**Q1.1**

Please refer to the code (line #11) for its implementation.

**Q1.2**

The probability is 0 because there is no evidence of the bigrams in the training corpus, these words never occur in the given pattern (pairs). Mathematically, to calculate the probability of and since the co-occurrence count of the word pair is 0 the overall probability of the sentence is zeroed out

**Q1.3**

The model parameters are ) and . In total there are parameters where is the total number of intermediate latent classes and is the total size of the vocabulary and

**)** indicates the probability of mapping *(the transition)* of the first word in the bigram to the hidden variable/class

**)** indicates the probability of generating the word *(the emission)* from the intermediate class

**Q1.4**

Apply the chain rule for probability

Apply the bigram assumption and cancel out common terms

,

For Saul & Pereira 1997 Model we can further simplify this as

**Q2.1**

It is necessary to model the END symbol so that we can ensure that sentence probability or likelihood corresponds to grammatical well-formedness (follows the morphology and syntax rules may not be semantically correct), meaning, a sentence which the language model assigns a higher probability is grammatically more well-formed compared to a sentence which is assigned a lower probability. For example, without an END symbol, the probability of an ungrammatical sequence *“I saw the”* would always be higher than that of the longer sentence *“I saw the red house”*. Mathematically, when we include the END symbol in our vocabulary and compute the probability of stopping (END) we are also controlling the length of the sentence generated (think about the generative story of language modeling).

**Q2.2**

Any language model should ensure that the sum of the probability of all possible sentences it can generate should be 1. So lets consider the case when the language model does not include the END symbol. We have the sum of the probability over all possible sentences (break it into the individual sums over all possible lengths 1,2 …

But in the above equation each individual sum itself would be 1, because for a given length we are doing a complete marginalization of the probability across possible word sequences (considering words in the vocabulary). This results in an infinite sum for the overall probability of the language (all possible sentences), a contradiction to the probability axiom that the sum should be 1. If instead we assume a compulsory END token (where END is not a real token of the vocabulary) appended to the end of every work sequence we would have,

And now by considering the END token in the language model we can ensure that the overall sum across all possible sentences is 1 by appropriately defining each of the individual joint probability distributions in the above equation

**Q3.1**

**Initialization**

We can initialize the parameters

**,** consider these *probabilities* to be stored in an emission matrix **EMATV\*C**

**,** consider these *probabilities* to be stored in a transition matrix **TMATC\*V**

Where classes **,** vocabulary **V**

**E-step**

Keeping the model parameters above fixed. For *each* unique bigram in the training corpus we update the distribution of the hidden latent variable class (local to this bigram instance). In pseudocode,

For

Use the model parameters from the previous step (or initialization) to calculate the posterior probability of the class . This probability can also be interpreted as finding the expected count of the bigrams that use as the intermediate state/class normalized by the expected count of the bigramssum over all possible classes. In the above notation, indicates the probability of the bigram and the intermediate state being chose as

**In pseudocode**

**Q3.2**

**M-step**

Using the posterior class probability calculated in the previous E step we now update the model parameters

This is the expected count of transitions to the intermediate state/class in all the bigram **w1w2** given that the first word in the bigram is **w1**

* To make it a probability we normalize this count
* The expectation is taken over the posterior class conditional probability distribution (calculated in the E-step)

The other update statement is,

This is the expected count of the emissions of the word **w2** (i.e. expected count of all bigrams with second word **w2**) generated by intermediate state **c** normalized by the expected number of ways to generate any bigram **w1w2** by intermediate state **c**

**Q4.1**

I defined my emission matrix as **EMATV\*C** and transition matrix as **TMATC\*V**

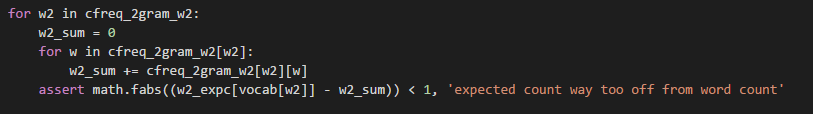
The transition matrix **TMATC\*V**, gives the distribution over all possible latent classes for a give word, so for a give word the expected count of all bigrams (starting with ) across all latent classes should be close to the actual word count

Likewise, the emission matrix **EMATV\*C** gives a distribution over all possible words that can be generated for a given latent class, so the expected count for a word being generated from all possible classes should be close to the actual word count

**Q4.2**

I verified this empirically using python condition [assert statements](https://docs.python.org/2/reference/simple_stmts.html#assert) on line 196 and 243 in the code





**Q4.3**

I verified this empirically using python condition [assert statements](https://docs.python.org/2/reference/simple_stmts.html#assert) on line 237, 201, 167







**Q4.4**

Marginal log likelihood calculation is implemented on line 135 and printed after each EM iteration



**Q4.5**

Assume a uniform unigram model for which

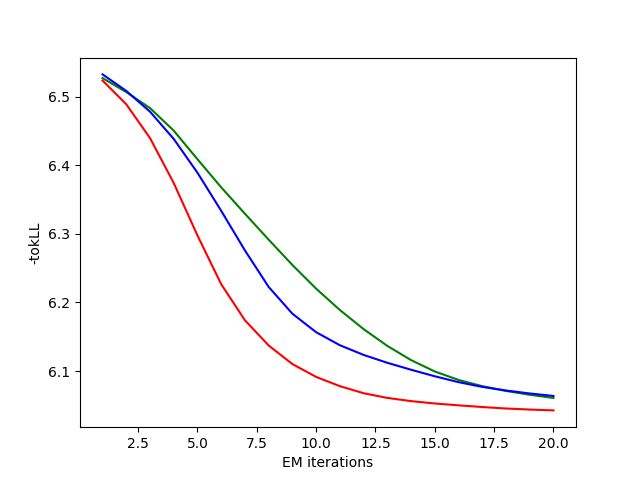
This is the “worst reasonable case “scenario since you could build a model like this without even looking at the training data.

Consider the per token average log likelihood

=

**Q4.6**

My implementation of the EM algorithm assumes 3 latent classes and runs for 20 iterations. It starts convergence at around iteration 18, 19 as the steps by which the per token average log likelihood increases is very small. If we change the number of latent classes this conclusion changes. Further, all the three random initialization converge to a fairly similar value as seen in the plot



**Q5.1**

|  |  |  |
| --- | --- | --- |
|  | **This model** | **Saul & Pereira 1997** |
| **Ratio** |  |  |
| **Latent Classes** | 3 | 16 |
| **Iterations** | 20 | 32 |
| **Training Corpus**  #of tokens, #sentences, Name | 1161192 (1.1 M), 57340 (Brown) | 78 M, 3M (ARPA NAB) |

Given the above training comparison this model reports results as expected

**Q5.2**

In a classic bigram model the log likelihood/probability of a sentence can be expressed as

So, if for even one bigram the probability is zero, the whole sentence probability is zeroed out. Hence, for the sentence *“colorless green ideas sleep furiously”*  the classic bigram reports zero probability.

While in an aggregate latent variable bigram model, each so zeros in the parameters and don’t necessarily mean a zero for . Hence, for the sentence *“colorless green ideas sleep furiously”* the aggregate latent variable bigram model gives a non-zero probability

**Q5.3**

Some other examples of sentences which are not in the training corpus which the model gets right include

|  |  |
| --- | --- |
| **Sentence (1st sentence order right, 2nd wrong)** | **Probability ratio (1st/2nd)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Q5.4**

My hypothesis—

When we make general concise statements like the ones used in news paper headlines, statements about the future, statements not directed to someone, or something or phrases which behave as conceptual units then the model gets the directionality wrong.

To resolve it we need better models, more diverse data and topical information

Example,

1. Phrase (conceptual unit)

|  |  |
| --- | --- |
| **Sentence (1st sentence order right, 2nd wrong)** | **Probability ratio (1st/2nd)** |
|  |  |

This happens because looking at the corpus we find many sentences of the form

…Maggie go from the *washing machine* to the baby to the stove and back again……the universal *grinding machine* was……he would not talk to them with a *recording machine* sitting in front of him….…A man can be an effective *fighting machine* throughout…….as to how the Howe *sewing machine* operated……seized four operative weapons, including a *Browning machine* gun…….smooth, *snug-fitting machine* when you……universal *knee-type milling machine* used throughout …

Although, there are sentences of the reverse direction too because of which the log likelihood of the sentences are not very vastly different (-25.521 and -22.691). So we need context information about the corpus to predict what is more probable, which would be different from one scenario to another

…Field does the planning for the *machine operations and fiscal processes*…….that the agency was literally dependent now on the *machine processing*…...development of nuclear technology, isotopic materials, and *machine radiation* sources in recent years…

1. Newspaper headlines

<http://nypost.com/2017/02/09/doorman-dies-after-freak-accident-while-shoveling-snow/>

|  |  |
| --- | --- |
| **Sentence (1st sentence order right, 2nd wrong)** | **Probability ratio (1st/2nd)** |
|  |  |

1. Statements about the future

<http://www.thehindu.com/news/national/The-Huddle-2017-Day-1/article17282391.ece>

|  |  |
| --- | --- |
| **Sentence (1st sentence order right, 2nd wrong)** | **Probability ratio (1st/2nd)** |
|  |  |

This hypothesis is true, as we can incorporate what the hypothesis demands, (add determiners), make the phrases complete and change future general statements to specific statements to observe that the sentence probability improves and the model gets the directionality right

|  |  |
| --- | --- |
| **Sentence (1st sentence order right, 2nd wrong)** | **Probability ratio (1st/2nd)** |
|  |  |
|  |  |
|  |  |

**Q6**

I now perform a case study of how my model [implementation of Saul & Pereira 1997] correlates sentence probability to well-formedness and during this analysis compare Chomsky’s school of thought and the information theoretic perspective

First, lets classify well-formedness of a sentence into three types:-

1. A word, phrase, clause or sentence maybe *grammatically well-formed* meaning it obeys the rules of morphology and syntax
2. A *semantically well-formed* utterance or sentence is one that is meaningful
3. A word may be *phonologically well-formed*, meaning it conforms to the sound pattern of the language

Type 1

This model gives a high likelihood to grammatically correct (syntax level) sentences as we see in **Q5.3** although there are some instances where it fails to handle finer details of syntax such as the distribution of grammatical morphemes like “*some”* or “*any”*, “*which”* or “*what”*.

Linguistic principle/rule—

Traditionally, it is viewed that *“who”* is used when speaking of humans and *“what”* is used when speaking of objects.

Consider the example,

So a purely statistical model fails in this case (gives more likelihood to **which**). We need a way to incorporate certain *“linguistic rules”* or principle fundamentals of the English language (Chomsky’s school of thought). Further, even if the sentence probability were correct and the directionality were right, it would be tough to infer the linguist rule mentioned above by looking only at probability numbers. It doesn’t give insight, it just says or provides evidence when something is right or wrong. For real world applications, that is what is needed but for scientific purposes where insight or internal workings of language needs to be uncovered this may not be favorable. Peter Norvig understands this by explaining the direction Chomsky comes from *“language generation”* which is a different topic compared to “*language interpretation”* (where bunch of user facing products and technologies lies).

But statistical models can provide high level insight. Below is printed the top words in each latent class assigned by the model (and then within each class I shuffle them to show the similar groupings highlighted). These groupings give insight that the words behave similarly and *play a similar function in sentences* or have similar usage

|  |  |
| --- | --- |
| **Class** | **Top words** |
| 0 | ['.', '!', '?', ';', "''", 'number', '1', '2', 'however', 'af', 'side', 'members', 'he', 'she', 'program', 'there', 'development', 'system', 'power', 'part'] |
| 1 | ['united', 'mrs.', 'mr.', 'has', 'had', 'have', 'looked', 'but', 'got', 'per', 'just', 'is', 'if', 'almost', ',', 'were', 'was', 'being', 'be', 'been'] |
| 2 | ['during', 'the', 'an', ‘a’, 'its', 'to', 'of', 'among', 'between', 'your', ‘our’, ‘my’, ‘their’, his’, 'into', 'against', 'under', 'toward', 'in', 'every'] |

Type 2

We now brainstorm the Chomsky’s claim that language is innate and statisticians should not waste time on language modeling

Consider the example pairs,

|  |
| --- |
|  |
|  |

For each of the above example pairs, the language model assigns almost similar sentence probability (log likelihood) although we know that the 1st sentence in each pair is more commonly used compared to the 2nd one. This is the innate aspect of the English language, all sentences in the above example are correct yet one is preferred (more well-formed) than the other, hence Chomsky’s comment. But as Peter Norvig says this is something that advanced models could fix, for instance we could train a trigram model to evaluate higher probability for

There are some examples where semantic differentiation is made indicating a scope for statistical models to achieve this and learn innate meaning of words in a language. In the below example, the language model understands that salami is a thing and cannot perform an action

|  |
| --- |
|  |

So before we use this language model inside of a dialogue system we need to improve on Type 2 well-formedness

Type 3

Consider the example pair,

Based on the above example, we can infer that if we use this language model inside of a speech recognition system it may fail to produce the right output for two similar sounding sentences. Here we come across the problem of generalizability. Another part of Chomsky's objection is "we cannot seriously propose that a child learns the values of 109 parameters in a childhood lasting only 108 seconds." Can a model trained for the purpose of language syntax checking work for speech-text systems. Although this model may not work too well, possibly we can relook this after looking at more involved and complex language models.