Alexander Chatron-Michaud
260611509
COMP 599 Assignment 1

Question 1

1. Every student read a book.
   a. The ambiguity in this phrase concerns the book(s). The sentence can either be interpreted as saying that each student read the same book, or that the students read one book each, where their books weren't necessarily the same. The word that causes this ambiguity is the word "*a*", and is an ambiguity at the lexical level. In order to disambiguate the passage, another phrase with reference to the book(s) would indicate if there was more than one book being referenced.
2. This cake is not bad.
   a. The ambiguity in this phrase concerns the specific sentiment toward the taste. The sentence can either be interpreted as stating that the cake is not a bad tasting cake, and the other interpretation is the usage of the phrase "not bad", which implies a more positive sentiment. The ambiguity is the phrase or set of words "not bad", which implies a syntactic ambiguity (is it an adjective all together or two words?). One thing that could disambiguate this sentence may be the phonology/tone around the phrase if it is spoken, or information about the sentiment of the actor saying the sentence.
3. She loves the topic of her class report.
   a. The ambiguity in this phrase concerns the object of the subject's love. It can be interpreted as either her loving the topic of {her class report} or loving the topic of {her class} report, where the latter implies that the report is by or for the whole class. Because the ambiguity revolves around the interpreted structure in the sentence, it is syntactic. It could be disambiguated if there was another reference to the report with respect to either her or her whole class.
4. My English teacher recently recovered from a bowel cancer operation... and he tried to show me a semi colon. (Source: The 2016 UK Pun Championship)
   a. The ambiguity in this phrase concerns what was shown to the student. It can be interpreted as half of a colon, or a punctuation symbol, semicolon (;). The ambiguity here is orthographic because it concerns the convention of connecting the word semicolon when referencing the punctuation mark. It could be disambiguated if we could be certain that proper language conventions were being followed (or the opposite).
5. Fighting bulls can be dangerous.
   a. The ambiguity in this phrase concerns what the phrase states is dangerous. One interpretation is that {fighting bulls} (n. pl) can be dangerous, and the other is that entering fights with bulls can be dangerous. The ambiguity is whether or not the two words are one noun phrase or two separate words, and is hence a syntactic ambiguity. It could be disambiguated if there was another reference to the danger (for example, a reference to the animal species would imply that it was meant to be a noun phrase.)

Question 3

We need to see that the sum of probabilities is 1,

$$\frac{1}{N}\sum_c \left(\frac{(c+1) f_{c+1}}{f_c} \right) f_c$$

(this is the probability of a word with frequency f_c, summed over each word in f_c, summed over all words)

$$= \frac{1}{N} \sum_c (c+1) f_{c+1}$$

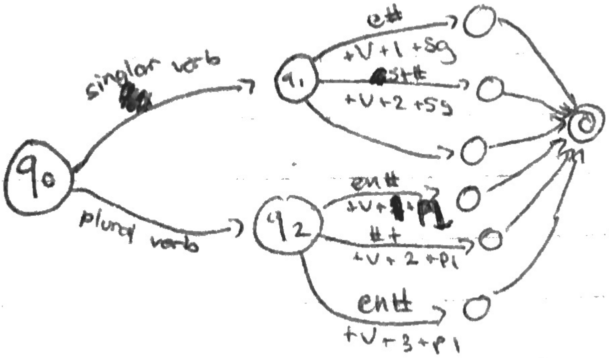
we can make this next step because f_0 isn't part of the probability distribution and neiter is f_{max+1} because there aren't any words that occur more than the max

$$= \frac{1}{N} \sum_c c f_c$$

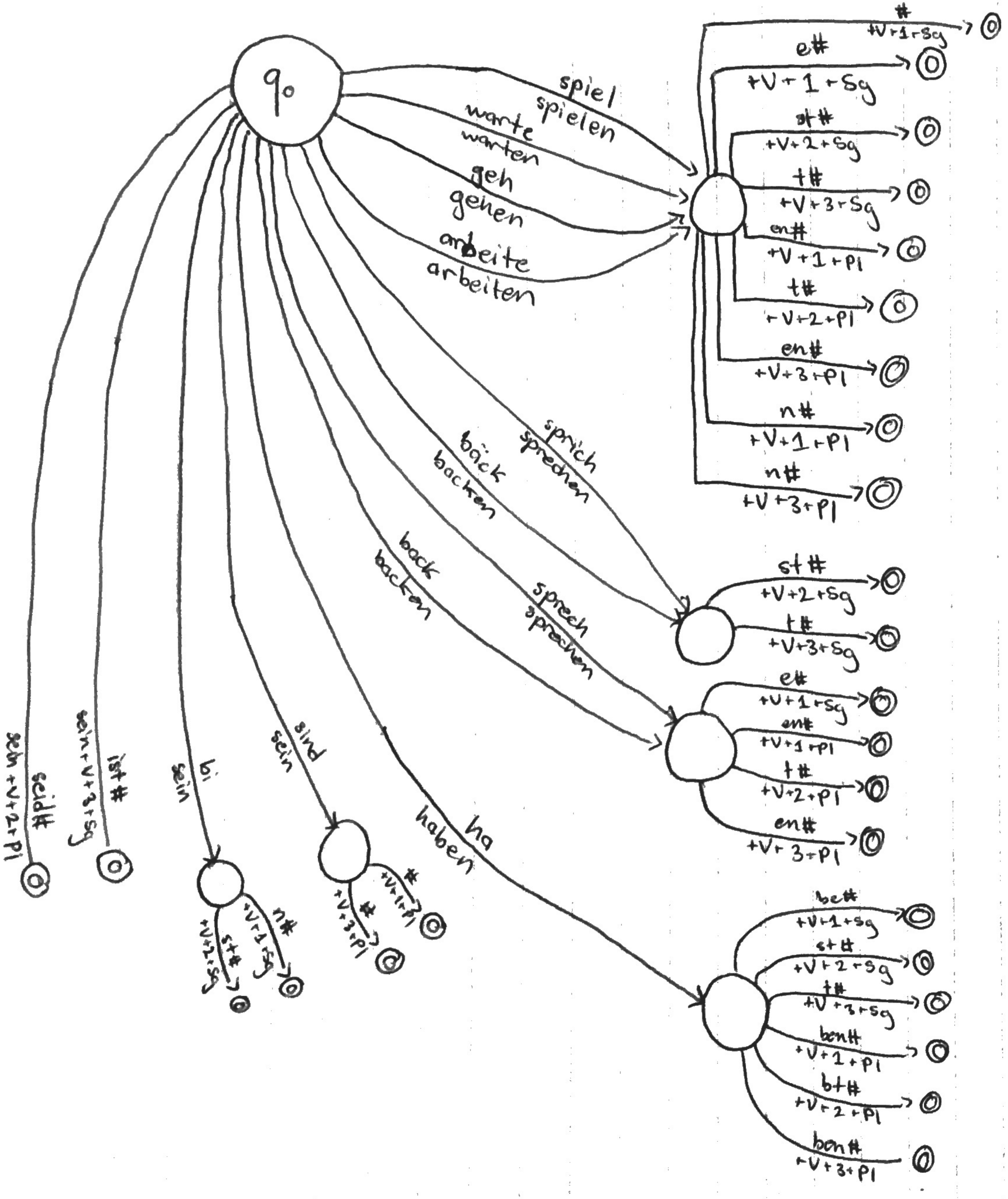
this sum is now just all the words in the corpus.

$$= \frac{1}{N} \cdot N = 1 \quad ✓$$

Question 2



Infinitive	1 Sg	2 Sg	3 Sg	1 Pl / 3 Pl	2 Pl
spielen	spiele n:e	spi el e:s n:t	spiele:t n:e	spielen	spiele:t n:e
warten	warte n:e	warte n:s e:t	warte n:t	warten	warten:t
gehen	gehe n:e	gehe:s n:t	gehe:t n:e	gehen	gehe:t n:e
arbeiten	arbeite n:e	arbeite n:s e:t	arbeiten:t	arbeiten	arbeiten:t
sprechen	sprenche n:e	spre:i che:s n:t	spre:iche:t n:e	sprechen	spre ch e:t n:e
backen	backe n:e	ba:äck e:s n:t	ba:äck e:t n:e	backen	back e:t n:e
sein	s:b e:i i:n n:e	s:be:i i:s n:t	s:i e:s i:t n:e	se:i i:n n:d	sei n:d
haben	habe n:e	hab:s e:t n:e	ha b:t n:e	haben	hab e:t n:e



* In this FST, the letters above are inputted and the letters below arrows are outputted. States for multiple letters were combined into single arrows to be concise.

(Prof. told me to clarify this.)

Question 4

Write a short report on your method and results, carefully document the range of parameter settings that you tried and your experimental procedure. It should be no more than one page long. Report on the performance in terms of accuracy, and speculate on the successes and failures of the models. Which machine learning classifer produced the best performance? For the overall best performing model, include a confusion matrix as a form of error analysis. Also, explain the role of the development set in the above experiment.

In my method for Q4, I created feature vectors by creating an ordered dictionary of all of the n-grams in the corpora in the set used for training, and then for each corpus (training, development) it received a feature vector corresponding to how many of each of the words were present in that ordered dictionary constructed from the training set. There were several hyperparameters that were then tuned using the development set. These parameters were;

- Lemmatization: when tokens were lemmatized, performance on the development set increased
- Lowercasing: when tokens were made lowercase, performance on the development set decreased
- n: unigrams performed better on the development set than bigrams and unigram+bigram concatenated as feature vectors
- Remove_top_percent: Percentage of most common words to disclude from the feature vector. This aimed to remove features like “the”, which didn’t contribute much to classifying the corpora. It was found that removing commonly occurring words continuously decreased performance on the development set.
- Count vs. Binary: It was found that using 1 or 0 to indicate the presence of a word or pair of words in the corpus performed better on the development set than using the count of the number of times it occurred

These results were also validated by performing all of the same tests on the test set after having trained the models on the 2010 articles and running them with these various hyperparameter settings to predict on the 2011 set. This showed the same results, which can be seen in the charts on the next page, giving the optimal hyperparameter settings as follows:

- (lemmatization = True, lowercase = False, n = 1, remove_top_percent = 0, count = False)

Of the three different models (Logistic regression, SVMs, Naive Bayes), Logistic Regression performed the best for all hyperparameter tests. It attained a 84.2% classification accuracy on the 2011 set after having been trained on the 2010 set with the hyperparameters listed above. It’s confusion matrix is as follows:

				Predicted Class		
		Accidents and Natural Disasters	Attacks	Health and Safety	Endangered Resources	Investigations and Trials
Actual Class	Accidents and Natural Disasters	136	2	7	33	2
	Attacks	10	145	4	12	9
	Health and Safety	0	3	193	1	3
	Endangered Resources	1	0	2	156	1
	Investigations and Trials	3	23	23	0	111

Appendix? Evidence for my analysis and implementation explain in Q4.

Model accuracy with/without lemmatize given n=1, lowercase = True, remove_top_percent = 0

	Logistic Regression	SVM	Naïve Bayes	Average
Lemmatize = False	0.8398	0.7545	0.5648	0.7197
Lemmatize = True	0.8409	0.7683	0.5625	0.7239

Model accuracy with/without lowercasing given n=1, lemmatize=True, remove_top_percent = 0

	Logistic Regression	SVM	Naïve Bayes	Average
Lowercase = True	0.8409	0.7683	0.5625	0.7239
Lowercase= False	0.8409	0.7761	0.583	0.733333333

Model accuracy removing the top x% most common words given n=1, lemmatize=True, lowercase = False

	Logistic Regression	SVM	Naïve Bayes	Average
0	0.8409	0.7761	0.583	0.733333333
5	0.7812	0.6784	0.5602	0.673266667
10	0.6466	0.5955	0.5239	0.588666667

Model accuracy as unigrams, bigrams, and both concatenated given lowercase = False, lemmatize = True, remove_top_percent = 0

	Logistic Regression	SVM	Naïve Bayes	Average
n=1	0.8409	0.7683	0.5625	0.7239
n=2	0.7159	0.6398	0.6636	0.6731
n=1,2	0.8295	0.7625	0.6432	0.745066667

Model accuracy on with binary and counted ngram features given n=1, lowercase = False, lemmatize = True, remove_top_percent = 0

	Logistic Regression	SVM	Naïve Bayes	Average
Binary features	0.842	0.8125	0.5955	0.75
Count features	0.8409	0.7683	0.5625	0.7239

	Logistic regression with binary feature unigrams, lemmatization, no lowercasing, no removal of common words	MLP with binary feature unigrams, lemmatization, no lowercasing, no removal of common words (this was just for fun)
Training Set Accuracy	1	1
Test Set Accuracy	**0.842 (final answer/best model)**	0.9091