Predicting proximity

Description

Tokens

NLP (natural language processing) systems donâ??t necessarily consider only words as the basic components of prediction. They may break words into smaller units roughly corresponding to syllables, punctuation marks and even single characters. The general term for these textual units is â??tokens.â?• An NLP system will include in its algorithmic workflow the construction of a lexicon (list) of tokens evident in its corpus. This lexicon may be derived from a statistical analysis of the occurrence of character sequences in the corpus, in some cases with the aid of a neural network (NN), optimised to classify tokens according to their properties (frequency, proximity, clustering, order) and to make predictions: given a string of text, what is likely to come next.

Identifying tokens in a body of texts (*tokenization*) is a nontrivial operation. As an example of a sophisticated NLP system, the tokens used in the ChatGPT-3 NLP system are stored in a lookup table (known as a *tokenizer*) that includes â??50,257 unique tokens.â?• There are fewer tokens than words in this universe, as the same tokens will crop up in many different words. Tokens may include words such as â??the,â?• â??and,â?• â??but,â?• â??then,â?• and â??a,â?• subwords such as â??in,â?• â??disc,â?• â??ret,â?• â??ion,â?• and special characters: â??,â?• â??.â?• â??!â?• â???â?• â??â??â?• â??-â??. The tokens in the lexicon will depend on the method deployed in the NLP training model and reflect the characteristics of the corpus. A lexicon derived from casual social media communications is likely to be different to one trained on government documents, or from a lexicon derived from both sources combined.

The tokens in the lexicon will be mapped on to numerical indices that bear no similarity to the character sequences they represent. This is common in data processing. It is the indices that get manipulated in the workflow. They are converted back to the tokens when ready to be presented as human-readable output. That said, for illustration it is easier to retain the use of tokens rather than indices. It is also easier in illustration to use whole words as if tokens.

Turning tokens into numbers

Algorithms work best with numbers. One such number indicates the closeness of words in terms of meanings, shared importance or relevance. For example, depending on the corpus, â??houseâ?• and â??gardenâ?• might be closer together than â??houseâ?• and â??rainbow.â?• Such relationships are not established by some pre-existing understanding of geography, physics, â??natural kinds,â?• typologies, or material evidence, but established by algorithmic analysis of the words in a particular corpus.

Data blogger Eligijus Bujokas provides a helpful example of how an NLP might quantify word proximity in a text corpus. The objective here is to see how certain words in a corpus cluster together. Iâ??ve attempted to translate his example to something spatial, a paragraph from a graffiti strategy for Edinburgh serves my purpose.

> Current procedures and guidelines have been reviewed and best practice identified to ensure that a balanced approach is taken. Robust policies and procedures on Graffiti Management are key components of the future strategy, aiming to reduce instances of â??tagging,â?• while still providing space for the more creative elements and potential benefits of graffiti, street art, and murals for local communities. This will also ensure that the cityâ??s residents and stakeholders are clear on the approach being taken by the local authority.

Lawrence, Paul. â??Graffiti Strategy for Edinburgh.â?• Culture and Communities Committee, September 15, 2020. Accessed March 21, 2021. Link, p.11.

Here is a lexicon of significant words from this simple corpus in alphabetical order and with their index numbers.

1. aiming
2. approach
3. art
4. authority
5. balanced
6. benefits
7. best
8. city
9. clear
10. communities
11. components
12. creative
13. current
14. elements
15. ensure
16. future
17. graffiti
18. guidelines

19. identified
20. instances
21. key
22. local
23. management
24. more
25. murals
26. policies
27. potential
28. practice
29. procedures
30. procedures
31. providing
32. reduce
33. residents
34. reviewed
35. robust
36. space
37. stakeholders
38. strategy
39. street
40. tagging

To cluster important words in this corpus, the NLP algorithm needs to identify a context for each word. Excluding ordinary words (â??theâ?•, â??aâ?•, â??beâ?•, etc) from the paragraph we can identify the context of any word (the focus word) as the list of words either side of that word up to a pre-specified number. We can arbitrarily set the context to be 1 word either side of a focus word. So, the context words for graffiti and procedures [focus word, context word] are as follows

[graffiti, procedures]
[graffiti, management]
[graffiti, benefits]
[graffiti, street]
[procedures, current]
[procedures, guidelines]
[procedures, policies]
[procedures, graffiti]

As shown in this subset of focus-context pairings, any word will assume the role of focus word or context word. Note that there is no information about the order in which words appear in the corpus. Thatâ??s a challenge for later discussion. We could already perform some simple numerical computation on this set of words via a network diagram and some method of counting, but a more general approach that scales to much larger corpora is to use a neural network.

Neural networks (NNs) are made up of a set of input nodes, a hidden layer of nodes and a set of output nodes. Each input node is connected to every hidden node, which are in turn connected to every output node. In my graffiti text example, every input node corresponds to a word in the lexicon (1-40).

Every output node also corresponds to a lexicon word (1-40). There can be any number of nodes in the hidden layer, though in his example Eligijus Bujokas uses just two nodes. As for my example, the context for any focus word is counted up to one either side. So thatâ??s just a context of 2. For Bujokas that hidden layer is simple enough and sufficient to illustrate clustering of words on a graph with 2 axes. Please refer to his example for the visualisation of word clusters.

During training, the NN algorithm cycles through all the focus-context pairings as inputs and outputs in the network, as values of 0 or 1. According to the NN model deployed, the network will adjust the weights of its connections and the values of the biases (thresholds) of its hidden nodes. So, one of the training examples will be input = â??graffitiâ?•, and output = â??proceduresâ?•, that is, it will assign 1 to input node 17, 1 to output node 30 and 0 to the other input and output nodes. (As it crops up in the NLP literature quite often, it is worth noting that this sparse binary presentation is known as *one hot encoding*.) Another input-output pair will be [graffiti, management], i.e. input node 17 and output node 23, etc. Iâ??ve not calculated the pairings precisely, but in my example the training set will consist of over 100 such input-output pairs.

Once so-trained, the network can be recruited to make predictions about the company any word keeps, words with which it is associated proximally. So, in test mode (i.e. running it to see what it has â??learnedâ?•), we present the network with a single input, e.g. â??graffitiâ?• and will expect it to generate an output consistent with its training set. Iâ??ll imagine here that the NN was trained on a larger corpus than shown here, with a correspondingly larger lexicon. So the word â??graffitiâ?• may occur in several places in the corpus, sometimes with a similar series of context words. In that case the output will be a series of words, such as â??proceduresâ?•, â??managementâ?•, â??benefitsâ?•, â??streetâ?•, â??artâ?•, each with a different calculated value, e.g. [2.1, 3.0, 2.0, 4.5, 1.0]. Through a simple procedure knows as *softmax*, that list of numbers can be converted to probabilities [0.06, 0.16, 0.06, 0.70, 0.02] that add up to 1.0. The neighbour of the word â??graffitiâ?• is most likely to be â??street,â?• but there are other possibilities as indicated by the list of probabilities.

So far, Iâ??ve considered an NN method for predicting where words might occur together in a corpus, but that does not yet yield a method for predicting or producing what words come next in a sequence. In a previous post I looked at Markov chains as providing an approach to sequencing relevant to decoding encrypted messages. See post: What comes next? But contemporary NLP generally employs other methods. A blog post by Stephen Wolfram provides a helpful explanation of word sequencing in NNs.

References

- Bujokas, Eligijus. â??Creating Word Embeddings: Coding the Word2Vec Algorithm in Python using Deep Learning.â?• *Towards Data Science*, March 5, 2020. Accessed 16 January 2023. https://towardsdatascience.com/creating-word-embeddings-coding-the-word2vec-algorithm-in-python-using-deep-learning-b337d0ba17a8
- Wolfram, Stephen. â??What is chatGPT doing and why does it work?â?• *Writings*, 14 February, 2023. Accessed 15 February 2023. https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

Note

- Featured image is generated by MidJourney from the prompt â??Graffiti strategy for Edinburgh.â?
  -

Category

1. Artificial Intelligence

Tags

1. graffiti
2. natural language processing
3. neural network

Date Created
February 18, 2023
Author
rcoyne99