COMP 550 A1

Yue Violet Guo
(260606976)

1 Ambiguity

- Every student took a course
  - The ambiguity is subject object agreement.
  - The different interpretations are that it could mean each student took their own chosen courses, or everyone took that exact same one course.
  - The domain of language this cause involves is semantics ambiguity.
  - The specific part in the passage that causes this ambiguity is "every" and "a." They make is unclear how "student" and "course" are quantified.
  - A human or machine needs to know whether student(s) and course(s) have a many to one or many to many relation.
- John was upset at Kevin but he didn't care.
  - The ambiguity is pronoun reference.
  - The different interpretations of "he" are two possible cases. It could refer to Kevin or John.
  - The specifics in the passage that causes the ambiguity is "he." Who is "he" referring to?
  - The domain of languages that causes involves is pragmatic. The sentence contains two clauses separated by "but," both of which can form a complete, grammatical sentence on their own. Since we have not covered parsers in class, I will present a rough idea. We can use the rules from English grammar to build two parse subtrees for the two clauses, and join them to form a tree for the whole sentence. We conclude that the most dominant ambiguity is pragmatics because (1) "he" has one dictionary literal definition when used as a pronoun (2) no evidence of different interpretations of sentence structure can be shown given what we have been taught up til now.
  - A human can infer the relationship and emotion from extralinguistic information. The two clauses both use simple past tense. Since John is upset at Kevin, it is unlikely that John is upset and not caring at the same time. A human or machine can distinguish with explicit referent to a person's name instead of "he."
- Sara owns the newspaper
  - The ambiguity is the what kind of newspaper Sara owns.
  - We can interpret Sara owns a copy of newspaper issued on a certain day, or Sara owns the company that produces a particular newspaper.
  - The specific in the passages that causes the ambiguity is "newspaper."
  - The domain of language of the ambiguity is lexical. The word for a publisher and a printed copy of an issue of a newspaper has the same form.
  - A human or machine needs to know more modifiers or nouns, whether the newspaper is the publisher that produces this particular newspaper, or an issue of a newspaper, such as "newspaper publisher" versus "this pile of newspaper." A human may also infer the social status or profession of Sara from more descriptions, and distinguish the kind of newspaper she owns.
- He is my ex-father-in-law-to-be.
  - The ambiguity is the order to parse the phrase, and which modifier in the phrase "ex-father-in-law-to-be" goes first is unclear.

  - The difference interpretations are "(ex)-(father)-(in-law)-(to-be)", a previous partner's father's future in law; or "ex-(father-in-law)-to-be", the subject's potential father-in-law from a previous relationship.
  - The specific part in the passage that causes this ambiguity is "ex-father-in-law-to-be". The specifics in the passage that causes the ambiguity are the undefined order of adjectival modifier in the phrase and the undefined granularity of each token joined by a dash.
  - The domain of language this ambiguity involves is syntactic.
  - A machine or human can disambiguate by knowing the dependency structure of the phrase "ex-father-in-law-to-be," which is learning how to properly group between lexicons the hyphens, and which one modifies which one(s). A human or machine may also be able to infer the syntactical hierarchy from more descriptions of the family tree.

- ttyl ;)
  - The ambiguity is that this spelling "ttyl" is not defined in a proper English dictionary. And this punctuation ";)" is not a defined combination in standard English.
  - The different interpretations are ttyl as a new word, or "T.T.Y.L." that represents the acronym of 4 words. The ";)" represents an emoji, or new hybrid punctuation.
  - The domain of language this ambiguity involves is orthographic.
  - The specifics in the passage is where the "ttyl" spelling is not defined in a proper English dictionary, and the combination of a semicolon and a closing bracket is also undefined in many standard of English grammar around the world.
  - A human or machine needs proper spelling to disambiguate. Use "T.T.Y.L." to clearly indicate that it is an acronym. For ";)", a human may use visual cortex, e.g. tilt his/her screen to see this hybrid punctuation as a smiley face. A human or machine can learn ttyl means a farewell if the sentences that follow indicate a separation. A machine or human can also learn emoji representations if the separation that follows is a happy one.

2 Naive Bayes and Logistic Regression

We look at a specific case of naive bayes with bernoulli distribution. As explained in lecture, a case where we have more than 2 categories can also be proven.

Let $P(y=1)=\mu$, $P(x_i|y=1)=\theta_{+i}^{x_i}(1-\theta_{+i})^{1-x_i}$, $P(x_i|y=0)=\theta_{-i}^{x_i}(1-\theta_{-i})^{1-x_i}$

$$\begin{aligned}
P(y=1|x) &= \frac{P(y=1)\prod_i P(x_i|y=1)}{P(y=1)\prod_i P(x_i|y=1)+P(y=0)\prod_{i\in x} P(x_i|y=0)} \\
&= \frac{1}{1+\frac{P(y=0)\prod_i P(x_i|y=0)}{P(y=1)\prod_i P(x_i|y=1)}} \\
&= \frac{1}{1+exp(log(\frac{P(y=0)\prod_i P(x_i|y=0)}{P(y=1)\prod_i P(x_i|y=1)}))} \\
&= \frac{1}{1+exp(log(\frac{\mu}{1-\mu})+log(\frac{\prod_i(\theta_{-i}^{x_i}(1-\theta_{-i})^{1-x_i}}{\prod_i \theta_{+i}^{x_i}(1-\theta_{+i})^{1-x_i}}))} \\
&= \frac{1}{1+exp(log\frac{\mu}{1-\mu}+\sum_i x_i(log\frac{\theta_{-i}}{\theta_{+i}}-log\frac{1-\theta_{-i}}{1-\theta_{+i}})+\sum_i log\frac{1-\theta_{-i}}{1-\theta_{+i}})}
\end{aligned} \tag{1}$$

We can rewrite the formula for Naive bayes into the same form as logistic regression, $P(y=1|x)=\frac{1}{1+exp(\sum_i w_i x_i+c)}$.

In the same feature space, if $w_i=log\frac{\theta_{-i}}{\theta_{+i}}-log\frac{1-\theta_{-i}}{1-\theta_{+i}}$ and $c=log\frac{\mu}{1-\mu}+\sum_i log\frac{1-\theta_{-i}}{1-\theta_{+i}}$, naive bayes has the same form as logistic regression.

3 Sentiment Analysis

Problem Setup

We are given two groups of movie reviews with label positive or negative, and we would like to predict the sentiment given a review. The problem is a classification problem where we predict whether the review is positive or negative. The models that we test are logistic regression, SVM, and naive bayes.

The data is relatively uniform, with roughly 50% of both classes. We decided to divide the corpus into training set (80%) and test set(20%). On the training set, we use cross validation in the sklearn's GridSearch function with default 3 fold split, which leaves one fold as the validation set. We then retrain on the original 80% train set with parameters from cross validation, and results are in Table 2.

Range of parameters

There are two kinds of parameters, one is associated with preprocssing and unigrams, the other one is regularization parameters in the model of choice to make an actual prediction. It is possible to take a combination of methods in Table 1 and find the best combination of parameters in unigram. However, due to lack of computation resource, only some combinations are experimented.

Table 1: parameters

Parameter	Range	best param
Lemmatizer	[with lemmatizer, without]	without
stemmer	[with stemmer, without]	without
unigram stopwords	[english, none]	none
unigram frequency	min_freq[0.001, 0.01, 0.1]	0
C in logistic	np.logspace(-4, 4, 20)	0.6158
C in SVM	np.logspace(-4, 4, 20)	545.5
α in Naive bayes	(0.1, 0.5, 1.0, 1.5, 2)	1.0

Experiment procedure

The experiment has mainly three steps. First, it preproceses the text, such as applying a stemmer or lemmatizer from NLTK. Second, it applies unigram model from sklearn to the text from preprocessings outlined in Table 1. Third, we run cross validation using sklearn's grid search. The experiment tries different setup outlined in the previous section.

We choose parameters according to validation accuracy. Then, we retrain on training set with the given hyperparameters, and compare the test accuracies in order to choose model that can generalize well to unseen data.

Results

Discussion:

Overall Success: We observe that under the same classification model, no preprocessing in unigram gives the best test accuracy. Under the same unigram preprocessing, Naive Bayes model has the best test accuracy of 78.50%. The overall best model is naive bayes with no preprocessing or additional parameters in unigrams. Grid search gives $\alpha = 1.0$ (Table 1) as the best parameter. All methods have test accuracies 13% to 28% above random dummy classifier.

Table 2: Results

Model	**Unigram**	**Train/Test Accuracy(%)**
Dummy	raw/lemma/stem	50.09/50.41
Logistic	Raw	97.66/77.34
SVM	raw	50.53/48.42
naive bayes	**raw**	**92.94/78.50**
Log	stop words	97.42/75.88
SVC	stop words	50.42/48.72
Naive Bayes	stopwords	93.57/76.51
log	infreq	67.28/65.04
SVC	infreq	65.72/63.01
NB	infreq	65.54/63.35
log	stem	96.16/76.89
SVC	stem	50.16/49.54
NB	stem	90.74/77.31
log	lemma	97.26/74.90
SVC	lemma	54.62/54.05
NB	lemma	92.23/77.11
NB	stop words, stem	90.68/76.33
log	stem, infreq	68.63/66.91
NB	stem, infreq	67.10/64.17

Model comparison: Logstic regression and naive Bayes are very close. It is shown in Question 2 of this assignment that they are both predict a probability. SVM's lower performance may be due to the fact that it is a non-probablistic model. In this problem, it is hard to say that a word/sequence is always positive or negative.

Failure: we can speculate that unigram regularization hurts performance. Both lemmatizer and stemmer failed to reduce overfitting, as we can observe that the difference between train and test accurarcies only drop around 2% for models using the two difference methods. The reason of lemmatizer and stemmer not working well could be that they cannot apply the same set of rules to foreign text, typos, etc in the noisy data.

Overfitting: Despite the high test accuracy, we can see many cases of overfitting in all methods except removing infrequent words and the last two rows in table 2. When we remove infrequent words, the accuracy is lower than the best ones in other unigram methods, but the train and test accuracies are both around $65 \pm 2\%$. The low performance, while still well above the dummy method, is due to the models' limited computational power. This shows that removing infrequent words is effective in preventing overfitting. We can explore this preprocessing with a more powerful model to achieve better results.

Error analysis/Confusion Matrix (eqn 2): The diagonal represents cases that are correctly classified. Precision is 80.59%, recall is 77.41%, and accuracy is 78.50%. The classifier model makes a balanced prediction on both positive and negative classes. The model does not predict biased results.

$$\begin{bmatrix} 1076 & 259 \\ 314 & 1017 \end{bmatrix} \tag{2}$$