Speech Act Classification of Video Game Data

Jeff Orkin

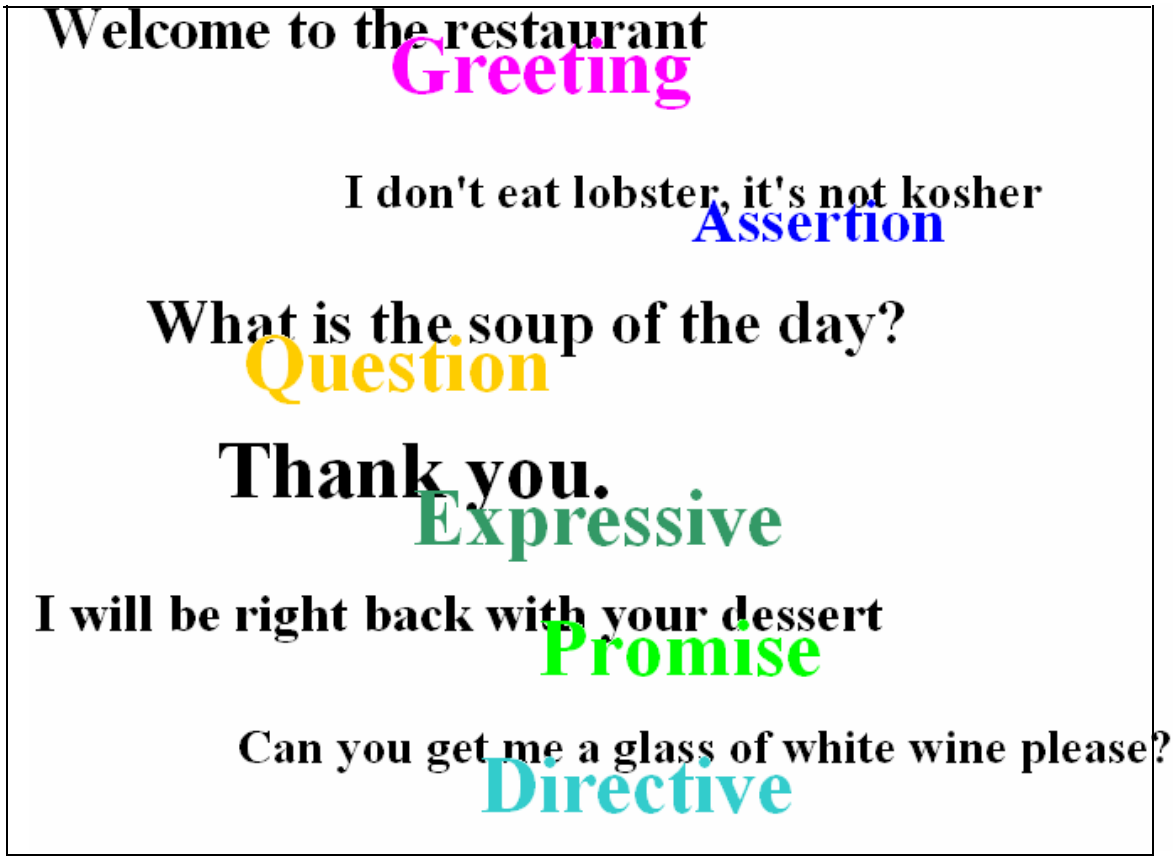
MAS.622J

Table of Contents:

Introduction

What is a Speech Act?

A speech act in the context of the restaurant game is a line of text that one player typed to another. I adapted six speech act classes from Searle's classifications that work well for labeling speech acts in the restaurant context.



About the Game

The restaurant game allows two humans to play the roles of a customer and waitress in a restaurant. Players can move around the restaurant, interact with objects, and (most importantly for this project) chat to each other with open-ended natural language text input.

Images removed due to copyright restrictions.

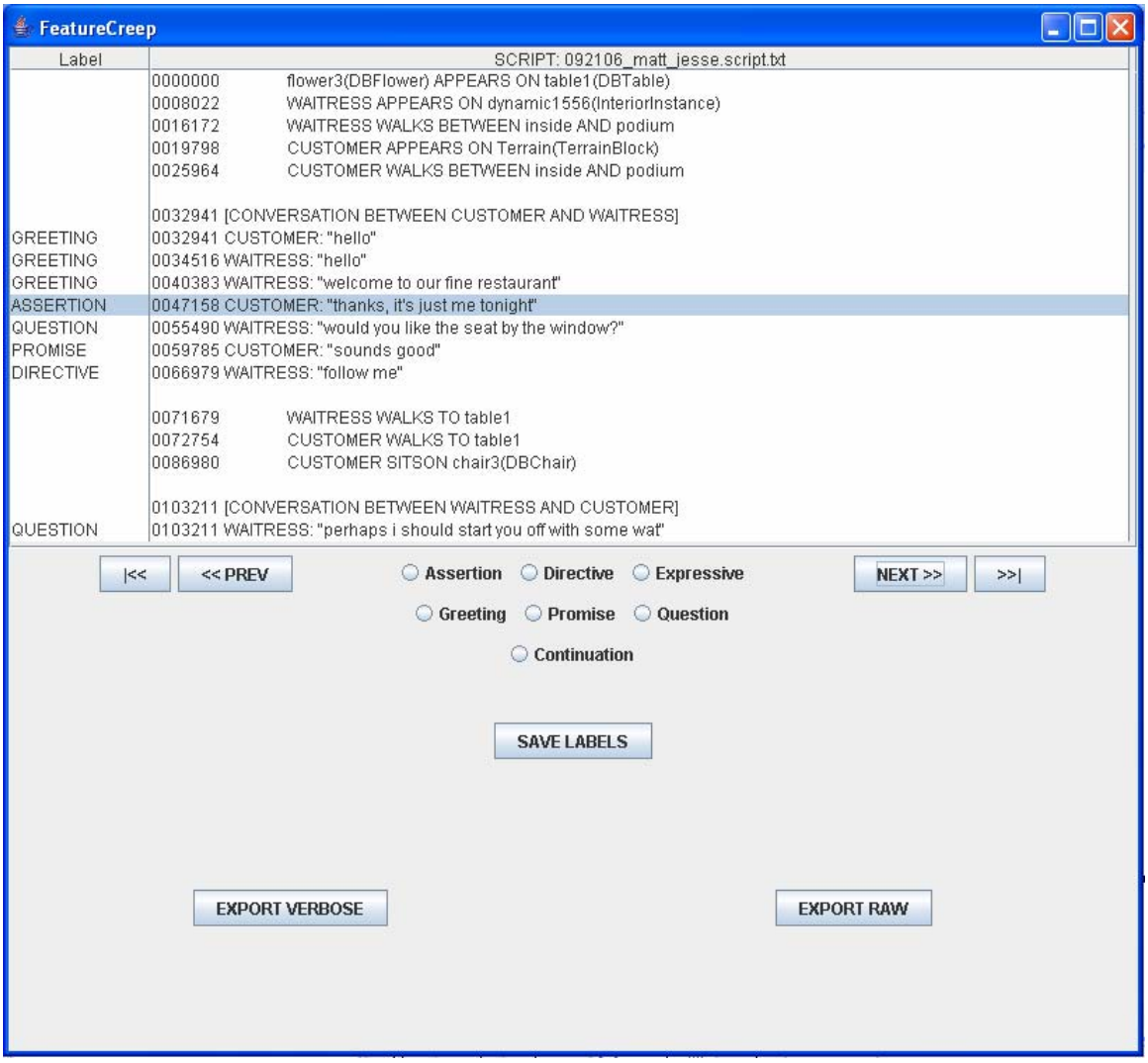
Gameplay Scripts

The game logs all actions taken by players, and all text typed to one another.Below is an excerpt from a script produced by the game. Each line of conversation is an unlabeled example of a speech act.

```
0040383 WAITRESS: "welcome to our fine restaurant"
0047158 CUSTOMER: "thanks, it's just me tonight"
0055490 WAITRESS: "would you like the seat by the window?"
0059785 CUSTOMER: "sounds good"
0066979 WAITRESS: "follow me"

0071679                 WAITRESS WALKS TO table1
0072754                 CUSTOMER WALKS TO table1
0086980                 CUSTOMER SITSON chair3(DBChair)

0103211 [CONVERSATION BETWEEN WAITRESS AND CUSTOMER]
0103211 WAITRESS: "perhaps i should start you off with some wat"
0106151 WAITRESS: "water"
0114058 CUSTOMER: "that sounds good, can i check out a menu?"
0121603 WAITRESS: "sure thing, coming right up"

0123796                 WAITRESS WALKS TO podium
0127389                 dynamic1733(DBMenu) APPEARS ON NULL
0127389                 WAITRESS PICKSUP dynamic1733(DBMenu)
0129197                 WAITRESS WALKS TO table1
0134782                 WAITRESS GIVES dynamic1733(DBMenu) TO CUSTOMER
0137059                 WAITRESS WALKS TO bar
0142659                 CUSTOMER LOOKSAT dynamic1733(DBMenu)

0148627 [CONVERSATION BETWEEN WAITRESS AND BARTENDER]
0148627 WAITRESS: "water please"

0149627                 dynamic1741(DBWater) APPEARS ON bar(DBBar)
0151876                 WAITRESS WALKS TO bar
0153818                 WAITRESS PICKSUP dynamic1741(DBWater)
0155565                 WAITRESS WALKS TO table1
```

Data Labeling and Feature Extraction

I wrote a Java program called FeatureCreep that allowed me to quickly label all 2,577 speech acts found in the 50 scripts, and export various subsets of features for different experiments. It took about three hours to label all of the data.



Why Classify Speech Acts?

I am using the restaurant game to teach AI characters how to behave and converse while playing different social roles in a restaurant by observing lots of pairs of humans play these roles. AI characters will need to learn the social norms of conversation.

Learning Social Norms of Conversation:

[**Question**] what would you like to order?
[**Directive**] may I have the spaghetti
[**Promise**] sure, coming right up.

[**Assertion**] Here you go.
[**Expressive**] thank you
[**Expressive**] You're welcome.

Things I Learned From This Project

- Mutual Information works well for selecting words to use as features.
- Speech Act Classification is a similar problem to Part of Speech Tagging.
- HMMs work well for tagging.

Speech Act Classification

In contrast to classification of sea bass and salmon, speech act classification can be difficult because the ground truth is debatable. Often a line of text may contain elements of multiple classes of speech acts. Humans may not agree on how to classify each line. Open-ended natural language input provides additional challenges, in allowing players to use words and phrases that may have never been seen in training examples.

Salmon or Sea Bass?

[**Greeting / Question**] Hello, what can I get you?
[**Expressive / Assertion / Question**] I'm sorry we don't have pizza. Would you like to see a menu?
[**Expressive / Assertion / Directive**] Hmm, the tart sounds good. Think I'll have that

People Say the Darndest Things:

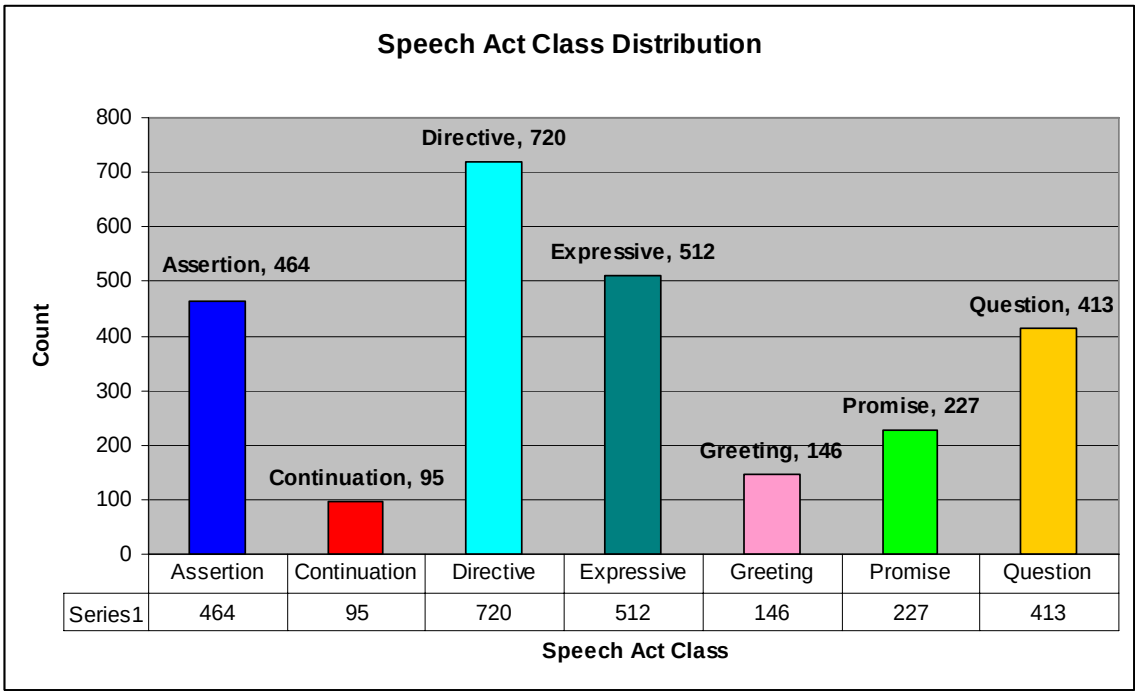
example 1: I think I'm ready to order

example 2: Here you are sir, you can pay at the register

example 3: Things haven't been the same since the cylon's arrived ,eh?

Baselines and Best Results

Below are counts of different types of speech acts found in 50 gameplay scripts. We can use these counts to calculate a baseline for comparing accuracy of different approaches.



50 Gameplay Scripts. Each script contains ~50 speech acts.
Total: 2577 Speech Acts in corpus
Baseline: Always choose Directive. Correct 720 / 2577 = **27.94%**

Comparison of Best Classification Results with Alternate Human Labels:

I had my office-mate label a portion of data (about 1/5th), and I use that for comparison too. I provided her with the guidelines that I had written for myself when initially labeling the data. I calculated both her accuracy at matching my labels, and the Kappa Statistic, which is an index which compares the agreements against that which might occur by chance (computed from the confusion matrix). The alternate annotator labeled 623 speech acts (out of 2577), and agreed with my labels 486 times. I would like to try

getting more humans to annotate more examples to get a better sense of how HMMs compare to human agreement.

Inter-Annotator Agreement: 486 / 623 = **78.01%**
My Best Classification Results: HMM with hold-one-out cross validation (holding out one script, so equivalent to 50 fold cross validation).

	Baseline	**Human Annotator**	**HMM**
Accuracy	**27.94%**	**78.01%**	**81.76%**
Kappa Statistic		**0.73**	**0.77**

$$Kappa = \frac{ObservedAgreement - ChanceAgreement}{1 - ChanceAgreement}$$

Throughout my experiments, I refer to the percent correctly classified as accuracy. As I ran experiments, I evaluated performance by looking at the accuracy, precision, recall, and confusion matrices. My results below are reported only in terms of accuracy for brevity.

Features

I had two subsets of features: text-based, and contextual (related to the physical context). For text-based features, most of the features are Boolean indicators that flag the presence or absence of some specified word (or bigram or trigram).

Text-Based Features:

- word count
- punctuation indicators
- word indicators (indicators for bigrams and trigrams of words too)

For Example: can I get you something to drink?

feature vector <wordcount, ?, !, :), :(, something, table, steak, hello, drink, car, . . . >

< 7, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, . . . >

Contextual Features:

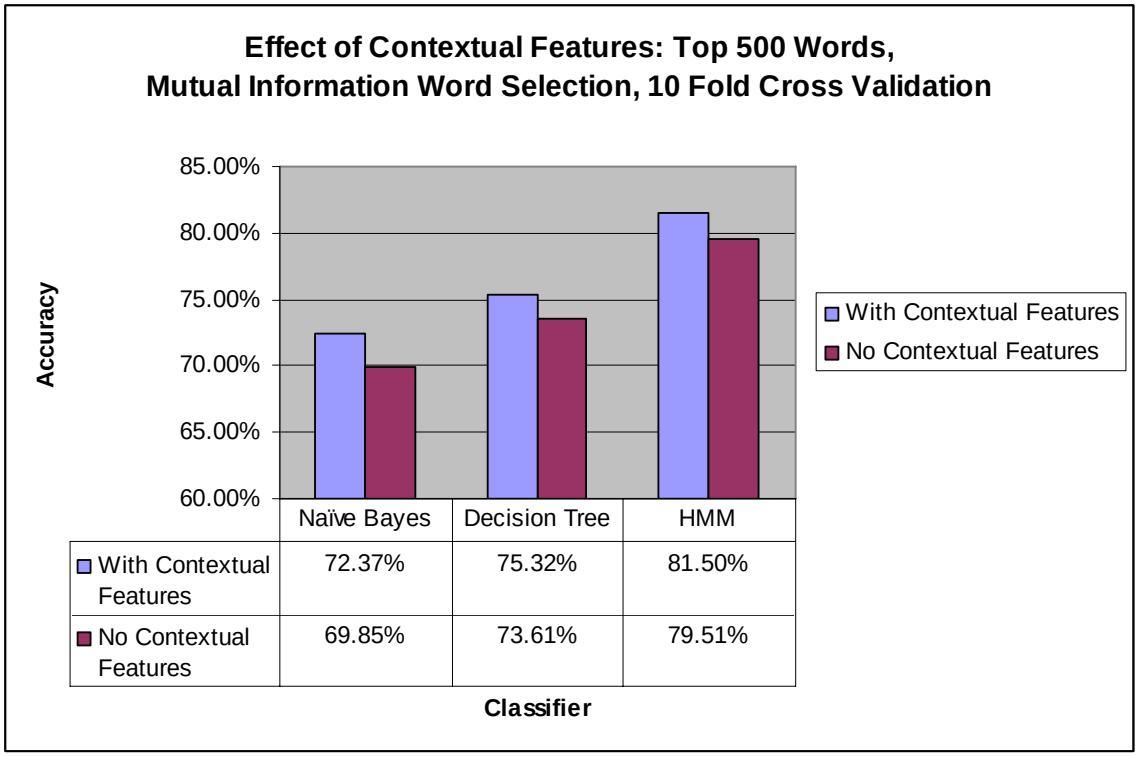
- Who is speaking?
- Who spoke last?
- Who is potentially listening?
- Is speaker sitting or standing?
- Speaker location indicators? (table, kitchen, bar, front door, inside, outside)
- Potential listeners location indicators?
- Time code of speech act.

Feature Count:

Text-Based Features: Experimented with various numbers of text features (see Word Selection section).
Contextual Features: Constant count of 32 features.
Total number of features ranged from 107 to 10,561.

Effect of Contextual Features

Below we see that including contextual features (in addition to text-based features) improves performance by approximately 2% for all classifiers that I tried.



Word Feature Selection: Probability Ratio vs Mutual Information

I experimented with exporting different numbers of indicator features for the top N most informative words per class. For each number of indicators per class, I tried two different metrics for measuring how informative a word was for a particular class: probability ratio, and mutual information. For each metric, I sorted the words by this metric once for each class, and took the top N words to export as indicator features. The term "word" here actually refers to exporting the top N unigrams, the top N bigrams, and the top N trigrams. In the graphs below we see that mutual information consistently out performs the probability ratio metric.

Metric 1: Probability Ratio

$$\text{Probability Ratio(word, Class)} = \frac{P(word \mid Class)}{P(word \mid \neg Class)}$$

The probability ratio measures how likely a word indicates some class, and does not indicate any other class.

Top 10 Indicator Words per Class Selected by Probability Ratio

CLASS	WORDS
Assertion	might ive theres tonight believe big board havent keep kitchen
Continuation	save youll case lunch salads specialty whole ghost after chair
Directive	mignon please follow enjoy lets jour du filet register cobb
Expressive	sorry oops haha tahnk thanks thank perfect apologies lol repeat
Greeting	bye hello night afternoon hi bob goodnight evening soon again
Promise	moment certainly coming clean shortly away right back minute bring
Question	anything everything finished where else miss whats wheres does perhaps

Metric 2: Mutual Information

Mutual Information(word, Class) = $P(word, Class)\log\frac{P(word, Class)}{P(word)P(Class)}$

where $P(word, Class) = P(word \mid Class)P(Class)$

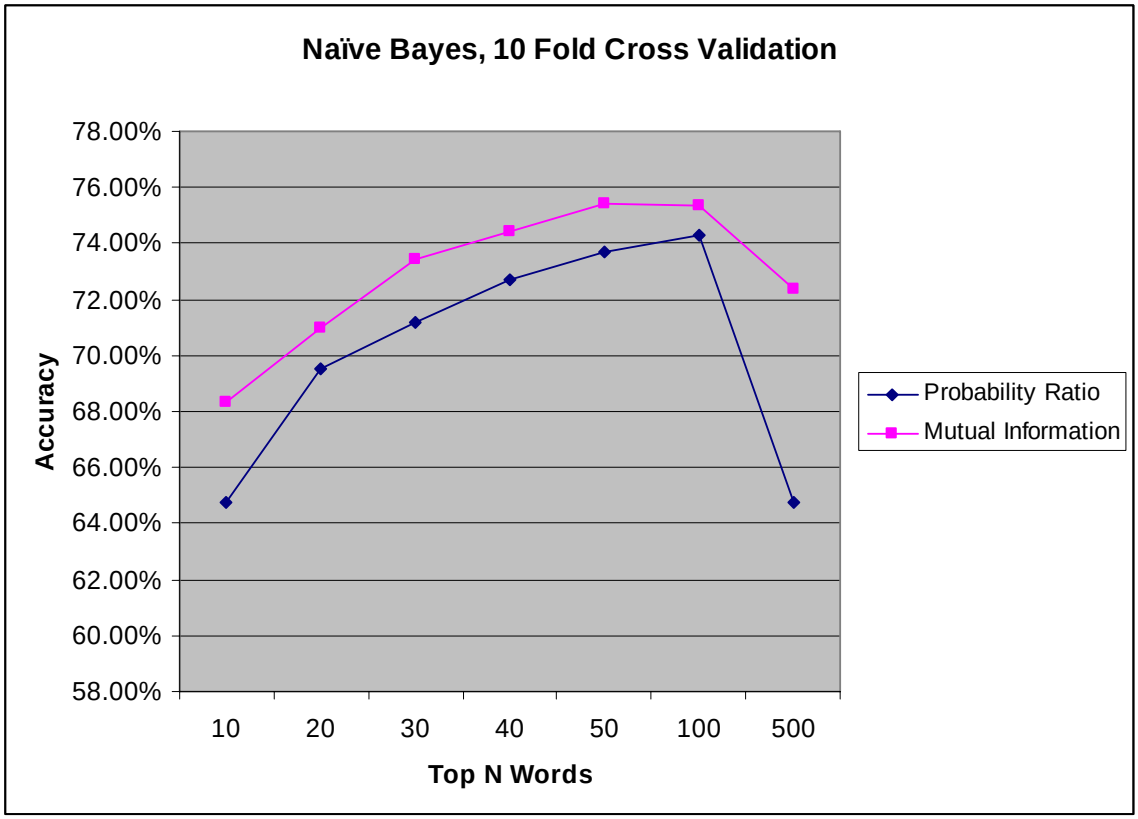
The mutual information measures how statistically dependent the word and Class are. The big difference between this metric and the probability ratio is that mutual information will pick words that are good indicators that an example is NOT of some class, as well as words that are good indicators that the example is of some class. The probability ratio only tries to select words that are indicators that the example is of some class. Also, probability ratio is more likely to select words that only appear once in the corpus (and thus appear to be good indicators, but rarely are.). For example, note that method selects the word Bob as an indicator for Greetings. As it turns out, Bob only occurred once in the 50 sessions. Mutual information does a better job of filtering out instances like this.

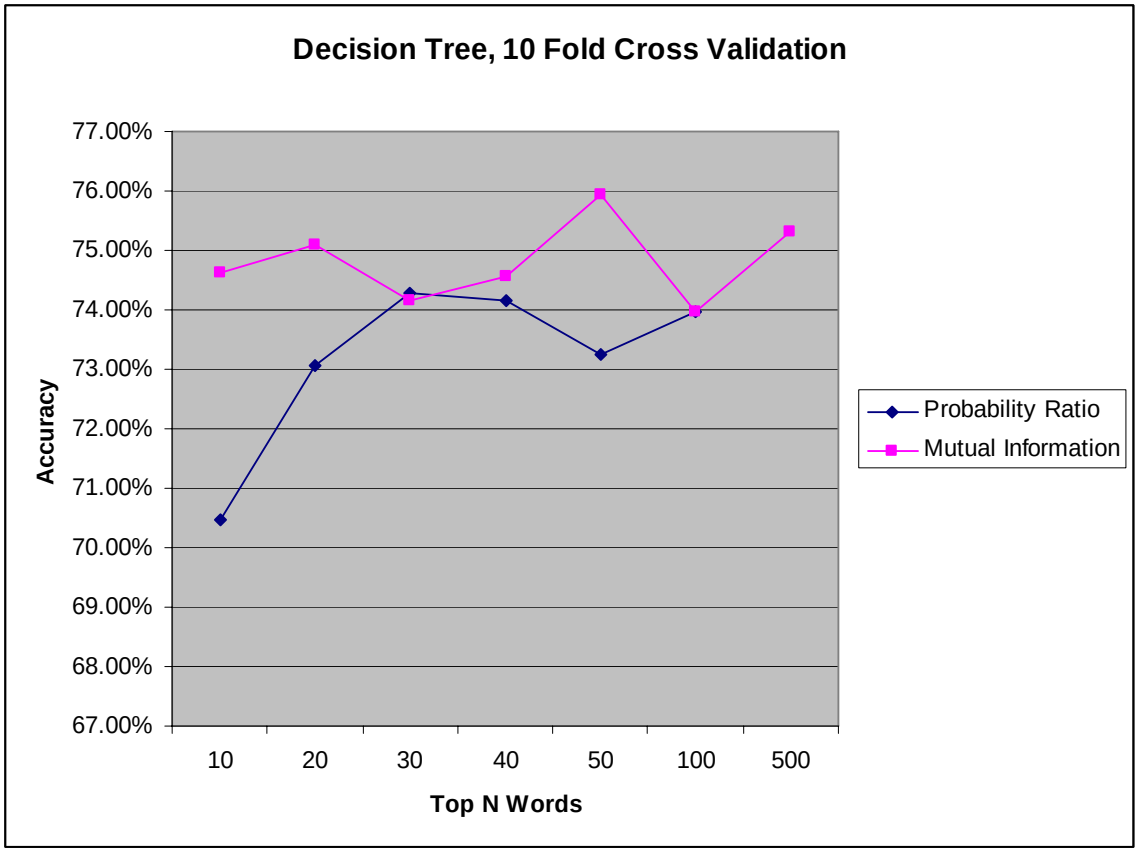
Top 10 Indicator Words per Class Selected by Mutual Information

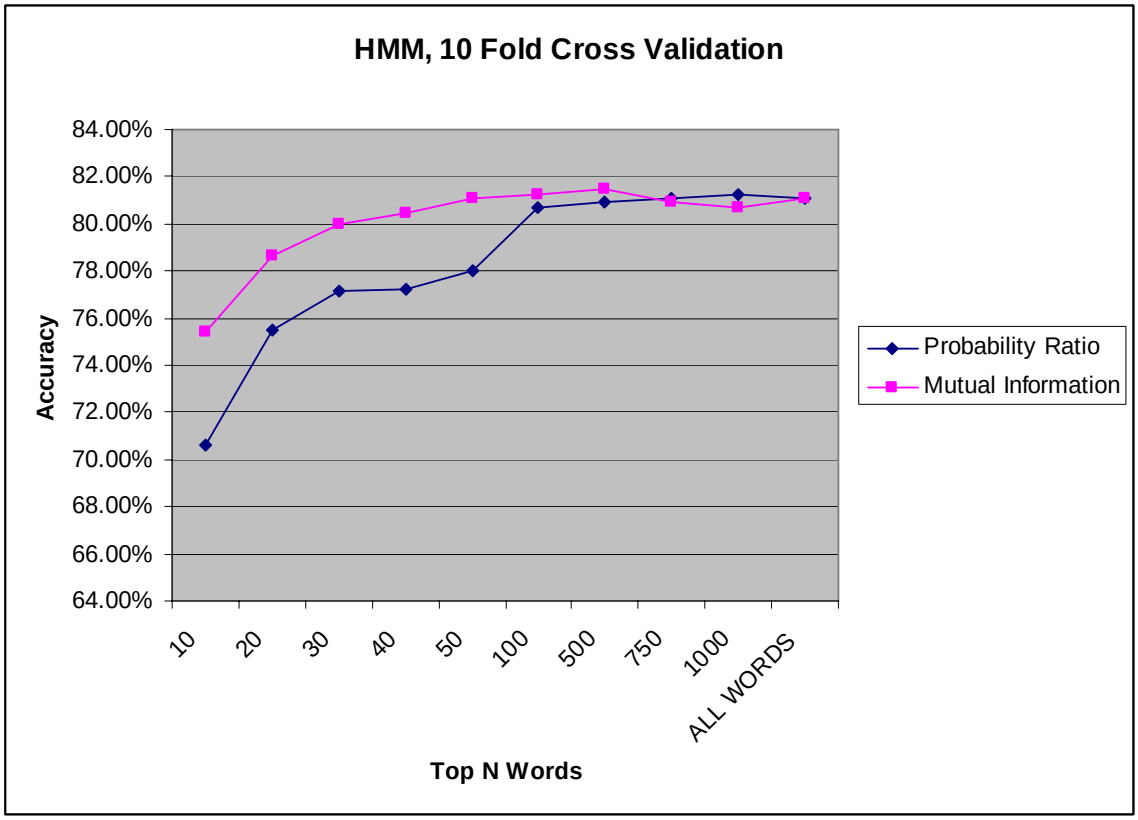
CLASS	WORDS
Assertion	here I is we heres not the your its sir
Continuation	and a the salmon nectarine tart house but grilled after
Directive	please beer have wine a can mignon red filet water
Expressive	thank thanks sorry great you im very was much excellent
Greeting	hello hi good bye come welcome again nice day afternoon
Promise	right ill be back sure moment ok will up let

Question	you would like what anything else how do are to

It is difficult to determine with the naked eye how good a set of words are for classification. Both sets of words in the tables above seem like reasonable choices. By running many experiments with each word selection method, and different values of N, we can see in the graphs below that mutual information selects words that are consistently better features for classification. In each graph we see that after some value of N, adding more features does not increase accuracy, or even decreases our accuracy. This is due to the fact that we sorted words by their relative infirmity. As we increase N, we end up using words as features that are less informative, and may result in confusing the classifiers more than they help discriminate.




Decision Tree, 10 Fold Cross Validation
77.00%
76.00%
75.00%
74.00%
73.00%
72.00%
71.00%
70.00%
69.00%
68.00%
67.00%
Accuracy
10
20
30
40
50
100
500
Top N Words
Probability Ratio
Mutual Information


HMM, 10 Fold Cross Validation
84.00%
82.00%
80.00%
78.00%
76.00%
74.00%
72.00%
70.00%
68.00%
66.00%
64.00%
Accuracy
10
20
30
40
50
100
500
750
1000
ALL WORDS
Top N Words
Probability Ratio
Mutual Information

Unigrams, Bigrams, and Trigrams, Oh My!

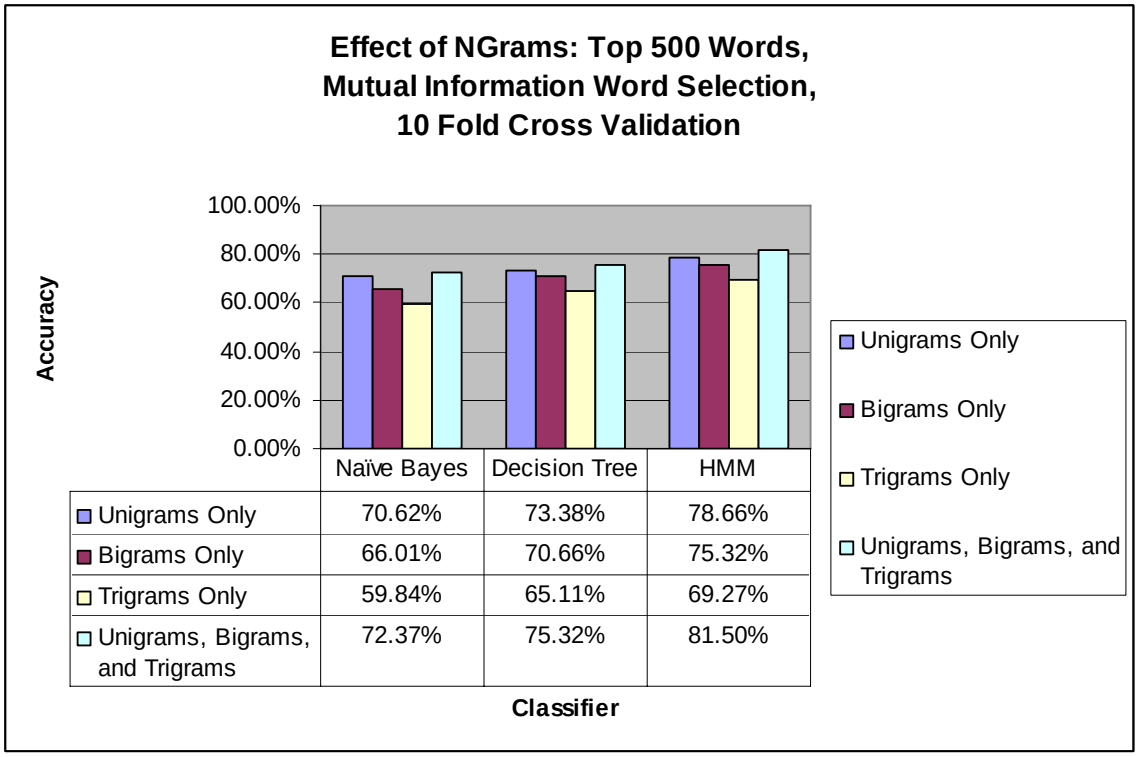
The graph below illustrates that we get the best classification when including indicator features for unigrams, bigrams, AND trigrams, rather than choosing one particular type of ngram (unigrams, bigrams, OR trigrams). Above the graph is a table of sample ngrams selected as indicators for examples of class Question.

Total Counts of Unique NGrams in Corpus:

- Unigrams: 1062
- Bigrams: 3817
- Trigrams: 4814

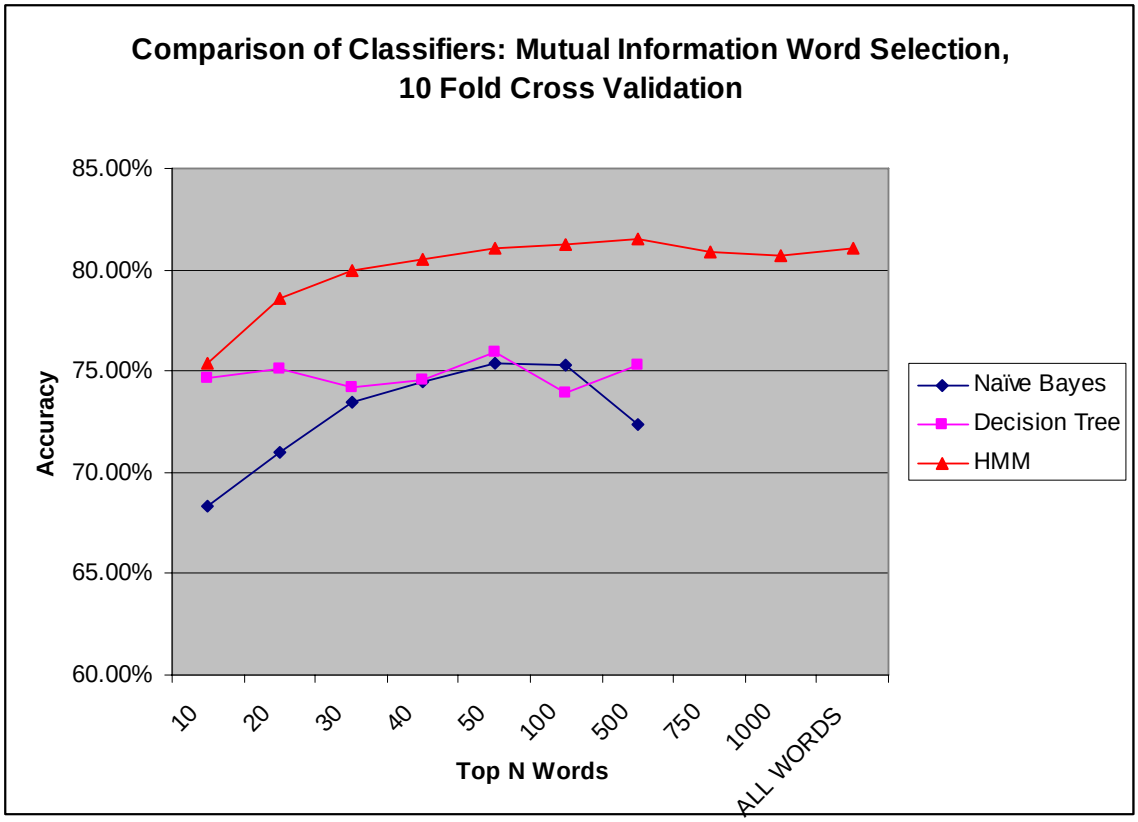
Top 10 Indicator unigrams, bigrams, and trigrams for the Question Class Selected by Mutual Information

NGRAM	WORDS
Unigram	you would like what anything else how do are to
Bigram	would_you you_like anything_else are_you do_you get_youcan_i i_get like_to to_drink
Trigram	would_you_like i_get_you can_i_get you_like_to do_you_have get_you_anything what_would_you you_anything_else you_like_a you_like_anything



Comparison of Classifiers

I compared three classification methods: naive Bayes, a decision tree, and an HMM. The graph below demonstrates that HMMs did a much better job of classifying speech acts. I used the WEKA package for naIve Bayes and J48 decision trees, and the SVMhmm package for the HMM. Naive Bayes is very fast, which is useful for quickly getting a rough sense of whether things are improving or getting worse. Below we can see that up until a point Naive Bayes' performance curve runs parallel to the HMM's curve. From this we can see that it is doing a good job of telling us how different numbers of features are affecting overall accuracy. Initially I thought decision trees might work well because most of my features are binary. I found that as I added more features, the decision tree's results improved little, or got worse. At some point, the number of features results in a tree that exceeds available memory. HMMs have the advantage of factoring in the best guess of the previous speech act, which is a very informative feature that the other classifiers lack. HMMs are able to treat the training data as a time series, and use speech act transition probabilities to predict future speech acts.



The best result in above graph was 81.5% accuracy with an HMM, using the top 500 words selected by Mutual Information, with 10 fold cross validation. With the same features and hold-one-out cross validation I achieved my overall best accuracy of 81.76% (holding out one script out of 50, so really 50 fold cross validation).

Tagging with HMMs

Classifying a sequence of speech acts based on observations is very similar problem to tagging a sequence of words with part of speech tags. HMMs have been applied successfully to the part of speech tagging problem, so I looked for an HMM that had been used for that purpose to apply to the speech act classification problem. Below is an example of part of speech tagging.

Part of Speech Tagging

INPUT:

Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT:

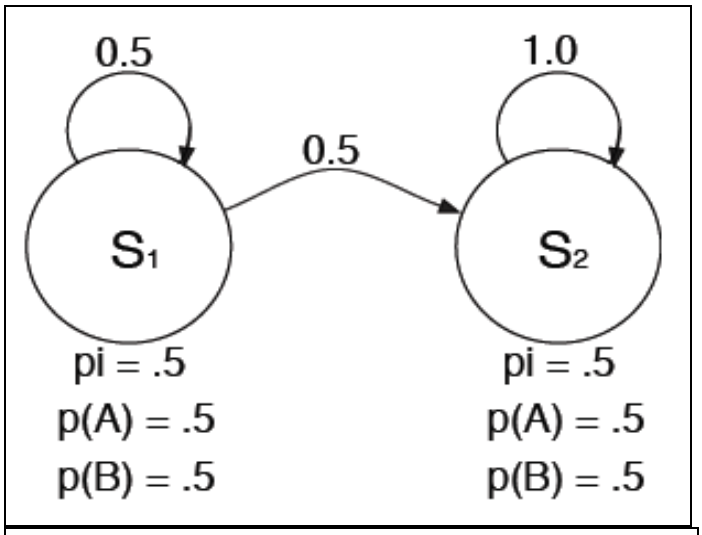
Profits/N soared/V at/P Boeing/N Co./N ,/, easily/ADV topping/V

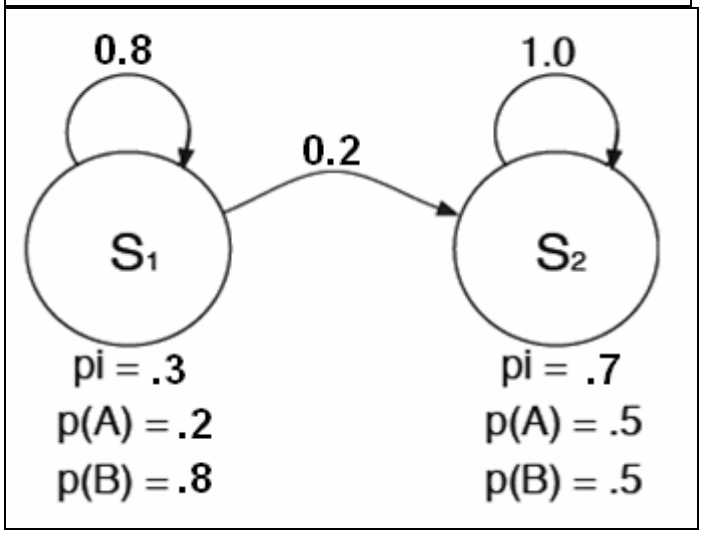
forecasts/N on/P Wall/N Street/N ,/, as/P their/POSS CEO/N

Alan/N Mulally/N announced/V first/ADJ quarter/N results/N ./.

N = Noun
V = Verb
P = Preposition
Adv = Adverb
Adj = Adjective

For part of speech tagging, we use an HMM in a different way than the way we used HMMs for classification on our problem set. For our problem set, we trained two different HMMs with sequences generated from two different models, and then used these HMMs to determine the probability that a test sequence came from one model or the other. In these HMMs the states are abstract, unlabeled entities.

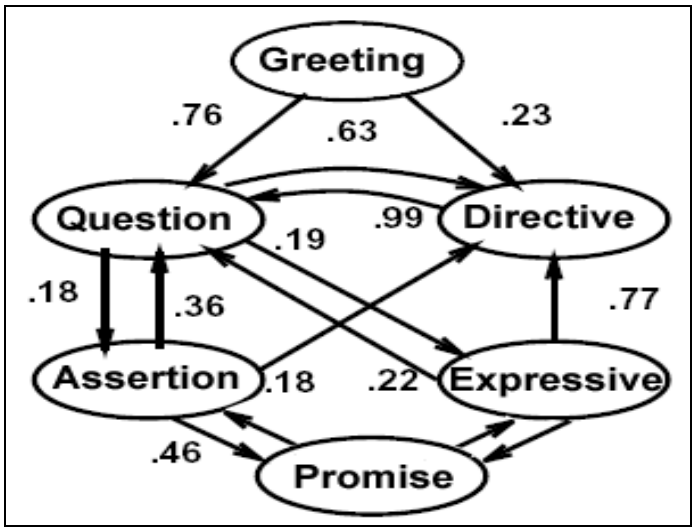




Class A **Class B**

Train two HMMs with observation sequences generated from either class A or B.
Compute the probability that a test sequence came from model A or B.

For part of speech tagging (or speech act labeling), we have only one HMM and the states actually represent the tags that we are trying to find. We can train this HMM with labeled data to learn the transition probabilities and the probability of different observations in each state. The entire feature vector for one example is a single observation. We can then use the Viterbi algorithm to find the most likely sequence of tags to produce the known sequence of observations. (Ignore the probabilities in the figure below. They are for illustration only, and are missing from some transitions).



Train a single HMM with labeled data (one tag per observation in the sequence).
Use Viterbi to find the most likely tag sequence to produce the sequence of observations.

Not every HMM toolkit allows training with labeled data, so I specifically looked for an HMM that had been used for part of speech tagging. I found SVMhmm, which boasts that it can handle vectors of over 400,000 features, and can outperform the three most popular part of speech taggers. It is developed by the same people who developed SVMlite, and is available from their wesbite here: http://svmlight.joachims.org/ SVMhmm is a unique HMM implementation that learns one weight vector for each tag, and one weight vector for the transition weights between adjacent tags. It would be interesting to test other HMMs to see how much my results were affected by this particular HMM implementation.

Monkey Wrench: the Continuation Class

I have a seventh speech act class that is a bit of a kludge to handle the fact that some people choose to break up one utterance among multiple lines. For example:

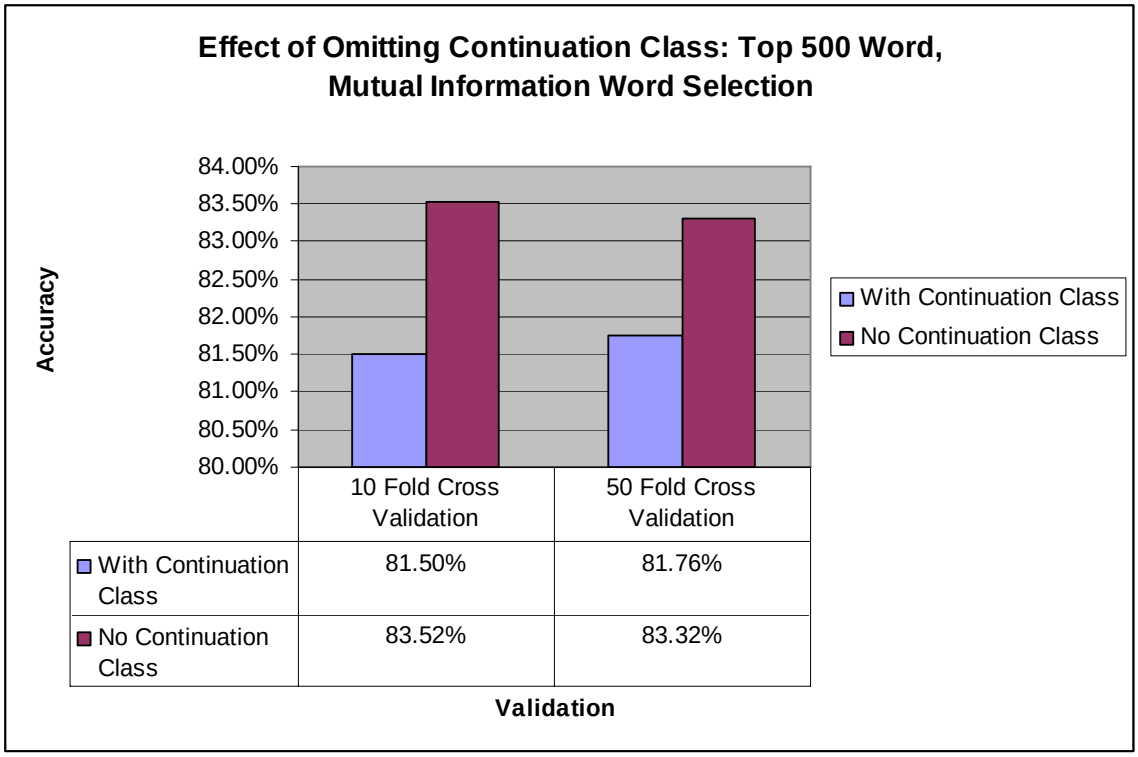
```
0332835 WAITRESS: "we have cheese cake"
0335002 WAITRESS: "a berry pie"
0337984 WAITRESS: "and a nectarine tart"
```

We can see from the confusion matrix, precision, and recall from our best classification settings that we classify Continuations poorly compared to other classes. This is due to

the fact that continuations are relatively rare, so there is less training data, and due to the fact that there's not necessarily any rhyme or reason to why people choose to break up speech acts.

```
=== Detailed Accuracy By Class ===

Precision   Recall  F-Measure   Class
   0.725     0.802      0.762   Assertion
   0.61      0.263      0.368   Continuation
   0.849     0.867      0.858   Directive
   0.794     0.814      0.804   Expressive
   0.91      0.829      0.867   Greeting
   0.855     0.859      0.857   Promise
   0.878     0.855      0.866   Question

=== Confusion Matrix ===

  a    b    c    d    e    f    g   <-- classified as
372    2   36   40    1    8    5   | a = Assertion
 19   25   23   14    1    4    9   | b = Continuation
 42    4  624   24    4    6   16   | c = Directive
 43    6   17  417    4   13   12   | d = Expressive
  5    0    7    6  121    1    6   | e = Greeting
 12    2   10    7    0  195    1   | f = Promise
 20    2   18   17    2    1  353   | g = Question
```

Out of curiosity, I tried omitting examples labeled as Continuations to see what the impact of this class was on overall accuracy. I found that an HMM could classify about 2% better without the Continuation class. This was not a valid test, because it relied on knowing a priori which examples to exclude. In retrospect, continuations might be best handled outside of the classifier, perhaps by concatenating speech acts coming from the same person within some threshold of time.

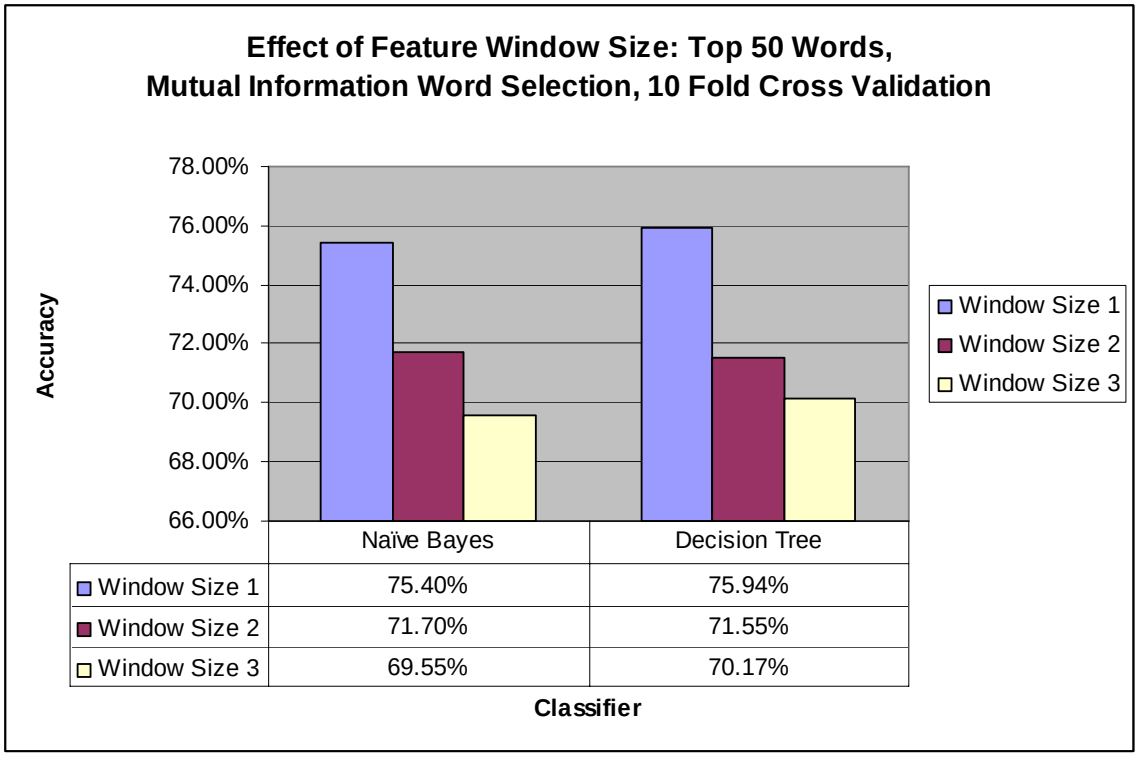


Other Things I Tried That Did Not Work Very Well

I tried a couple other things that did not work as well as I had hoped.

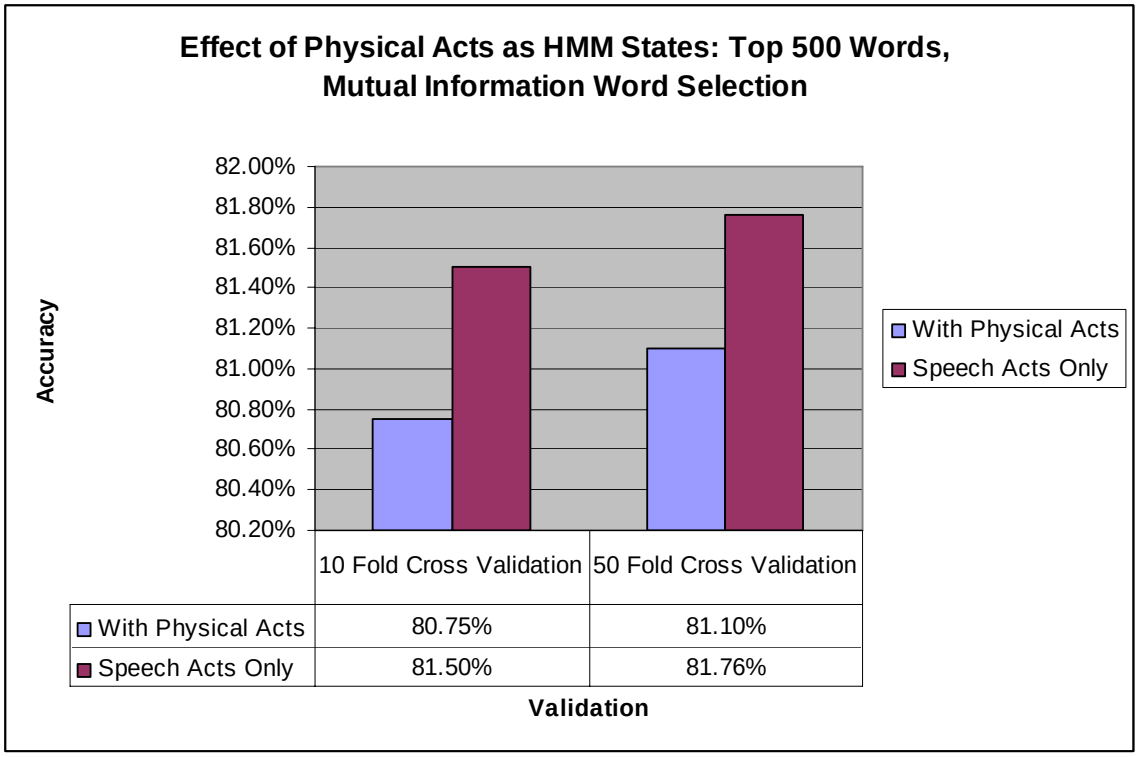
Effect of Feature Window Size

Before trying an HMM, I tried to give the other classifiers some means of taking history into account. Exporting features with a window size bigger than one would include all of the features from the previous one or two examples. I found that the accuracy decreased as the window got bigger, unfortunately. I believe this is due to the variability of the features. Using a window of features might work better if I had much more training data.



Effect of Physical Acts as HMM States

I thought I could improve classification with an HMM by including the physical acts as states, in addition to the speech acts. This would allow the HMM to learn all of the dynamics of the script. I added a set of features that acted as perfect indicators of physical acts, since they are fully observable. The result was a small decrease in classification accuracy. I believe there is a lot of noise in the ordering of physical acts, so they are not as informative as I had hoped. I think this approach is still worth pursuing, but maybe with more specifics about the physical acts, including who is taking which physical acts, and possibly what objects are involved.


Effect of Physical Acts as HMM States: Top 500 Words, Mutual Information Word Selection
82.00%
81.80%
81.60%
81.40%
81.20%
81.00%
80.80%
80.60%
80.40%
80.20%
Accuracy
With Physical Acts
Speech Acts Only
10 Fold Cross Validation
50 Fold Cross Validation
With Physical Acts 80.75% 81.10%
Speech Acts Only 81.50% 81.76%
Validation