

Twitter Thread by Alejandro Piad Morffis



Alejandro Piad Morffis

@AlejandroPiad



Hey folks, today is a Mindblowing Monday ■!

Today I want to tell you about Language Models, a type of machine learning techniques that are behind most of the recent hype in natural language processing.

■ Want to know more about them?

■■ 1/15

A language model is a computational representation of human language that models which sentences are more likely to appear in a given language.

■ Formally, a language model is a probability distribution over the sentences in a language.

■ What are they used for?

■ 2/15

■■ Language models allow computers to understand and manipulate language at least to some degree. They are used in machine translation, speech to text, optical character recognition, text generation, and many more applications!

They come in many flavors:

■ 3/15

■ The simplest language model is the *unigram model*, also called a *bag of words* (BOW).

In BOW, each word is assigned a probability P_i , and the probability of a sentence is computed assuming all words are independent. But of course, this isn't true.

■ 4/15

For example, "water" is a more commonly used word than "philosophy", but the phrase "philosophy is the mother of science" is arguably much more likely than the phrase "water is the mother of science".

■ The likelihood of a phrase depends upon all its words.

■ 5/15

■ This dependency can be modelled with an *n-gram model*, in which the likelihood of a word is computed w.r.t. the words before in a given phrase.

■ If we start a phrase with "philosophy", is more likely to see the word "science" than "shark".

■ 6/15

■ The problem with n-gram models is that the total number of parameters you need to store grows exponentially with n.

If you want to capture phrases of length $n=10$, you need N^{10} numbers, where N is the number of words in the language!

■ 7/15

■ Neural language models (aka continuous space models) are a solution to this exponential explosion.

They try to learn jointly a vectorial representation for all words (aka an embedding) and some mathematical operation among them that approximates the likelihood.

■ 8/15

■ Neural language models are built by training a neural network to predict some relationships between words and the phrases in which they appear.

The most popular neural language model is *word2vec*, trained in predicting a word given a small window around it.

■ 9/15

■ Modern neural language models have more complex neural network architectures.

Popular examples are BERT and the family of GPT models, of which GPT-3 recently took the Internet by surprise with its ability to speak nonstop about anything, often without much sense.

■ 10/15

■ The nice thing about language models is that they can be trained independently of any NLP problem and then used inside specific applications with a little fine-tuning.

■ 11/15

■ They also improve efficiency. A big company (like OpenAI or Google) can train a big language model and then the rest of us mortals can use them without having to pay millions in GPU training time.

■■ But they don't come without issues!

■ 12/15

■ Language models encode "common" language use, so all human bias is implicitly stored in them.

For example, the phrase "boy is a programmer" may be considered more likely than "girl is a programmer", simply because the Internet has more examples of the first.

■ 13/15

■■ If used without care, these language models will introduce subtle biases in your application that are very hard to discover and debug. Understanding and fixing these biases is one of the most exciting and important issues in AI safety!

■ 14/15

■ And that's it for today.

If you'd like to talk about language models, reply in this thread or [@](#) me at any time.

Feel free to ♥ like and ■ retweet if you think someone else could benefit from knowing this stuff.

■ 15/15