ASSIGNMENT 2: [COMP550]

JOSEPH D. VIVIANO

1. POS Tagging

Behold, our corpus:

- that/C that/N is/V, is/V .
- that/C that/N is/V not/N, is/V not/N .
- is/V that/N it/N ?
- is/V it/N that/N good/J ?

The initial probabilities are calculated using only the first words from each sentence (so our corpus size for the following calculations is 4). The laplace-smoothed initial probabilities are:

$$\pi_C = \frac{\#C + 1}{\#corpus + 4} = \frac{3}{8} = 0.375 \tag{1.1}$$

$$\pi_N = \frac{\#N + 1}{\#corpus + 4} = \frac{1}{8} = 0.125 \tag{1.2}$$

$$\pi_J = \frac{\#J + 1}{\#corpus + 4} = \frac{1}{8} = 0.125 \tag{1.3}$$

$$\pi_V = \frac{\#V + 1}{\#corpus + 4} = \frac{3}{8} = 0.375 \tag{1.4}$$

The lexicon frequencies for the lexicon $l = \{that, is, not, it, good\}$ is:

- that = 6
- is = 6
- not = 2
- it = 2
- good = 1

The transition probabilities α for the tagset $\{C, N, V, J\}$ are:

Date: Sept 2018.

	C	N	V	J
C	0.000	1.000	0.000	0.000
N	0.000	0.333	0.500	0.167
V	0.000	0.800	0.200	0.000
J	0.000	0.000	0.000	0.000

The laplace-smoothed transition probabilities α for the tagset $\{C, N, V, J\}$ are:

	C	N	V	J
C	0.167	0.500	0.167	0.167
N	0.100	0.300	0.400	0.200
V	0.111	0.556	0.222	0.111
J	0.250	0.250	0.250	0.250

The emission probabilities β are:

$$C \rightsquigarrow that = \frac{2}{2} = 1 \tag{1.5}$$

$$N \rightsquigarrow that = \frac{4}{8} = 0.5 \tag{1.6}$$

$$N \rightsquigarrow not = \frac{2}{8} = 0.25 \tag{1.7}$$

$$N \rightsquigarrow it = \frac{2}{8} = 0.25 \tag{1.8}$$

$$V \rightsquigarrow is = \frac{6}{6} = 1 \tag{1.9}$$

$$J \rightsquigarrow good = \frac{1}{1} = 1 \tag{1.10}$$

Which we show in full table form here:

	that	is	not	it	good
C	1.000	0.000	0.000	0.000	0.000
N	0.500	0.000	0.250	0.250	0.000
V	0.000	1.000	0.000	0.000	0.000
J	0.000	0.000	0.000	0.000	1.000

The smoothed emission probabilities are done per state, so our new β are:

When we run the viterbi algo for POS tagging, we get the tagging "that/C is/V good/N.":

	that	is	not	it	good
C	0.429	0.143	0.143	0.143	0.143
N	0.385	0.077	0.231	0.231	0.077
V	0.091	0.636	0.091	0.091	0.091
J	0.167	0.167	0.167	0.167	0.333

	that	is	good
C	**0.38 * 0.43 = 0.16**	max(0.16*0.17*0.14; 0.05*0.10*0.14; 0.03*0.11*0.14; 0.02*0.25*0.14) = 0.00384	max(0.00*0.17*0.14; 0.01*0.10*0.14; 0.02*0.11*0.14; 0.00*0.25*0.14) = 0.00027
N	0.12 * 0.39 = 0.05	max(0.16*0.50*0.08; 0.05*0.30*0.08; 0.03*0.56*0.08; 0.02*0.25*0.08) = 0.00619	**max(0.00*0.50*0.08; 0.01*0.30*0.08; 0.02*0.56*0.08; 0.00*0.25*0.08) = 0.00073**
V	0.38 * 0.09 = 0.03	**max(0.16*0.17*0.64; 0.05*0.40*0.64; 0.03*0.22*0.64; 0.02*0.25*0.64) = 0.01709**	max(0.00*0.17*0.09; 0.01*0.40*0.09; 0.02*0.22*0.09; 0.00*0.25*0.09) = 0.00035
J	0.12 * 0.17 = 0.02	max(0.16*0.17*0.17; 0.05*0.20*0.17; 0.03*0.11*0.17; 0.02*0.25*0.17) = 0.00449	max(0.00*0.17*0.33; 0.01*0.20*0.33; 0.02*0.11*0.33; 0.00*0.25*0.33) = 0.00063

Now we consider two new sentences:

(1) "Bad is not good ."
(2) "Is it bad ?"

First, we generate a new smoothed emission probabilities matrix including the word "bad":

	that	is	not	it	good	bad
C	0.375	0.125	0.125	0.125	0.125	0.125
N	0.357	0.071	0.214	0.214	0.071	0.071
V	0.083	0.583	0.083	0.083	0.083	0.083
J	0.143	0.143	0.143	0.143	0.286	0.143

We then use the expectation maximization algorithms to update previously trained model to account for "bad" (i.e., the Baum-Welch algorithm). To do this, we first

calculate an **a** matrix using the viterbi algorithm for each sentence:

	bad	is	not	good
C	0.38 * 0.12 = 0.05	(0.05*0.17 + 0.01*0.10 + 0.03*0.11 + 0.02*0.25)0.12 = 0.00208	(0.00*0.17 + 0.00*0.10 + 0.01*0.11 + 0.00*0.25)0.12 = 0.00035	(0.00*0.17 + 0.00*0.10 + 0.00*0.11 + 0.00*0.25)0.12 = 0.00005
N	0.12 * 0.07 = 0.01	(0.05*0.50 + 0.01*0.30 + 0.03*0.56 + 0.02*0.25)0.07 = 0.00342	(0.00*0.50 + 0.00*0.30 + 0.01*0.56 + 0.00*0.25)0.21 = 0.00216	(0.00*0.50 + 0.00*0.30 + 0.00*0.56 + 0.00*0.25)0.07 = 0.00008
V	0.38 * 0.08 = 0.03	(0.05*0.17 + 0.01*0.40 + 0.03*0.22 + 0.02*0.25)0.58 = 0.01330	(0.00*0.17 + 0.00*0.40 + 0.01*0.22 + 0.00*0.25)0.08 = 0.00044	(0.00*0.17 + 0.00*0.40 + 0.00*0.22 + 0.00*0.25)0.08 = 0.00009
J	0.12 * 0.14 = 0.02	(0.05*0.17 + 0.01*0.20 + 0.03*0.11 + 0.02*0.25)0.14 = 0.00250	(0.00*0.17 + 0.00*0.20 + 0.01*0.11 + 0.00*0.25)0.14 = 0.00045	(0.00*0.17 + 0.00*0.20 + 0.00*0.11 + 0.00*0.25)0.29 = 0.00019

	is	it	bad
C	0.38 * 0.12 = 0.05	(0.05*0.17 + 0.01*0.10 + 0.22*0.11 + 0.02*0.25)0.12 = 0.00468	(0.00*0.17 + 0.03*0.10 + 0.01*0.11 + 0.01*0.25)0.12 = 0.00075
N	0.12 * 0.07 = 0.01	(0.05*0.50 + 0.01*0.30 + 0.22*0.56 + 0.02*0.25)0.21 = 0.03259	(0.00*0.50 + 0.03*0.30 + 0.01*0.56 + 0.01*0.25)0.07 = 0.00118
V	0.38 * 0.58 = 0.22	(0.05*0.17 + 0.01*0.40 + 0.22*0.22 + 0.02*0.25)0.08 = 0.00537	(0.00*0.17 + 0.03*0.40 + 0.01*0.22 + 0.01*0.25)0.08 = 0.00137
J	0.12 * 0.14 = 0.02	(0.05*0.17 + 0.01*0.20 + 0.22*0.11 + 0.02*0.25)0.14 = 0.00548	(0.00*0.17 + 0.03*0.20 + 0.01*0.11 + 0.01*0.25)0.14 = 0.00132

and our new emission probabilities β are (we are omitting all calculations beyond this point for clairty, please see attached code for more details):

	bad	is	not	good
C	0.003	0.021	0.118	1.000
N	0.006	0.017	0.124	1.000
V	0.004	0.021	0.104	1.000
J	0.005	0.018	0.141	1.000

	is	it	bad
C	0.016	0.094	1.000
N	0.013	0.096	1.000
V	0.016	0.088	1.000
J	0.014	0.106	1.000

This allows us to calculate the $p(O|\theta)$ which is 0.00041908 and 0.0046167 for each sentence respectively.

We next calculate gamma, the probability of being in state i at time t given the input sequence and the model, and xi, the probability of transitioning from state i at time t to state j at $t+1$. We present the former below for both input sequences, the latter being to large to reproduce (see the attached code for details).

	bad	is	not	good
C	0.387	0.102	0.098	0.130
N	0.126	0.139	0.641	0.201
V	0.292	0.654	0.109	0.225
J	0.195	0.105	0.151	0.444

	is	it	bad
C	0.162	0.096	0.163
N	0.026	0.677	0.255
V	0.759	0.102	0.296
J	0.053	0.125	0.287

From these values, we can then maximize the likelihood of the training data given expected counts we just calculated, and use these values to derrive new initial parameters, a, and β matrix:

$$\pi_C = 0.275 \tag{1.11}$$

$$\pi_N = 0.076 \tag{1.12}$$

$$\pi_J = 0.124 \tag{1.13}$$

$$\pi_V = 0.525 \tag{1.14}$$

	C	N	V	J
C	0.104	0.053	0.158	0.202
N	0.418	0.168	0.366	0.330
V	0.102	0.339	0.155	0.206
J	0.064	0.066	0.064	0.140

	that	is	not	it	good	bad
C	0.125	0.191	0.142	0.153	0.148	0.241
N	0.125	0.144	0.197	0.213	0.148	0.173
V	0.125	0.271	0.136	0.136	0.147	0.185
J	0.125	0.154	0.146	0.159	0.187	0.229

This solution assigns probability 0.00251934 to the new input sequence "that is bad" (found using the forward algorithm followed by summing the final column of the a matrix).

2. French Context Free Grammar

We used the following 35 nonterminal items to generate the grammar found in **grammar.txt**:

- **S** sentence
- **NP** noun phrase
- **VP** verb phrase
- **N-M-SG** noun male singular
- **N-F-SG** noun female singular
- **N-M-SG** noun male plural
- **N-F-SG** noun female plural
- **N-PL** noun plural
- **DT-M-SG** determiner male singular
- **DT-F-SG** determiner female singular
- **DT-PL** determiner plural
- **PR-1-SG** pronoun first-person singular
- **PR-2-SG** pronoun second-person singular
- **PR-3-SG** pronoun third-person singular
- **PR-1-PL** pronoun first-person plural
- **PR-2-PL** pronoun second-person plural
- **PR-3-PL** pronoun third-person plural
- **V-1-SG** verb first-person singluar
- **V-2-SG** verb second-person singluar
- **V-3-SG** verb third-person singluar
- **V-1-PL** verb first-person plural
- **V-2-PL** verb second-person plural
- **V-3-PL** verb third-person plural
- **DOPR-1-SG** direct object pronoun first-person singular
- **DOPR-2-SG** direct object pronoun second-person singular
- **DOPR-3-SG** direct object pronoun third-person singular
- **DOPR-1-PL** direct object pronoun first-person plural
- **DOPR-2-PL** direct object pronoun second-person plural
- **DOPR-3-PL** direct object pronoun third-person plural
- **PN** proper noun
- **A-M-SG-PRE** adjective male singular pre-noun
- **A-F-SG-PRE** adjective female singular pre-noun
- **A-M-PL-PRE** adjective male plural pre-noun
- **A-F-PL-PRE** adjective female plural pre-noun
- **A-M-SG-POST** adjective male singular post-noun
- **A-F-SG-POST** adjective female singular post-noun
- **A-M-PL-POST** adjective male plural post-noun
- **A-F-PL-POST** adjective female plural post-noun

2.1. **What are some advantages of modelling French grammar with a CFG, compared to using an FSA?.** CFGs are capable of modelling a much larger number of sentences with the same number of rules. With an FSA, we were forced to define a rule for each character. With CGFs, we can define rules at the word level. Even though we end up with some duplicate terminal words in the rule set, the rule set is much more compact than the equivilant FSA would have been. Furthermore, the grammar is much easier for another person to review and expand upon.

2.2. **What are some disadvantages of modelling French grammar with a CFG?.** CFGs still lead to an enormous explosion of rules very quickly when trying to deal with the edge cases of French grammar. Many rules end up pointing to only one or two specific terminal words. Therefore this approach seems unlikely to scale to the French language as practiced.

2.3. **What are some aspects of French grammar that your CFG does not handle?** This CFG does not handle conjugation properly (e.g., *"il n'y a pas"*), nor does it handle negation (*ne + [verb] + pas*). It also has no functionality to cover the different tenses (past, present, future, future perfect, etc.)

3. Book Report

3.1. **Summary.** The authors investigated whether *lexicalization* helps the performance of probabilistic context free grammars (PCFGs) or not. Lexicalization refers to using *head words* to annotate phrasal nodes. Another *unlexicalized* approach is to annotate every node by it's parent category. The paper shows that unlexical annotation can approach the performance of lexicalized PCFGs.

Models were trained on sections 2-21 of the **WSJ** section of the Penn Treebank. The first 393 sentences of section 22 were used as the development set. Section 23 was used as a test set. Unsmoothed maximum-likelihood estimates were used for rule probabilities, and grammar parsing was done using the CKY parser. The optimal tag sets were selected in this environment.

The authors present an analysis showing the effect advantage of multiple annotation approaches (i.e., how many parent levels to tag in the lexicon). They largely show an advantage of tagging words with their parent context.

The authors then show a series of annotations that split the symbol space into finer categories. They created a new tag to indicate when a production was only meaningful in locally (replacing head node annotation in lexicalized PCFGs), created two special tags distinguishing determiners from demonstratives. They found that tagging each part of speech (POS) with the the parent's POS as well was useful. Many changes were very specific: e.g., splitting the word *in* to account for the unique usage as a conjugation, complementizer, and preposition.

The authors also experimented with pushing parent-tag data down to the preterminal level of the parse tree. They marked head words that predict constituent's behaviour (e.g., possessive noun phrases). They split verb phrases using head tags, and tags denoting attachment/conjunction distance between words (e.g., noun phrases followed by multiple adjectives) using a **DOMINATES-X** tag, which encodes verbal distance between some tag tag, and tag **X**.

3.2. **Evaluation.** Advantages: They claim that these models are preferable to lexicalized PCFGs because the lower bound on the grammar's capacity can be better defined, the grammar itself is easier to understand, more compact (9255 states), easier to build (no lexicalization of content words), and more easily optimized.

Limitations: Hard to build on this work because the 1636 additions to the grammar are not detailed. Authors give little insight into the tag that were rejected during model-selection. They also did not test their model on an external corpus with a different scope (e.g., non-financial news), so some of their new tags may not have broad relevance.

Relevance to class: This work shows that encoding the relationships between words in context (parent annotation, domination, pushing functional tags down) is key for good models, and that linguistic insights can be useful for both feature engineering and error analysis.

3.3. **3 Questions:**

- Does increasing the number of closed-class word tags hurt generalization to datasets from other domains?
- What would happen if model selection preferred models with less rules?
- Is there an explanation for why marking all preterminals was the most useful annotation strategy?

McGill University
Email address: joseph@viviano.ca