Deliverable 8

Deliverable 8.1 (8 points):
Why is Naive Bayes a generative model? Why is Perceptron a discriminative model? Which family does logistic regression belong to? What do you see as a core difference between a generative model and a discriminative model?

Naive Bayes is a generative model as it models the joint probability distribution P(X, Y) where Y = label and X = features. Naive Bayes models joint probability distribution P(X, Y) by using Bayes rule. It computes the probability P(X|Y)[generative] and P(Y) and uses these to classify a new unseen data.
Perceptron is a discriminative model as it directly learns the boundary between the classes without modeling probability distributions. The perceptron doesn't estimate any probabilities. It just adjusts its decision boundary until all the data points are correctly classified.
Logistic regression is both discriminative and probabilistic as it directly computes the conditional probability of the label P(Y|X). It does not compute P(Y|X) by computing P(X|Y) and P(Y), instead it directly computes P(Y|X), therefore it is both discriminative and probabilistic.

Core Difference between Generative & discriminative models:
Generative Models - They model how data was actually generated and therefore, model joint probability distribution P(X, Y). Example - Naive Bayes which models P(X, Y) using Bayes rule.
Discriminative Models - They don't model any probabilities. They aim to directly learn the decision boundary between classes. Example - Perceptron, SVM

Deliverable 8.2 (4 points):

Briefly describe your bakeoff design.

I designed the following neural network:

```
my_new_model = torch.nn.Sequential(
   torch.nn.Linear(4882, 800, bias = True),
   torch.nn.ReLU(),
   torch.nn.Linear(800, 250, bias = True),
   torch.nn.ReLU(),
   torch.nn.Linear(250, 4),
   torch.nn.LogSoftmax(dim=1)
   #nn.Softmax()
```

)

-> **Network description** - Used a 3 layer neural network where 1st layer consisted of 4882 neurons, the second layer has 800 neurons and the third layer has 250 neurons. I used a softmax layer at the end which can be used for predicting the label. I used the nn.NLLLoss() function for training the neural network.

-> **Features used for training** - The features created in 7.2 are used as features for training the network. Features - counters converted to a numpy array appended with discretized token-type ratio

-> Initially, I used tanh and sigmoid activation after each layer. Since, ReLU is known to handle vanishing gradient problem and has better performance over Tanh and Sigmoid, so I replaced Tanh & Sigmoid with ReLU.

-> I also performed parameter tuning to achieve better results. The table below shows the accuracy achieved under different parameter settings.

Activation	**Learning Rate**	**Iterations**	**Momentum**	**Accuracy**
ReLU	0.04	1000	0.5	56.5%
ReLU	0.08	1000	0.5	56.7%
ReLU	**0.02**	**1000**	**0.9**	**57.6%**
ReLU	0.1	1000	0.1	57.55%

Deliverable 8.3 (8 points):

You will select a research paper at ACL, EMNLP or NAACL that performs *document* classification, using text. Summarize the paper, answering the following questions:

Paper selected: ‘Question Classification Transfer’ by Anne-Laure Ligozat

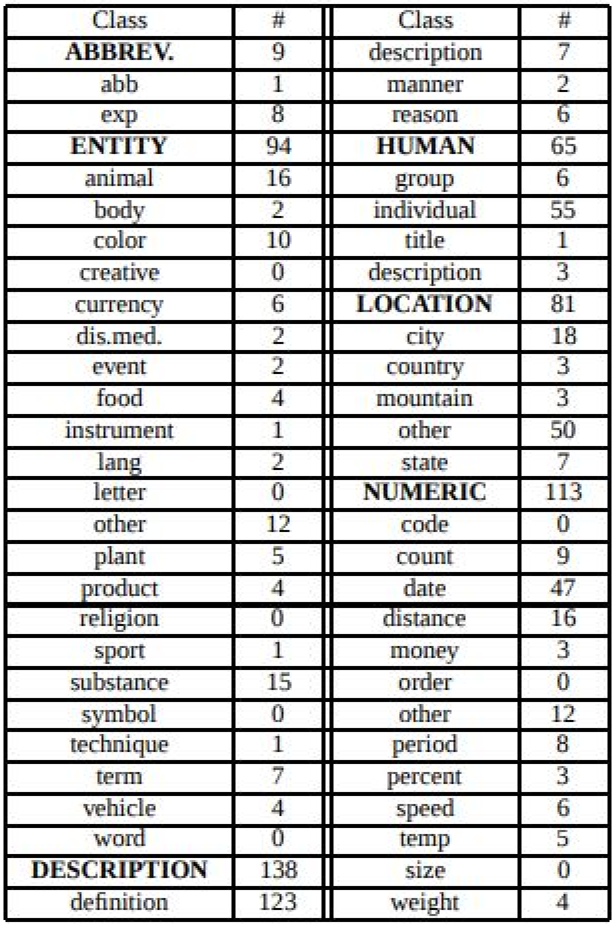
Question classification requires huge amount of labelled data (questions and corresponding question types). However, most of annotated data is available only for English language. Authors in this paper, propose 2 innovative methods of using the labelled English language data for performing question classification on French language data. The idea was to create an effective question classification system for French, with minimal annotation effort.

What are the labels, and how were they obtained?

- **English Language Data** - Authors used the data from (Li and Roth, 2002)[2], which includes data from USC, UIUC, and TREC collections. This data was manually labeled according to the taxonomy proposed by Li and Roth, 2002. Li and Roth defined a two-layered classification. The hierarchy contains 6 coarse classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE) and 50 fine classes. These were used as labels for classification. Table I shows the complete list of labels used for classification.
- The training set contains 5,500 labeled questions in English, and the testing set contains 500 questions in English language.
- **French data** - Questions were gathered from several evaluation campaigns: QA@CLEF 2005, 2006, 2007, EQueR and Quæro 2008, 2009, and 2010. A corpus of 1,421 questions was created. This was divided into a training set of 728 questions, and a test set of 693 questions. These were manually annotated based on the taxonomy mentioned in Table I.

- **Why is it interesting/useful to predict these labels?**
  - Question answering machines play a significant role in NLP. They perform the task of answering questions or justifying a snippet. These question answering systems include a question classification step which determines the type of answer to expect. For example - “What is the area of CMU campus?” -> The machine should detect that the answer expected is of ‘area’ type. Therefore, a good question classification system is crucial for any question answering machines.
  - Prediction of question labels in a multi-class classification problem and requires a large amount of annotated data. Unfortunately, most resources/data are available only for the English language this acts as a barrier in creating Question Answering machines for a different language.
  - Authors in this paper present the idea of using English language data (annotated question corpus) for performing question classification for a text in French.

- Source - Li & Roth, 2002



Class	#	Class	#
ABBREV.	9	description	7
abb	1	manner	2
exp	8	reason	6
ENTITY	94	**HUMAN**	65
animal	16	group	6
body	2	individual	55
color	10	title	1
creative	0	description	3
currency	6	**LOCATION**	81
dis.med.	2	city	18
event	2	country	3
food	4	mountain	3
instrument	1	other	50
lang	2	state	7
letter	0	**NUMERIC**	113
other	12	code	0
plant	5	count	9
product	4	date	47
religion	0	distance	16
sport	1	money	3
substance	15	order	0
symbol	0	other	12
technique	1	period	8
term	7	percent	3
vehicle	4	speed	6
word	0	temp	5
DESCRIPTION	138	size	0
definition	123	weight	4

Table 1: The distribution of 500 TREC 10 questions over the question hierarchy. Coarse classes (in bold) are followed by their fine class refinements.

- **What classifier(s) do they use, and the reasons behind their choice? Do they use linear classifiers like the ones in this problem set?**
  - Authors are solving a multiclass classification problem in which given a question, the aim is to predict the type of question. Authors use libSVM with default parameters that perform one-vs-one multiclass classification.
  - The authors mentioned that this classifier has performed well on similar classification tasks in the past, and therefore, they decided to use it for their experiments.
  - Yes, authors use LibSVM which is a linear classifier.
- **What features do they use? Explain any features outside the bag-of-words model, and why they used them.**

- The authors extracted bag-of-n-gram features from the dataset and used it for training the classifier.
- bag-of-n-gram features with n = 1 and 2.
- Bag-of-ngram - A n-gram is a sequence of n tokens or words. So, a bag of n-grams is a collection of n-grams in a given text or document. A n-gram is more informative than bag of words as it gives more context. For n=1. N-gram is same as bag-of-words.
- Example - "its water is so transparent that", we could break this up into following n-grams: 1-grams: {its, water, is, so, transparent, that}; 2-grams: {its water, water is, is so, so transparent, transparent that}
- Authors used bag of n-grams with n=1 and n=2. Authors used these features as they were used by Zhang & Lee[1], which authors use for comparison.

What is the conclusion of the paper? Do they compare between classifiers, between feature sets, or on some other dimension?

- **Conclusion** - This paper presents a comparison between two techniquesof using English question classification data for performing question classification for French language. Authors show that translating the training corpus (English language data) into French and then training a model gives better results than translating the test corpus.

  Comparison

- There is no labeled question corpus for the French language. Since creating a French language question corpus is expensive, authors propose the use of labeled English language question corpus for classification of French questions. They propose two strategies of using English question corpus:
  a) **Test on the source** - learning a classification model on English language corpus, and during test time, convert French language data into English using Google translate.
  b) **Train on target** - translating the training corpus from English to French using Google Translate. Train model on translated data and test on the original French test data.

  Comparison 1 : Authors show that 'Train on target' performs better than 'Test on source'.

  Comparison 2 : Authors compared the performance of 'Train on target' and 'Test on source' classifier with Zhang and Lee, 2003[1] that used a corpus of several thousands of questions and obtained 90% correct classification. Zhang & Lee also used the same labels, features and classifier for question classification as the authors. The authors demonstrate performance comparable with Zhang & Lee using ‘Train on Target’ technique.

Give a one-sentence summary of the message that they are trying to leave for the reader.

- Question classification can be performed effectively for French-language (language for which we do not have a large labeled corpus) by using existing and large annotated corpus of the English language.

Reference

[1] D. Zhang and W.S. Lee. 2003. Question classification using support vector machines. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 26–32. ACM

[2] X. Li and D. Roth. 2002. Learning question classifiers. In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics.

Recitation

Q1. Imagine you have two sets S_1 and S_2. S_1 contains all English words in lowercase forms. S_2 contains all English words in uppercase forms. The rank-frequency relationship for both sets follows the same Zipfian distribution.We now create a new set by sampling words from S_1 and S_2. 90% of time we sample from S_1 and 10% of time we sample from S_2 (and assuming we sample an infinite number of times so we get an idealized distribution in expectation).For the most frequent N word types in the new set, what percentage of them come from S_1 and what percentage of them come from S_2? Does it depend on N? Does it depend on the Zipfian parameter k?

S1 - contains only lower case words

S2 - contains only upper case words

Since, we are sampling from S1 90% of the time and 10% of time from S2. If we sample 1000 words, where we get 900 words from S1 and 100 words from S2. We create N most frequent word distribution, then what percent came from S1 and S2 will depend on the size of N. For example, if N=1, and let’s say all the 100 words sampled from S2 were the same word eg ‘HI’ whereas all the 900 words sampled from S1 were all distinct. So, in that case 100% of them came from S2 and 0% from S1. Therefore, it should depend on the size of N.

Q2.

Naive Bayes is a generative model where we model the joint distribution P(X, Y). If there are 4 categories, the distribution of their counts will follow multinomial distribution with success parameter as theta = (m1, m2, m3, m4) respectively for the 4 categories. We want to estimate these parameters, one approach would be to use MLE. Another approach to learn this would be to use MAP estimate. Since, dirchlet distribution is a conjugate prior for multinomial distribution, it is a common choice for prior. Dirchlet distribution has parameters alpha. So, on solving the MAP problem, we obtain the Naive Bayes with laplace smoothing.