Alexander Chatron-Michaud

260611509

Q1:

\*Note: I did this in excel. The spreadsheet where I typed out the formulae, drew the arrows, and did the Laplace calculations can be found in my mycourses submission.

Trained HMM parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Π = | | | | |
|  | that | is | it | not |
| Q1 | 0.375 | 0.250 | 0.250 | 0.125 |

|  |  |  |  |
| --- | --- | --- | --- |
| A = | | | |
|  | C | N | V |
| C | 0.200 | 0.600 | 0.200 |
| N | 0.125 | 0.250 | 0.625 |
| V | 0.143 | 0.571 | 0.286 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B = | | | | |
|  | that | is | it | not |
| C | 0.500 | 0.167 | 0.167 | 0.167 |
| N | 0.364 | 0.091 | 0.273 | 0.273 |
| V | 0.100 | 0.700 | 0.100 | 0.100 |

Run of Viterbi’s Algorithm (Up down and left arrows used to denote pointers to the previous max value used)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | not |  | is |  | not |  | that |  | is |
| C | 2.083E-02 | v | 7.102E-04 | vv | 3.551E-04 | v | 1.453E-04 | < | 4.842E-06 |
| N | **3.409E-02** | ^ | 1.136E-03 | v | **2.324E-03** | < | **2.113E-04** | ^ | 7.924E-06 |
| V | 1.250E-02 | ^ | **1.491E-02** | < | 4.261E-04 | ^ | 1.453E-04 | ^ | **9.245E-05** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Observed Emissions: | not | is | not | that | is |
| Viterbi Output: | N | V | N | N | V |

Q2:

S -> NPVP NP | DOP

NPVP -> PR1 V1 | PR2 V2 | NP3 V3 | PR4 V4 | PR5 V5 | NP6 V6

NP -> NP3 | NP6

DOP -> PR1 DT V1 | PR2 DT V2 | PR3 DT V3 | PR4 DT V4 | PR5 DT V5 | PR6 DT V6 |

NP3 -> il | elle | DT\_M\_SG A\_before\_m N\_SG\_M A\_after\_m | DT\_F\_SG A\_before\_f N\_SG\_F A\_after\_f | PR

NP6 -> ils | elles | DT\_M\_PL A\_before\_pl\_m N\_PL\_M A\_after\_pl\_m | DT\_F\_PL A\_before\_pl\_f N\_PL\_F A\_after\_pl\_f

DT -> DT\_M\_SG | DT\_F\_SG | DT\_M\_PL | DT\_F\_PL

PR1 -> je

PR2 -> tu

PR3 -> il | elle

PR4 -> nous

PR5 -> vous

PR6 -> ils | elles

DT\_M\_SG -> le

DT\_F\_SG -> la

DT\_M\_PL -> les

DT\_F\_PL -> les

A\_before\_m -> ε | beau | joli | dernier | jeune

A\_before\_f -> ε | belle | jolie | dernière | jeune

A\_before\_pl\_m -> ε | beaux | jolis | derniers | jeunes

A\_before\_pl\_f -> ε | belles | jolies | dernières | jeunes

A\_after\_m -> ε | noir | heureux | dernier | brun

A\_after\_f -> ε | noire | heureuse | dernière | brune

A\_after\_pl\_m -> ε | noirs | heureux | derniers | bruns

A\_after\_pl\_f -> ε | noires | heureuses | dernières | brunes

PR -> Jackie | Montréal | Alexander

N\_SG\_M -> poisson | chat | sac

N\_SG\_F -> télévision | plume | chaise

N\_PL\_M -> chats | poissons | sacs

N\_PL\_F -> télévisions | plumes | chaises

V1 -> regarde | mange | aime | ai

V2 -> regardes | manges | aimes | as

V3 -> regarde | mange | aime | a

V4 -> regardons | mangeons | aimons | avons

V5 -> regardez | mangez | aimez | avez

V6 -> regardent | mangent | aiment | ont

Q2 Questions:

**What are some advantages of modelling French grammar with a CFG, compared to using an FSA?**

FSA’s by definition cannot be infinite, while sentences can be correct and infinitely long.

CFG’s can be more efficient than explicit states for each case.

**What are some disadvantages of modelling French grammar with a CFG?**

There are many exceptions to rules which makes the CFG very complicated with many subgroupings.

French cannot be modeled perfectly and efficiently using a CFG, and if there was a CFG that models French perfectly it would be so explicit that it loses part of it’s purpose and advantage over FSAs.

**What are some aspects of French grammar that your CFG does not handle?**

Different tenses

Indirect object pronouns

Adverbial pronouns

Reflexive pronouns

Adverbs

Prepositions

Q3:

HMM Accuracy on cipher1:

|  |  |  |
| --- | --- | --- |
|  | With Laplace smoothing | Without Laplace smoothing |
| With language model | **0.9844** | 0.8424 |
| Without language model | 0.9766 | 0.0987 |

HMM Accuracy on cipher2:

|  |  |  |
| --- | --- | --- |
|  | With Laplace smoothing | Without Laplace smoothing |
| With language model | 0.8303 | **0.8797** |
| Without language model | 0.8312 | 0.1498 |

For the first cipher, both Laplace smoothing and the language model increased performance on the test set, more so when combined. For the second cipher, they each improved performance but not when combined. It turns out that for the second cipher, language model alone increases performance the most.

One obvious reason Laplace smoothing increases performance is that we can account for letters that are not present or whose representation in the training set is not near the true probabilistic distribution of the language. The language model also helps these issues, providing more examples for state transitions so that the representation is closer to that of real occurring frequencies in English. It is possible though that either of these can cause overfitting (not in the context of parameters, but rather to the data), which may be the reason for decreased performance when both are added in the second cipher. When observing the output, chains of ‘z’s can be seen which may indicate that the data either did not provide enough examples of the letter z or we may have provided only/too many examples of z followed by z, in either case making our trained model relying on that data too heavily.

We do not achieve perfect accuracy likely due to the fact that we don’t have enough data to probabilistically represent sequences of letters in English, and even if we did, it would probably be too biased toward the mean and it could end up doing worse on our test data overall due to its probabilistic nature (in contrast to these ciphers, which are created in a purely deterministic fashion).