Universidad de Zaragoza

Entrega práctica 1

Language models (HMMs)

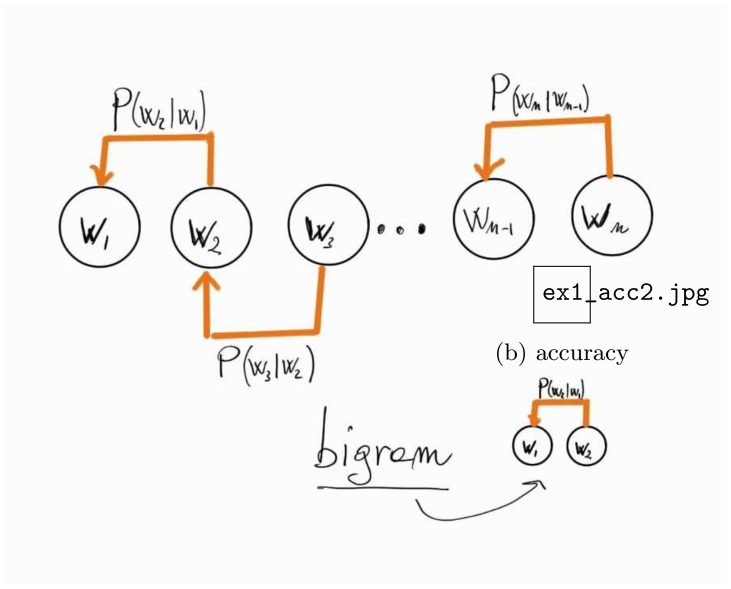
Cristiano Marinelli (853882)

October 2021

Language models (HMMs)

1 Graphical models for Bigrams and Trigrams

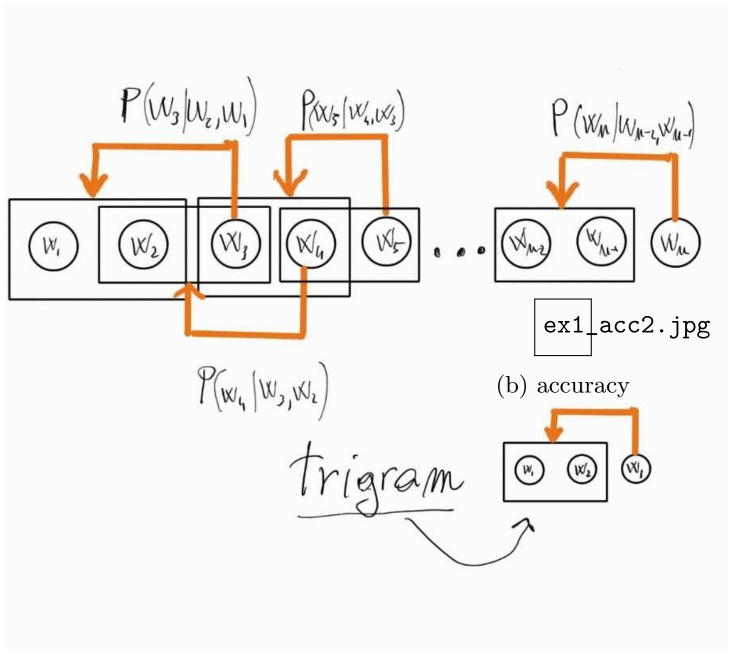
The n-grams are a linguistic computational model that analyzes the probability of encountering a word knowing the previous n-1s. The connection with the precedents makes them Markov chains. When n is two it is called bigram, while with three it is called trigram. The following images show the graphical model for both bigrams and trigrams.



ex1_acc2.jpg

(b) accuracy

(a) bigram



`ex1_acc2.jpg`

(b) accuracy

(a) trigram

2 Jupyter Notebook questions

¿Puedes explicar qué y con qué finalidad se hace el código de la construcción del modelo y en su transformación a probabilidades?

With the code

```
modelTrigrams = defaultdict(lambda: defaultdict(lambda: 0))
```

a nested dictionary is created. In particular, the first arguments are a pair of words (those that are encountered in the text) whose definitions are another dictionary with the list of the ”third” words encountered. The value associated with the end corresponds to the conditional probability of matching that third word having as precedents the two of the outermost dictionary.
After opening the text file, commands are executed to separate punctuation and to make the text a list of words. In addition to the present punctuation, other punctuation marks were added to divide which caused repetition of trigrams.
Thanks to the function `trigrams` from `nltk` package, a `for` loop, to count the number of times each sequence is encountered, is executed over the words in the trigrams generated. Then a nested `for` loop on the pairs of the ”outermost” dictionary allows you to count how many times these pairs are present in the text and to transform the counters of the previous `for` loop into conditional probabilities.
`modelTrigrams[w1_w2][w3] /= total_count`

¿Por qué aparecen `None` en los trigrams? tenen cuenta los parámetros: `trigrams(texto2, pad_right=True, pad_left=True)`

The `None`s appear due to the parameters `pad_left=True` and `pad_right=True` in the function `trigrams` and are necessary to preserve the symmetry of the model, in fact without them (two at the beginning and two at the end for the trigrams) the first word of the text (`pad_left=True`) and the last (`pad_right=True`) would generate fewer trigrams.

Realiza diferentes pruebas con secuencias de dos palabras que se encuentren seguidas, ¿tiene sentido?
`dict(modelTrigrams["Don", "Quijote"])`

This command displays the ”innermost” dictionary associated with a pair of chosen words. Trying with ”Don” and ”Quijote” it turns out that the maximum probability is the ”comma”, while with ”cuando” and ”el” it is ”sol”. Furthermore, the first couple tested has many more words than the second, and this makes sense since Don Quijote is the protagonist of the book. By entering made-up words or words from other languages the output is `{}` as expected

¿Qué significa el resultado obtenido? ¿Por qué?

This cell shows the first n pairs of the dictionary in the order in which they are encountered, and prints the sub-dictionaries with the respective probabilities for each third word on the screen.In fact the first printed pair is `None, None` (due to `pad_left=True`), then the license and the title, and so on

Explicar cómo funciona la generación de frases, desde la variable r, accumulator, puedes utilizar el log obtenido para interepretar el algoritmo

The sentence generation algorithm requires two initial words (the first pair) and other parameters. The parameters are the length of the sentence (`maxNumberofWords`) and the variable `r`. This variable is a random number less than 1 (a probability) that acts as a threshold for choice. In fact, a while loop scrolls through the words of the sub-dictionary relative to the pair and adds the respective probability value to an `accumulator` variable. When the `accumulator` exceeds the threshold `r` the word will be chosen and the new while loop will start with this new pair, a new `r`, until the predetermined number of words is reached..
Activating the `log` shows how the summation and the choice are made
This choice technique is conditioned by the order of the words in the sub-dictionaries, in fact the last ones can be chosen only with a very high `r`, and the last one only with `r` equal to 1. To solve this problem, the words of the sub-dictionaries could be randomly mixed at each loop.

3 Bigram and Comparison with the trigram

3.1 bigram from the trigram

The bigram model is a ”simplification” of the trigram. The commands to execute are almost the same. The changes implemented are mainly on the commands that referred to the number of words to be analyzed. For example, the outermost dictionary no longer contains a pair but a single word that has a sub-dictionary of ”second” words. So when pairs were called in the for loop they are now replaced with a single word.

3.2 Comparison with Gutenberg project books

The goal of this last part is to compare different books of the same language and generate sentences through the bigram and the trigram. The books are first merged into a list and then the dictionaries are generated. This way the database will be larger and there will be more variety in the sentences. The languages analyzed are Italian (with three books) and English (with 5 books). For both languages some examples will be given with different starting words and different lengths.

3.2.1 Italian

Specifically, the Italian books are ”I Promessi Sposi” by Alessandro Manzoni, a book by Mazzini and one by the Nobel Prize winner Grazia Deledda.
See the examples from the Jupyter notebook

```
examples of sentences generated by the trigram:

un uomo peggio conciato e più in Dio li riunisca altrove ;



quel giorno ; ma nell'animo di Renzo ; e si barattano saluti e buoni



da tempo , la stanchezza cresciuta , e col fuso in mano a prender la benedizione





examples of sentences generated by the bigram:
```

```
un uomo , e addobbate qua e stoppa , senza prender


quel povero curato . | Mi fa più a chi meno ne ho


da rispondere a cavallo ; e vuoi : aveva tanto uscito di nuovo la vera
```

3.2.2 English

The English books are Dubliners, Heart of Darkness, Romeo and Juliet, Moby-Dick and Frankstein.
As before the examples from the Jupyter Notebook

```
examples of sentences generated by the english trigram:


a man could not have come from the shore ; and when


that day , at that castle which overhangs yon precipice ; and so highly


since yesterday , and the scene was enveloped in an infernal mess of rust , filings




examples of sentences generated by the english bigram:


a minute circumstances , all ? It was by the room


that she sometimes I discovered to sit upon the wars going on the
```

```
since Christian souls to follow ; he had the room light , where some sort
```

3.2.3 conclusions and comments

Analyzing the meaning of the sentences, as expected, those generated by the trigrams seem to make more logical sense. By increasing the database and the n of the ngrams, we expect the sentences to make even more sense. In fact, ngrams projects like that of Google generate much more complex and meaningful sentences. The books taken from the Gutenberg project, moreover, are not very recent and use a language that is no longer widely used.
However, the ngram model still seems far from the possibility of writing a book, what it would generate would be more similar to a Joyce "stream consciousness" than to a real novel or essay.