Norges teknisk–naturvitenskapelige universitet
Institutt for datateknologi og informatikk

TDT 4310
Spring 2023

Lab 1

1 Tokenization

Consider the following sentence:
My first car (1974 Ford Pinto) was a trash-can on wheels...

1) *How would* you *tokenize this sentence into words? No coding required (nor is there a "correct"answer)*
- I would tokenize it like this: ['My', 'first', 'car', '1974', 'Ford', 'Pinto', 'was', 'a', 'trash-can', 'on' 'wheels'].

2) *If you were to type the sentence on your phone, what would you expect the next prediction to be after typing "My first car"?*
- I would expect it to be some verb, likely 'is', 'has', another tense of those, or a modal verb.

3) *What are some of your initial thoughts on the difficulty of next-word prediction in general?*
- I think finding the best data to work with can be challenging. With different habits between all the users of the predicting software, to start from one data set and end up with something that fits as many as possible is difficult.

2 Introduction to language modeling

Language modeling is, in short, the task of predicting the next word in a sentence. You may have heard of BERTM, GPT, and other language models. A simpler language model can be implemented using **n-grams**. You will be implementing a bigram and trigram model in this lab. Before you tackle this task, some basic knowledge might come in handy.
With the sentence above, use NLTK to get:

1) *Bigrams*

```
import nltk
sentence = 'My first car (1974 Ford Pinto) was a trash-can on w
tokens = nltk.tokenize.word_tokenize(sentence)
bigrams = nltk.bigrams(tokens)
print(list(bigrams))
```

```
» [('My', 'first'), ('first', 'car'), ('car', '('), ('(', '1974'), ('1974', 'Ford'),
('Ford', 'Pinto'), ('Pinto', ')'), (')', 'was'), ('was', 'a'), ('a', 'trash-can'),
('trash-can', 'on'), ('on', 'wheels'), ('wheels', '...')]
```

2) *Trigrams*

```
trigrams = nltk.trigrams(tokens)
print(list(trigrams))
```

```
» [('My', 'first', 'car'), ('first', 'car', '('), ('car', '(', '1974'), ('(', '1974',
'Ford'), ('1974', 'Ford', 'Pinto'), ('Ford', 'Pinto', ')'), ('Pinto', ')', 'was'),
(')', 'was', 'a'), ('was', 'a', 'trash-can'), ('a', 'trash-can', 'on'), ('trash-can',
'on', 'wheels'), ('on', 'wheels', '...')]
```

We can continue the n-gram model infinitely (4-grams, 5-grams, ...)

3) *What issues may occur if we select a large* n value for a small corpus? What would you guess to be ideal for a smart keyboard?
- The likeliness that you hit a corresponding stream of words in the corpus quickly gets pretty small. It would depend on the corpus data the smart keyboard was built on, but I would guess maybe 4-5 words. It would ideally degradethe ngrams used, so that it uses 4-grams if it can't ifnd a good suggestion with 5-grams.

3 Introduction to word representations

Vectorization, word representations, or wod embeddings, are a way to represent words or documents as vectors. Word vectors will be described in greater detail later int he course.
For now, we'll consider a very simple implementation based on the bag-of-words representation of a document.
With the sentence: *That that is is that that is not. Is that it? It is.*

1. NLTK allows us to create the following frequency distribution using FreqDist:
   - {'is': 6, 'that': 5, 'not': 2, 'it': 2}
2. By considering the words as their indices in the vocabulary...
   - $that \rightarrow 0, i \rightarrow 1, not \rightarrow 2, it \rightarrow 3$
3. The sentence can be represented as the values of each index:
   - $[5, 6, 2, 2]$

Answer the following:

1) *What are your thoughts on the usefulness of this representation? Can you think of a way to improve it?*
- Such a representation can be used to gauge how close in meaning/topic some sentences are. You can use different ways of calculating the distance between sentences in the n-dimensional room created by the dictionary used. I would say though, that I'm not too sure how helpful it is to count the words. It could maybe be better to just 1-hot encode them, or to not include stop words. In this case not including stop words wouldn't help us very much though.

2) *How could you use this tecnique to compare sentences with each other?*
- Like mentioned above, you can calculate the distance between them to gauge their relatedness.

4 Implementation notes

As a final part of the lab, I want you to briefly discuss your approach to the implementation task. This could be things you had to learn, difficult parts of the lab, etc.

- I found the nltk library to be very large, and quite confusing at times. While I felt like I had a straight ahead approach that I could probably have coded myself from scrathc"(of course at the loss of speed), but using the nltk functionalities made me use more different parts than I would normally like for doing something this simple. I ended up trying to simplify things a bit, in order to not meddle with too many functionalities of the package starting off.