Kate Harwood
krh2154@columbia.edu Homework 1

Experimental Design

To investigate which features are useful for predicting debates on religion vs. other topics, I looked at the split accuracy of each model I tried out.

For a given model, say **model1**, with features I was trying out, I split the test data and ran model1 on the religious debates and the non-religious debates to ascertain on which set of debates model1 did worse. I would then create a different model, **model2**, with new features, and check the split accuracy once again. If the accuracy on religious debates increased on model2 in relation to model1, but the accuracy on non-religious debates decreased, I would know that those new features were better for predicting religious debates, but not helpful for non-religious debates (or vice versa).

I could then create a model that routed religious debates to classifiers with certain features, and non-religious debates to classifiers with other features, and increase the *overall* accuracy.

For example, I created my ngrams+lex+ling model using **length** as my first linguistic feature. The overall accuracy increased from 0.60 to 0.74. Length clearly was a good feature to use. When I experimented with choosing my second linguistic feature, I observed that in a lot of cases the overall accuracy of my model (ngrams+lex+ling1+ling2) went down. But, when I conducted the experiment described above, with model1=ngrams+lex+ling1 and model2=ngrams+lex+ling1+ling2 (with ling2 as **personal pronouns)**, I saw that while my ling2 personal pronoun feature decreased the accuracy on non-religious debates, it actually increased the accuracy on religious debates to **0.63** from 0.59 (Figure 1).

Therefore I knew that if I used the personal pronouns feature for only religious debates, but not for non-religious debates, I would increase my overall accuracy. When I created a model that routed religious debates to the ngrams+lex+ling1+ling2 model and non-religious debates to the ngrams+lex+ling1 model, overall accuracy went up to **0.75** (Figure 1).

Figure 1:

Model	**Accuracy**
NGRAMS+LEX overall	0.60
NGRAMS+LEX on RELIGIOUS	0.58
NGRAMS+LEX on NON-RELIGIOUS	0.60
NGRAMS+LEX+LING1 overall	0.74
NGRAMS+LEX+LING1 on RELIGIOUS	0.59
NGRAMS+LEX+LING1 on NON-RELIGIOUS	0.78

NGRAMS+LEX+LING1+LING2 on RELIGIOUS	**0.63**
NGRAMS+LEX+LING1+LING2 on NON-RELIGIOUS	0.74
NGRAMS+LEX+LING RELIGIOUS and NON-RELIGIOUS SPLIT MODEL	**0.75**

Performance

The best model was the *Ngram+Lex+Ling+User* model. The ngrams features and using the length of the debate sides as features had the most impact on accuracy overall. **Appendix A** shows all the accuracies and scores of all models, including the models' separate accuracies on religious and non-religious debates.

Model	**Accuracy**
Simple Majority Baseline	0.53
Ngram	0.59
Ngram + Lex	0.60
Ngram + Lex + Ling	0.75
Ngram + Lex + Ling + User	0.78

Analysis

Part A: Linguistic Features

The two linguistic features I incorporated were **length** and **personal pronouns.**
When exploring the debate data, one of the aspects I looked at was average debate length. I looked at the total length of all the pro sides of all the debates and the total length of all the con sides of all the debates[1]. I also looked at the total number of pro wins and total number of con wins and saw that it was possible that length was slightly inversely correlated with winning (Figure 2), and might prove to be a useful feature for predicting the winner.

1. Length here is equal to the number of characters in the debate. This method (rather than taking the length of the tokenized debate) has the advantage of taking into account the length of words as well as the number of words.

Figure 2 (using training data):

Total length of pro sides	13353283
Total length of con sides	13127838
Total number of pro wins	676
Total number of con wins	916

The length feature addition helped the accuracy score a lot, but it helped more on non-religious debates than religious debates. So when looking for my second linguistic feature, I would prioritize a feature that would increase accuracy on religious debates. I looked at the number of religious vs. non-religious debates where the pro side and the con side won (Figure 3). This showed that the pro side was more likely to win in religious debates than it was to win in non-religious debates. So if we found a linguistic feature that made the model more likely to predict a pro win, we could increase the accuracy on religious debates.

Examining the debate data, I found that there *seemed* to be more instances of personal pronoun use in pro debates, so counting the frequencies of personal pronouns might be a good way to predict more pro debate wins, and therefore increase religious accuracy.

Figure 3 (using training data):

1: # religious debates with pro winners	146
2: # religious debates with con winners	224
3: # non religious debates with pro winners 530	676
4: # non religious debates with con winners 692	916
Ratio of 1 to 2	.65
Ratio of 3 to 4	.77

Examples of Linguistic Features Helping Identify Debate Winners

Example 1: Length

As mentioned above, it seems length is inversely correlated to winning. The average pro side of a debate is ~8390 characters long and the average con side is ~8260 characters long. The following example shows where the length feature likely helped predict the correct outcome, correcting the previous model's mistake:

Debate #9
Length: 6616
Pro length: 3488
Con length: 3128
Winner: Con
Ngrams+Lex model prediction: Pro
Ngrams+Lex+Ling model prediction: Con

The con side is shorter than the pro side in this debate, and the con side is the winner.

Example 2: PRP

Below is an example where the PRP feature may have helped predict the correct outcome, correcting the previous model's mistake:

Debate: #0
Winner: Pro
Ngrams+Lex model prediction: Con
Ngrams+Lex+Ling model prediction: Pro
Example sentences from the Pro side:

> "Thank **you** con for **your** response. **I** will do my best to go over **your** arguments and defend my case."
> "**I** am interested in debating the free will topic, So if **you** are interested, Challenge **me**."

The pro side in this debate has a large number of PRPs and is the winner (as hypothesized above).

Part B: Lexicographic Features

I chose the **NRC-VAD** lexicon because I wanted to experiment with the different types of word scores (valence, arousal, dominance) and experiment with thresholding on their scores rather than just going by overall sentiment.

To pick the features I would use from the lexicon, I experimented with different thresholds of the valence arousal and dominance scores to determine which words I would incorporate. I tried high and low cutoffs. With a high cutoff (in the .90s range) for the valence and dominance scores, I got

words like *['happily', 'happy', 'generous', 'magnificent', 'love', 'enjoyable']* and *['powerful', 'success', 'leadership']*. Counting the frequencies of these words in the debate and including that as a feature made the model's accuracy go down, while using words with a high cutoff for the arousal score (*['homicide', 'terrorism', 'exorcism', 'aggresive', 'killer']*) made the accuracy go up. This led me to believe that counting negative words had a better impact on model accuracy.

I tried out many combinations of high and low thresholds for all scores, and combinations of frequency counts of the words that this produced. Using a threshold which allowed more than just a few words made the accuracy go down. Using words low-scored on valence and dominance in addition to the words with a high arousal score also made the accuracy go down. The best performing feature was a simple count of the frequency of the words with an **arousal** score of at least **0.97**. The addition of this feature created a model that beat the ngram model's accuracy by one percentage point.

Examples of Lexicographic Features Helping Identify Debate Winners

Example 1
Debate: #19
Winner: Pro
Ngrams+Lex model prediction: Con
Ngrams+Lex+Ling model prediction: Pro
High arousal word (**homicide**) appearing in the debate, on the con side:

> *"According to the United Nations' Human Development Report (2005), countries with high levels of atheism such as Norway, Iceland, Australia, Canada, Sweden, Japan, the Netherlands, Denmark, and the United Kingdom rank as world leaders in terms of life expectancy, adult literacy, per-capita income, educational attainment, gender equality, **homicide** rate, and infant mortality."*

Example 2
Debate: #366
Winner: Pro
Ngrams+Lex model prediction: Con
Ngrams+Lex+Ling model prediction: Pro
High arousal word (**killer**) appearing in the debate, on the con side:

> *"There is teen pregnancies that could ruin her future with school and career, Maybe she just doesn't want a child at this time, Why bring a child into the world and than resent it, We have plenty of children like that, That end up turning into serial **killer**s."*
>
> *"by the way the government is there to impose and govern society based on the majority rule of the people. most people agree that life starts at birth and therefore have elected government officials to impose that as law and govern that. you don't agree with it and so you created this*

debate. unfortunately you are the few that still believe a woman has no choice. that she has to have a child because she accidentally got pregnant. and what about in case of a rape. is she supposed to have the bastard/serial ***killer*** *child? "*

These examples show where using words with high arousal scores (negative words) might hurt chances of winning.

Part C: User Information

Looking at how the voters' beliefs line up with the debaters beliefs seems like it would help a lot in predicting the voters' choice. I chose to look at the similarity between voters' views and debaters' views on the "big issues" in the user data, as well as the overlap between the voters' religion and the debaters' religions.

At first I tried to find out which big issues were the most important for predicting debate outcomes. I systematically went through each issue, and used the similarity of the voters' and pro debater's stance on that issue as a feature, as well as the similarity of the voters' and the con debater's stance as a feature. *Appendix B* shows my by-hand analysis of the new accuracies from the addition of features constructed from some of the big issues, split on religious and non-religious debates. I was looking for certain issues that would help prediction on religious debates, since the accuracy on those debates was still lagging that of non-religious debates. There were certain features, like "Socialism" and "Animal Rights" that increased some scores in the classification report when added to the current-best religious-non-religious split Ngrams+Lex+Ling model, however none were enough on their own to increase accuracy by more than a fraction of a percentage point.

When I looked at the similarities between the debaters and the voters on *all* big issues, the accuracy increased. The features I added in this iteration were the "scores" for the big issues: for each big issue, the score is 1 if the debater is for the issue, and -1 if the debater is against the issue. The voters are aggregated into one score using the same method (adding/subtracting 1 for every pro/con voter for that issue). The scores for all the issues are added together for the pro debater, the scores for all the issues are added together for the con debater, and scores for all the issues are added together for the voters, leaving us with a score for the pro debater across all the issues, a score for the con debater across all the issues and a score for the voters across all the issues as our 3 new features. The addition of these features created a model that beat my previous accuracy by 2 percentage points.

This feature tests how many big issues the debaters and voters agree on, not which ones. We could also add the users' stance on each big issue as a feature. So instead of adding 3 new features we would add 3x(# big issues) features. This would give more granularity in case some big issues are more important than others (and I know some of them are from my previous analysis when I was just looking at one big issue at a time).

Examples of User Data Features Helping Identify Debate Winners

Debate: #24
Winner: Pro
Ngrams+Lex+Ling model prediction: Con
Ngrams+Lex+Ling+User model prediction: Pro
Voter score: 35.0
Pro debater score: 6.0
Con debater score: -20.0

Debate: #335
Winner: Pro
Ngrams+Lex+Ling model prediction: Con
Ngrams+Lex+Ling+User model prediction: Pro
Voter score: 13.0
Pro debater score: 19.0
Con debater score: -20.0

We can see that in these two debates, the voters and the pro debaters have more in common on the big issues, and the model corrected the error of the previous model.

More User Features

I also added features looking at how well the voter's religions matched the pro debater and con debater's religions. I added two more features: count of how many voters in each debate matched the pro debater's religion, and count of how many voters in each debate match the con debater's religion. The addition of these features created a model that beat my previous best model by one percentage point. A way to further improve these features would be cleaning the religion tags. There are multiple types of Christianity in the religions ("Christian - <type of christianity>"), and this method only looks for exact matches. It might yield better results to strip the types and just match on "Christian".

Note: I also looked at using only certain religions, based on the most frequent religions in the user data:
Not Saying 29961
Atheist 3731
Christian 2639
Agnostic 1817
Christian - Catholic 1444
Using these frequent religions, I tried looking at the ratios of the voters of one religion to the voters of another religion. I also tried looking at raw numbers of voters of these religions (which is also related

to the number of voters in general, which could be a feature all on its own). These combinations did not improve the accuracy.

Part D

In particular, I found that the addition of the personal pronoun frequencies as a feature hurt my accuracy on non-religious debates, but improved the accuracy on religious debates. There were also certain big issues where user similarities had more impact on religious debates, and some that had more impact on non-religious debates. In particular using "Legalized Prostitution" or "Abortion" as the big issue made accuracy on religious debates go down. As shown in *Appendix A*, each feature addition had a different impact on the accuracies of each model on religious vs. non-religious debates.

Perhaps the least surprising feature that had a bigger impact on the model accuracy on religious debates vs. non-religious debates is the similarity between the debaters and voters' religion. As shown in *Appendix A*, the addition of the religion similarity feature (or "User2") increased accuracy from 0.68 to 0.72 on religious debates, while only increasing the accuracy from 0.79 to 0.80 on non-religious debates.

It makes sense that a debater's personal views will enter into their arguments in a debate, and likewise voters personal views will enter into their choice of debate winner (even if they shouldn't). When the debate topic is religion, and debaters' personal religious views leak into their debate rhetoric, it makes sense that voters would be more likely to side with the debater whose leaked views most match their own personal views. When the debate topic is not religion, it is not as likely that debaters' religious views will leak into the debate (although it may happen occasionally, accounting for the one percentage point increase in accuracy from 0.79 to 0.80).

All in all these findings show that, along with length of debate, (which was the single greatest factor in the prediction of debate winner overall), personalization in the debaters' rhetoric - whether that be use of personalizing language, like personal pronouns, or views on big issues - strongly affects the outcome of the debate, and features that capture this personalization will help to predict the winner.

A Note on Hyperparameters

In most of my models, I do not use hyperparameters for the LogisticRegression classifier. I tried out many different params, and almost none seemed to improve my accuracy (and many hurt it). For the TfidfVectorizer, I found after some experimentation that the best hyperparam settings were max_df = 0.9 and max_features = 200.

Appendix A

MAJORITY BASELINE

	precision	recall	f1-score	support
0	0.53	1.00	0.69	211
1	0.00	0.00	0.00	188
accuracy			**0.53**	399
macro avg	0.26	0.50	0.35	399
weighted avg	0.28	0.53	0.37	399

NGRAMS

	precision	recall	f1-score	support
0	0.58	0.83	0.68	211
1	0.62	0.31	0.41	188
accuracy			**0.59**	399
macro avg	0.60	0.57	0.55	399
weighted avg	0.60	0.59	0.55	399

NGRAMS PLUS LEX

	precision	recall	f1-score	support
0	0.58	0.82	0.68	211
1	0.63	0.34	0.44	188
accuracy			**0.60**	399
macro avg	0.61	0.58	0.56	399
weighted avg	0.61	0.60	0.57	399

NGRAMS PLUS LEX PLUS LING

	precision	recall	f1-score	support
0	0.71	0.85	0.77	211
1	0.79	0.61	0.68	188
accuracy			**0.74**	399
macro avg	0.75	0.73	0.73	399
weighted avg	0.75	0.74	0.73	399

NGRAMS PLUS LEX PLUS LING, LING2

	precision	recall	f1-score	support
0	0.67	0.91	0.77	211
1	0.84	0.49	0.62	188
accuracy			**0.71**	399
macro avg	0.75	0.70	0.69	399
weighted avg	0.75	0.71	0.70	399

NGRAMS PLUS LEX PLUS LING, LING2 RELIGIOUS SPLIT

	precision	recall	f1-score	support
0	0.71	0.87	0.78	211
1	0.80	0.61	0.69	188
accuracy			**0.75**	399
macro avg	0.76	0.74	0.74	399
weighted avg	0.76	0.75	0.74	399

NGRAMS PLUS LEX PLUS LING PLUS USER

	precision	recall	f1-score	support
Con	0.77	0.80	0.78	211
Pro	0.76	0.73	0.75	188
accuracy			**0.77**	399
macro avg	0.77	0.77	0.77	399
weighted avg	0.77	0.77	0.77	399

NGRAMS PLUS LEX PLUS LING PLUS LING2 PLUS USER

	precision	recall	f1-score	support
Con	0.77	0.79	0.78	211
Pro	0.76	0.73	0.75	188
accuracy			**0.76**	399
macro avg	0.76	0.76	0.76	399
weighted avg	0.76	0.76	0.76	399

NGRAMS PLUS LEX PLUS LING, LING2 RELIGIOUS SPLIT PLUS USER

	precision	recall	f1-score	support
Con	0.77	0.80	0.78	211
Pro	0.76	0.73	0.75	188
accuracy			**0.77**	399
macro avg	0.77	0.77	0.77	399
weighted avg	0.77	0.77	0.77	399

NGRAMS PLUS LEX PLUS LING PLUS LING2 PLUS USER PLUS USER2

	precision	recall	f1-score	support
Con	0.76	0.84	0.80	211
Pro	0.80	0.70	0.75	188
accuracy			**0.78**	399
macro avg	0.78	0.77	0.77	399
weighted avg	0.78	0.78	0.78	399

****BEST MODEL OVERALL****

NGRAMS PLUS LEX PLUS LING, LING2 RELIGIOUS SPLIT PLUS USER,USER2

	precision	recall	f1-score	support
Con	0.77	0.84	0.80	211
Pro	0.80	0.71	0.75	188
accuracy			**0.78**	399
macro avg	0.78	0.78	0.78	399
weighted avg	0.78	0.78	0.78	399

RELIGIOUS NON-RELIGIOUS SPLITS

RELIGIOUS NGRAMS

	precision	recall	f1-score	support
0	0.49	0.88	0.63	42
1	0.71	0.24	0.35	51
accuracy			**0.53**	93
macro avg	0.60	0.56	0.49	93
weighted avg	0.61	0.53	0.48	93

NON RELIGIOUS NGRAMS

	precision	recall	f1-score	support
0	0.60	0.82	0.70	169
1	0.61	0.34	0.43	137
accuracy			**0.60**	306
macro avg	0.60	0.58	0.56	306
weighted avg	0.60	0.60	0.58	306

RELIGIOUS NGRAMS+LEX

	precision	recall	f1-score	support
0	0.52	0.88	0.65	42
1	0.77	0.33	0.47	51
accuracy			**0.58**	93
macro avg	0.65	0.61	0.56	93
weighted avg	0.66	0.58	0.55	93

NON RELIGIOUS NGRAMS+LEX

	precision	recall	f1-score	support
0	0.60	0.81	0.69	169
1	0.59	0.34	0.44	137
accuracy			**0.60**	306
macro avg	0.60	0.58	0.56	306
weighted avg	0.60	0.60	0.58	306

RELIGIOUS NGRAMS+LEX+LING

	precision	recall	f1-score	support
0	0.53	0.79	0.63	42
1	0.71	0.43	0.54	51
accuracy			**0.59**	93
macro avg	0.62	0.61	0.59	93
weighted avg	0.63	0.59	0.58	93

NON RELIGIOUS NGRAMS+LEX+LING

	precision	recall	f1-score	support
0	0.77	0.87	0.81	169
1	0.81	0.67	0.73	137
accuracy			**0.78**	306
macro avg	0.79	0.77	0.77	306
weighted avg	0.78	0.78	0.78	306

RELIGIOUS NGRAMS+LEX+LING+LING2

	precision	recall	f1-score	support
0	0.56	0.86	0.68	42
1	0.79	0.45	0.58	51
accuracy			**0.63**	93
macro avg	0.68	0.65	0.63	93
weighted avg	0.69	0.63	0.62	93

NON RELIGIOUS NGRAMS+LEX+LING+LING2

	precision	recall	f1-score	support
0	0.70	0.93	0.80	169
1	0.85	0.51	0.64	137
accuracy			**0.74**	306
macro avg	0.78	0.72	0.72	306
weighted avg	0.77	0.74	0.73	306

RELIGIOUS NGRAMS+LEX+LING+LING2+USER

	precision	recall	f1-score	support

Con	0.62	0.76	0.68	42
Pro	0.76	0.61	0.67	51
accuracy			**0.68**	93
macro avg	0.69	0.68	0.68	93
weighted avg	0.69	0.68	0.68	93

NON RELIGIOUS NGRAMS+LEX+LING+LING2+USER

	precision	recall	f1-score	support
Con	0.80	0.82	0.81	169
Pro	0.77	0.75	0.76	137
accuracy			**0.79**	306
macro avg	0.79	0.78	0.78	306

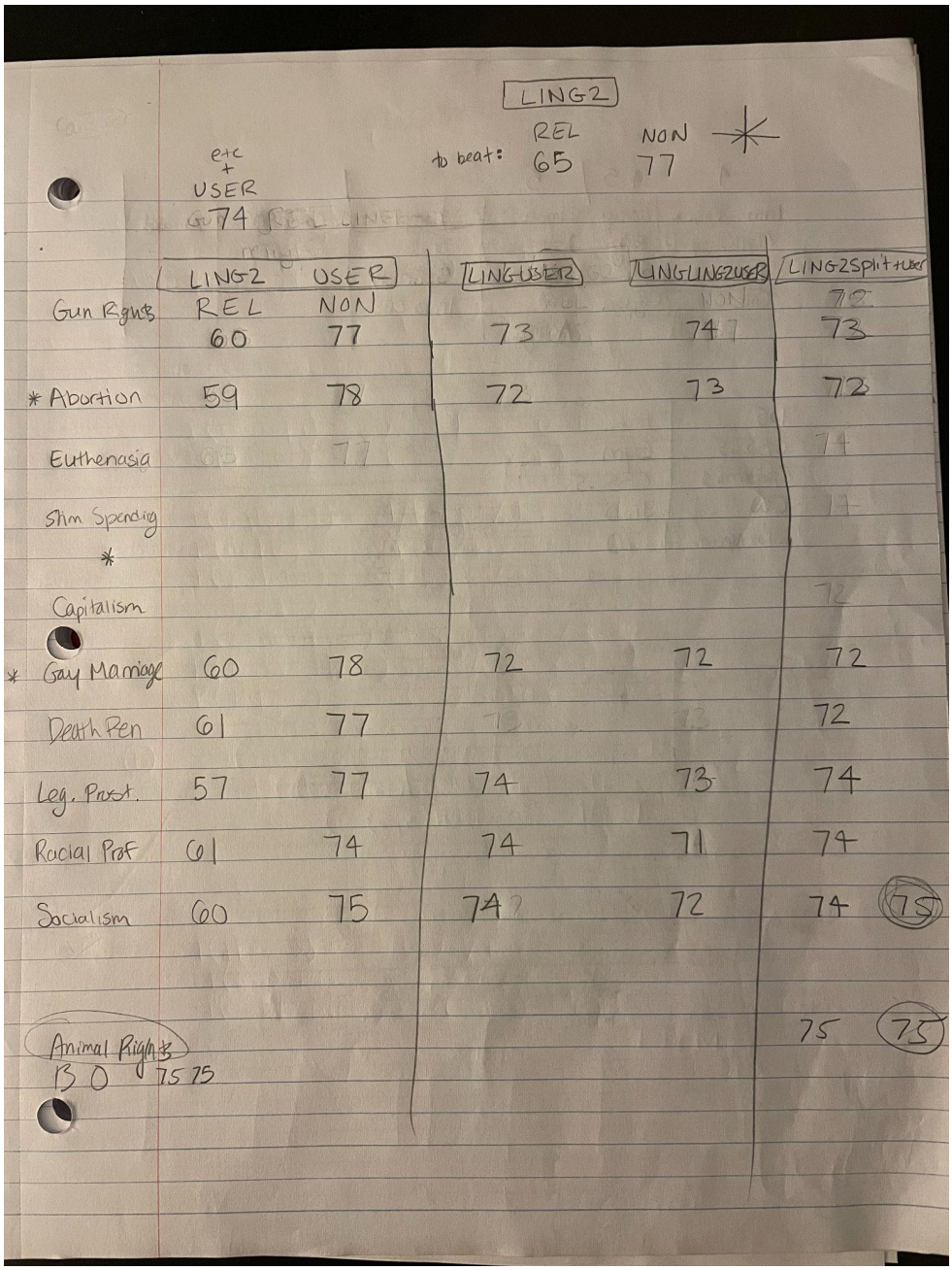
RELIGIOUS NGRAMS+LEX+LING+LING2+USER+USER2

	precision	recall	f1-score	support
Con	0.65	0.81	0.72	42
Pro	0.80	0.65	0.72	51
accuracy			**0.72**	93
macro avg	0.73	0.73	0.72	93
weighted avg	0.74	0.72	0.72	93

NON RELIGIOUS NGRAMS+LEX+LING+LING2+USER+USER2

	precision	recall	f1-score	support
Con	0.79	0.88	0.83	169
Pro	0.83	0.71	0.76	137
accuracy			**0.80**	306
macro avg	0.81	0.79	0.80	306
weighted avg	0.81	0.80	0.80	306

Appendix B



LING2

to beat: REL 65 NON 77

etc + USER 74

	LING2 REL	LING2 NON	LING-USER	LING-LING2USER	LING2Split+User
Gun Rights	60	77	73	74	73
* Abortion	59	78	72	73	72
Euthenasia					74
Stim Spending *					
Capitalism					72
* Gay Marriage	60	78	72	72	72
Death Pen	61	77			72
Leg. Prost.	57	77	74	73	74
Racial Prof	61	74	74	71	74
Socialism	60	75	74	72	74 75
Animal Rights 60 75 75					75 75

COMS 4705 NLP HW1

krh2154@columbia.edu

October 6th 2021

Problem 4: Perplexity

The first step is to estimate the probabilities of the bigrams using the training data.

Using add one smoothing, we know that:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n) + 1}{C(w_{n-1}) + |V|}$$

We compute the probabilities:

$$P(B|B) = \frac{9x9 + 1}{91 + 10} = 82/101$$

$$P(B|S) = \frac{1 + 1}{1 + 10} = 2/11$$

$$P(S|B) = \frac{1 + 1}{91 + 10} = 2/101$$

$$P(S|S) = \frac{0 + 1}{1 + 10} = 1/11$$

Then we use these probabilities to find the probabilities of the test sequences, using the equation:

$$P(w_1^n) = \Pi_{k=1}^n P(w_k|w_{k-1})$$

Then perplexity is given by:

$$\sqrt[10]{1/P(w_1^n)}$$

So we can compute:

$$Perplexity(sequence1) = \sqrt[10]{1/P(B|B)^4 \cdot P(S|B) \cdot P(B|S) \cdot P(B|B)^3}$$

$$\boxed{Perplexity(sequence1) = 2.03}$$

$$Perp(sequence2) = (1/P(B|B)\cdot P(S|B)\cdot P(S|S)\cdot P(B|S)\cdot P(B|B)\cdot P(S|B)\cdot P(S|S)\cdot P(B|S)\cdot P(B|B))^{1/10}$$

$$Perplexity(sequence2) = \sqrt[10]{1/P(B|B)^3 \cdot P(S|B)^2 \cdot P(S|S)^2 \cdot P(B|S)^2}$$

$$\boxed{Perplexity(sequence2) = 5.298}$$

$$Perplexity(sequence3) = \sqrt[13]{1/P(B|B)^{11} \cdot P(S|B)}$$

$$\boxed{Perplexity(sequence3) = 1.612}$$

Problem 5: Smoothing

Part A

If $c(i,j) > 0$ then

$$P_{Katz}(j|i) = \frac{c^*(i,j)}{c(i)}$$

If $c(i,j) = 0$ then

$$P_{Katz}(j|i) = \alpha(i) \cdot \frac{P(j)}{\Sigma_{j':c(i,j')=0} P(j')}$$

$$d = 0.75$$

$$\alpha = 1 - \Sigma_{w \epsilon A(w_{i-1})} \frac{c^*(w_{i-1}, w)}{c(w_{i-1})}$$

$$P(I| < sos >) = (3 - 0.75)/4 = 0.5625$$
$$P(went|I) = (3 - 0.75)/3 = 0.75$$
$$P(to|went) = (4 - 0.75)/4 = 0.8125$$
$$P(the|to) = (3 - 0.75)/4 = 0.5625$$
$$P(gas|the) = \alpha(the) \cdot (1/(42 - 3 - 3)) = 0.75 \cdot 1/36 = 0.0163$$

$$P(station|gas) = \alpha(gas) \cdot (3/(42 - 2)) = 0.75 \cdot 3/40 = 0.05625$$

$$P(Franciso|gas) = \alpha(gas) \cdot (5/(42-2)) = 0.75 \cdot 5/40 = 0.09375$$
$$P(to|gas) = \alpha(gas) \cdot (4/(42-2))) = 0.75 \cdot 4/40 = 0.075$$

$$P(< eos > |station) = \alpha(station) \cdot (4/42-3-3) = 0.25 \cdot (4/36) = 0.0278$$
$$P(< eos > |Francisco) = (3-0.75)/5 = 0.45$$
$$P(< eos > |to) = \alpha(to) \cdot 4/(42-4-4) = 0.25 \cdot (4/34) = 0.0294$$

So we end up with:

1. *station*: (3-0.75)/4 x (3-0.75)/3 x (4-0.75)/4 x (3-0.75)/4 x 0.75(1/36) x 0.75(3/40) x 0.25(4/36) = **0.00000627636**

2. *Francisco*: (3-0.75)/4 x (3-0.75)/3 x (4-0.75)/4 x (3-0.75)/4 x 0.75(1/36) x 0.75(5/40) x (3-0.75)/5 = **0.00016946196**

3. *to*: (3-0.75)/4 x (3-0.75)/3 x (4-0.75)/4 x (3-0.75)/4 x 0.75(1/36) x 0.75(4/40) x 0.25(4/34) = **0.00000886075**

So *Francisco* would be given as the most likely to end the sentence.

Part B

This is not what we would expect. We want *station* to be the word chosen as the most likely.

One way we can achieve this is to somehow take into account the fact that *Francisco* almost always appears with "San" before it. "San Francisco" is very likely in our corpus, but *Francisco* itself is not. To account for this, we can use Kneser-Ney discounting.

Using Kneser-Ney discounting, we would look at the context the words appear in, answering the question "How likely is w to appear as a novel continuation?" We base our probability estimates of word w "on the number of different contexts word w has appeared in, that is, the number of bigram types it completes," (Jurafsky and Martin, 2020). Since *Francisco* mostly appears with "San", it only completes 2 unique bigrams in our corpus. *station* on the other hand, completes 3 unique bigrams in our corpus. *to* completes 1 unique bigram.

Using the equation:

$$P_{KN}(w_i|w_{i-1}) = \frac{max(C(w_{i-1}, w_i) - d, 0)}{C(w_{i-1})} + \lambda(w_{i-1})P_{CONTINUATION}(w_i)$$

We can see that the first part of the equation will be 0 for each word *station*, *Francisco*, *to*, because w_{i-1} is "gas" in each case, and none of the words have appear with "gas" before. So we get:

$$P_{KN}(w_i|w_{i-1}) = \lambda(w_{i-1})P_{CONTINUATION}(w_i)$$

$\lambda(\text{“}gas\text{”})$ will be the same for each word word *station*, *Francisco*, *to*, leaving us with the only difference between the words as $P_{CONTINUATION}(w_i)$ where $P_{CONTINUATION}(w_i)$ is the number of unique bigrams the w_i completes.

$$P_{CONTINUATION}(station) = 3$$
$$P_{CONTINUATION}(Francisco) = 2$$
$$P_{CONTINUATION}(to) = 1$$

Therefore, we can see we would now get *station* as the most likely completion of the sentence.