'college-going' class (P = 0.728) overwhelmed the Naïve Bayes algorithm when it only had basic, descriptive information about the essays. In nearly all cases, it chose a 'positive' tag for the essays, leading to many misclassifications of the 'negative', non-college-going essays. As a representative example, in the case of the 'length' baseline model and the dev set, the algorithm correctly classified 202 of the 216 positive test documents, but then only correctly classified 11 of the 72 negative examples. Because the prior class probabilities were almost 2:1 'positive' to 'negative,' I thought that the class priors might be overly influential in the Naïve Bayes algorithm. Given any document in the dev set of a previously unobserved length, vocab size, or lexical richness, the class prior would lead to a positive prediction.

To try to address this and see how the baseline models performed in a situation where the class priors were equal for each class, I randomly selected 314 of the positive documents and created a dataset ('Even') for which the prior P(pos) = P(neg) and reran the baseline models. While the models did correctly classify more of the negative documents, the loss in classification accuracy of the positive documents led to an overall drop in accuracy of 18 to 23 percentage points per model on the test set. I hypothesize that the reduced number of positive training examples led the

models to be less confident in the positive class predictions, leading to the overall drop in accuracy.

Having learned that the class priors were not a primary concern behind the high misclassification rates of 'negative' documents in the baselines, I moved forward in my error analysis down a different route. In all the models, it seemed that more content-specific lexical information obtained through n-gram models would improve the classification accuracy. For instance, in the length baseline model, the negative documents that were misclassified as positive may have been better classified with word-specific information: some documents contained words like "military" and "army" that should have indicated non-college-going. Lexical information could also improve the classification of positive documents. Misclassified positive documents were often very short (a common trait of the negative documents), but they contained specific references to "career" and "scholarship," words that might have a greater association with college-going, in spite of the short length of the essay. As I examined the errors in the baseline systems, I thought that identifying the presence and absence of specific words (rather than the raw count of tokens or their diversity) would improve the model performance, so I implemented three n-gram models for unigrams, bigrams, and trigrams.

For all of the n-gram models, I include the document length as a feature, since it was the best-performing of the baseline models, and I wanted to build upon this foundation. The unigram model's accuracy (0.74) is 1.3 percentage points higher than the document length baseline, based primarily upon improved classification of the positive documents, rather than the negative ones. It only misclassified two positive documents in the test set (as opposed to 6 in the length baseline), but still performed poorly on the negative documents, misclassifying 37 of the 43 negative documents. Surprisingly, some of the misclassifications of negative documents contained words that I would have expected to highly indicate the negative class. For example, one document containing "army" was still classified as college-going. I hoped that, with the inclusion of additional content information though bi- and trigram features, the models would become more accurate.

While the bi- and trigram models performed increasingly better on classifying the non-college-going/negative essays (correctly classifying 10 and 14 negative essays, respectively), overall, there was a drop in accuracy as compared to the unigram model, which I did not expect. At first, I was puzzled by this outcome, but the lower accuracies of the MaxEnt and SVM combined with insights gained through my error analysis shed light on these unexpected outcomes.

Across both groups (college-going and not), students seem to be using the same content. For example, one student used the phrase "free ride scholarship" twice in the essay, but then indicated that she was not planning to attend college. Another student wrote "military grant," but then indicated she *was* going to college. Of the 20 most common bigrams in each group of essays, 14 are the same for each group. Because of this high content overlap, MaxEnt and SVM, which both depend on *only* content, perform more poorly in classification of the essays. Because my N-gram models also account for document length, they perform better. The lower performance of the bigram and trigram models as compared to the unigram model is likely related to this same issue. It seems that the higher a model's dependence

upon content, the lower its performance. The unigram model provides just enough lexical content to improve upon the baseline without losing performance accuracy in the face of the highly overlapping content.

The topic modelling analysis below provides a perfect opportunity to explore content overlap between the two groups.

Part II. Topic Modelling

Latent Dirichlet Allocation

Latent dirichlet allocation (LDA) is a generative, probabilistic model for identifying latent topics that undergird the production of a text corpus. LDA is based on a text generation story that poses that there is a dirichlet prior over the distribution of documents and topics and also words and topics. Through co-occurring words in the text corpus, LDA identifies a user-specified *k* number of topics in the corpus. In my analysis, I use Mallet (McCallum 2002) both as a stand-alone program and in an R wrapper[3] to perform the LDA analysis. Mallet performs stopword removal, but I leave the text un-stemmed for clarity purposes in deciphering the words in each topics.

Results and Output Analysis

I started by running LDA with 200 topics but found that the keywords for each topic frequently overlapped with the keywords of other topics. I then changed to 50 topics, but the topics in this instance seemed too broad and few made sense. Finally, I settled on 100 topics (Table 3 shows a selection of topics and keywords). While many of the topic keywords grouping were nonsensical (such as topics 0, 24, and 99 in Table 3), others made sense. The 'Counselors' topic was very clearly a topic consisting of the names of the counselors at each school

Label	#	Keywords
?	0	early late brings smart passion lemonade
Family	2	sister older brothers friend grandparents uncle
Application	11	essays write applications test essay writing
Payment	12	savings account bank fair tours finance summer
?	24	sister set size info bach readily
Counselors	29	ms [R] [D] mrs [L] [P] testing [L]
Athletics	41	scholarship hope sports coaches play football savings win soccer
Support	56	parents scholarships teachers grants proud guidance friends
Extra-curricular	57	activities extracurricular cost decent participate extracuricular involved
Military	63	military join force air active
Academics	76	sat act high scores gpa score test
Funding	81	grant hope scholarship pell
?	99	hardest website screwed blessings

Table 3. Selection of LDA Topics with My Assigned Labels, Topic Number, and Keywords

(reduced to the letter of their names in the table for confidentiality). Others, like 'Athletics' and 'Military' seemed to reflect students' future plans; the topics I have labelled 'Application,' 'Funding,' and 'Academics' seemed to address topics related to school and college-going.

Using cosine similarity scores measuring the similarity of the topic usage between any given pair of essays, I construct a heat map to look for patterns of topic usage among the college-going and non-college-going groups of students.[4] I put the documents in order so that the non-college-going essays are indices 1-314 and the college-going essays are

[3] http://cran.r-project.org/web/packages/mallet/mallet.pdf

[4] I wanted to do this using only the topics that made sense, but I could not figure out how to filter the topics. I plan to implement this in the future.

indices 315-1494. I had hoped to see a heatmap that looks something like this (this is a sample dataset, exaggerated for effect):

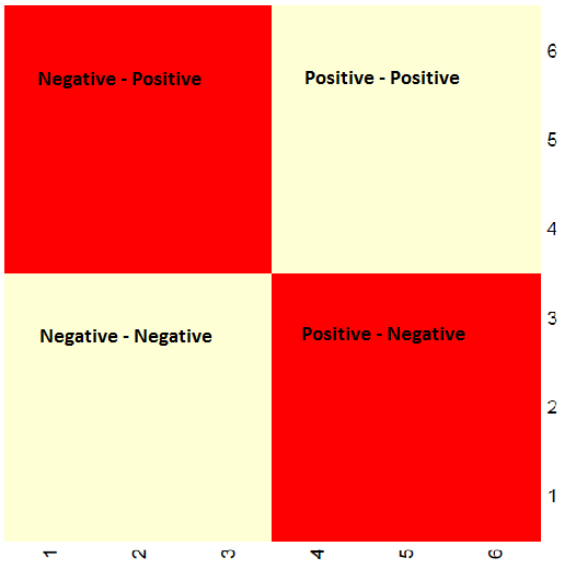

Figure 1. Fictional, Demonstrative Heatmap

In this case, negative documents (low numbers, 1-2) show high similarity to each other (white is 'hotter' than red), as do the positive documents (higher numbers, 3-5) in terms of the topics they address in their essays. Instead, my actual heatmap shows the same level of similarity across nearly all documents. Though there are scattered pairs with higher similarity, the only discernable pattern in the heatmap is the diagonal band, which indicates (as we would expect) that each document is highly similar to itself. For the purposes of space, Figure 2 is a heatmap of a selection of 50 of my documents, 25 negative and 25 positive. Its patterns reflect that of the entire heatmap, while its smaller size allows it to be more visible in this limited space. If there were patterns of similar topic usage among the positive and negative essay groups, we would expect to see warmer heatmap regions in the bottom left and top right quadrants, but there is no evidence of this kind of pattern.

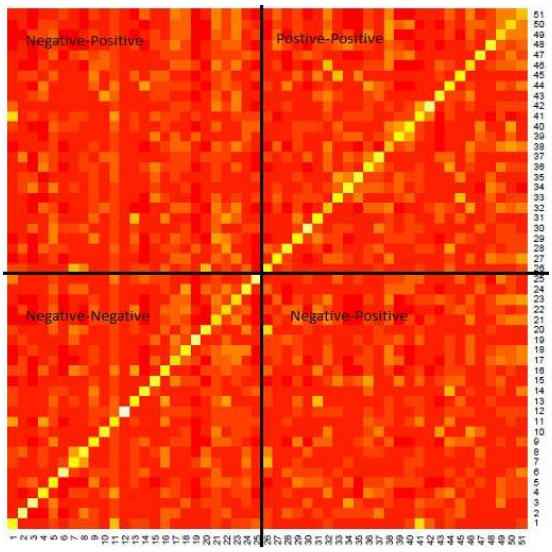

Figure 2. Heatmap Based on Cosine Similarity of 50 Sampled Essays

The results of the LDA analysis reinforce my interpretation of the classification accuracies across different models. The models that depend only on the words that students use (MaxEnt and SVM) were the poorest performers, which I suggest is because the student responses are very similar to one another, regardless of whether or not they say they are going to college. The LDA output, though indicating that there are some distinct, recognizable topics in the corpus, shows graphically this homogeneity of topics across essays.

Conclusions and Future Directions

At this point, I cannot classify student essays with enough accuracy to be of use to a counselor in a school setting, nor does the LDA analysis allow me to demonstrate coherency among the essay topics of either college-going or non-college-going students. Yet, the error analysis indicates important future steps for the project.

First, I plan to ask more open-ended questions in the next round of data collection. The questions that we used definitely guided students towards words

like ‘scholarship,’ ‘counselors,’ and ‘family.’ More open-ended questions may allow for more diversity and greater vocabulary across the corpus which would enable better classification. As an example of this principle in action, for all of the n-gram models, the unigram ‘sad’ is one of the three most informative features and strongly predicts a non-college-going tag. Looking through the corpus, I found that all of the instances of ‘sad’ occur in response to the question about how a student’s family will be affected if the student goes to college. This is our most open-ended question and produced the most variation in student responses. I think inclusion of more open-ended questions like this one would increase the variation in content in the essays and allow for both better classification and topic modelling. I hope to include “What are your plans for after high school?” and “What kind of work would you like to do?” as questions in the next round of data collection, to spark more variation in student responses and resolve some of the homogeneity that I think affected the MaxEnt, SVM, and LDA outputs.

References

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python.* Beijing: O'Reilly.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. “Latent Dirichlet Allocation.” *Journal of Machine Learning Research* 3: 993-1022.

Grimmer, Justin and Brandon M. Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis* 21(3): 267-297.

Nigam, Kamal, John Lafferty, and Andrew McCallum. 1999. “Using Maximum Entropy for Text Classification.” In *Proc. of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61-67.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. “Thumbs up? Sentiment Classification using Machine Learning Techniques.” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (ACL).*

Porter, M.F. 1980. “An Algorithm for Suffix Stripping.” *Program* **14**(3): 130−137.

Ramage, Daniel, Evan Rosen, Jason Chuang, Christopher D. Manning, and Daniel A. McFarland. 2009. “Topic Modeling for the Social Sciences.” *NIPS Workshop on Applications for Topic Models.*

Wang, Sida and Christopher D. Manning. 2012. “Baselines and Bigrams: Simple, Good Sentiment and Topic Classification.” In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.