COMP 6721 Applied Artificial Intelligence (Fall 2023)

Worksheet #9: Introduction to Natural Language Processing (NLP)

Language Model. In Natural Language Processing (NLP), a *bigram* language model is a simple yet effective way to understand the probability of a word sequence. It calculates the likelihood of a word w_n appearing after a given word w_{n-1}. We use *Maximum Likelihood Estimation* (MLE) to determine these probabilities from a given corpus: $P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$, where $C(w_{n-1}w_n)$ is the count of the occurrence of w_{n-1} followed by w_n. So, given the following corpus of three sentences:

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>
```

compute the following bigram probabilities:

$P(\texttt{I}|\texttt{<s>}) =$ .... $P(\texttt{Sam}|\texttt{<s>}) =$ .... $P(\texttt{am}|\texttt{I}) =$ ....

$P(\texttt{</s>}|\texttt{Sam}) =$ .... $P(\texttt{Sam}|\texttt{am}) =$ .... $P(\texttt{do}|\texttt{I}) =$ ....

Sentence Probability. Given an English language model with the following *bigram* probabilities, compute the probability for the sentence *"I want to eat British food"*:

P(on|eat) =	.16	P(want|I) =	.32	P(eat|to) =	.26
P(some|eat) =	.06	P(would|I) =	.29	P(have|to) =	.14
P(British|eat) =	.001	P(don't|I) =	.08	P(spend|to) =	.09
...		...		...	
P(I|<s>) =	.25	P(to|want) =	.65	P(food|British) =	.6
P(I'd|<s>) =	.06	P(a|want) =	.5	P(restaurant|British) =	.15
P(</s> |British) =	.1	P(</s> |food) =	.25	P(</s> |restaurant) =	.35

$P(I\ want\ to\ eat\ British\ food)$

$=$ ....

$=$ ....

$=$ ....

Corpus Probabilities. Given a corpus with $|V| = 1616$ different words and a total of $N = 10000$ bigrams:

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	...	***Total***
I	8	1087	0	13	0	0	0		C(I)=3437
want	3	0	786	0	6	8	6		C(want)=1215
to	3	0	10	860	3	0	12		C(to)=3256
eat	0	0	2	0	19	2	52		C(eat)=938
Chinese	2	0	0	0	0	120	1		C(Chinese)=213
food	19	0	17	0	0	0	0		C(food)=1506
lunch	4	0	0	0	0	1	0		C(lunch)=459
...									...
									...
									N=10,000

compute the probabilites for $P(II) =$ ...., $P(I|I) =$ .... and $P(lunch|I) =$ ....

Smoothing. We can avoid zero probabilities by *smoothing*, here we use add-one (or *Laplace*) smoothing:

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*	...	***Total***
I	~~8~~ 9	~~1087~~ 1088	1	14	1	1	1		
want	~~3~~ 4	1	787	1	7	9	7		C(want) + |V| = 2831
to	4	1	**11**	**861**	**4**	**1**	**13**		C(to) + |V| = 4872
eat	1	1	23	1	20	3	53		C(eat) + |V| = 2554
Chinese	3	1	1	1	1	121	2		C(Chinese) + |V| = 1829
food	20	1	18	1	1	1	1		C(food) + |V| = 3122
lunch	5	1	1	1	1	2	1		C(lunch) + |V| = 2075
...									

computing the new bigram probabilities as $P_{\text{Add1}}(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+|V|}$ and $P_{\text{Add1}}(w_{n-1}w_n) = \frac{C(w_{n-1}w_n)+1}{N+B}$, where B is the number of "bins" we added +1 to (so here, $|V|^2$). Compute the new probabilities:

$P(II) =$ ...., $P(I|I) =$ .... and $P(lunch|I) =$ ....

Part-of-Speech Tagging. Given the following lexicon, assign a *part-of-speech* (POS) tag to each word for the sentence below:

Lexicon:

```
N --> flight | trip | breeze | morning       // noun
V --> is | prefer | like                     // verb
Adj --> direct | cheapest | first            // adjective
Pro --> me | I | you | it                    // pronoun
PN --> Chicago | United | Los Angeles        // proper noun
D --> the | a | this                         // determiner
Prep --> from | to | in                      // preposition
Conj --> and | or | but                      // conjunction
```

I	prefer	a	direct	flight	to	Chigaco.

Parsing. Now, given the following context-free grammar:

Grammar:

```
S --> NP VP                          // I + prefer United
NP --> Pro | PN | D N | D Adj N      // I, Chicago, the morning
VP --> V | V NP | V NP PP            // is, prefer + United,
PP --> Prep NP                       // to Chicago, to I ??
```

create a *parse tree* for the sentence, *"I prefer a direct flight to Chicago."* using the POS tags you assigned above:

Word Sense Disambiguation. Using the following probabilities you obtained from a training corpus ($|V|$=50):

- P(*the*|BANK1) = (5+.5) / (30+.5V) P(*the*|BANK2) = (3+.5) / (12 + .5V)
- P(*world*|BANK1) = (1+.5) / 55 P(*world*|BANK2) = (0+.5) / 37
- P(*and*|BANK1) = (1+.5) / 55 P(*and*|BANK2) = (0+.5) / 37
- ...
- P(*off*|BANK1) = (0+.5) / 55 P(*off*|BANK2) = (1+.5) / 37
- P(*Potomac*|BANK1) = (0+.5) / 55 P(*Potomac*|BANK2) = (1+.5) / 37
- P(BANK1) = 5/7 P(BANK2) = 2/7

Using "add 0.5" smoothing as shown above, with a context window of ±3, find the correct sense for *bank* in the sentence, *"I like the Potomac bank"*:

1. Score(BANK1) = ...
2. Score(BANK2) = ...

Note: Words not shown in the list above have an *unsmoothed* probability of 0. Use logs.