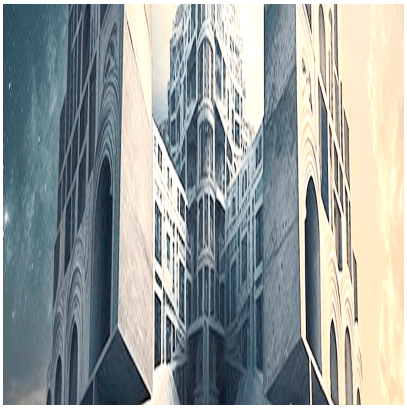

Architecture in multidimensional feature space

Description

In a previous post (Predicting proximity), I reviewed the NLP (natural language processing) operation of calculating the relationships between words in a corpus of texts. So the word â??architecturalâ?• is closer to the word â??urbanâ?• than â??architecturalâ?• is to â??culinary.â?• Very close word proximity could mean that one word can be substituted for another in certain cases (â??architectural designerâ?• = â??urban designerâ?•), or there is a high probability the words appear together in the same sentence: e.g. â??We teach architectural and urban design.â?•

â??Architecturalâ?• also occurs in a range of contexts that are different to one another: e.g. related to buildings, computer science, politics, management, etc. That ambiguity further complicates attempts to visualise the relationship between words. From an NLP point of view, words occupy a **multidimensional feature space**.

Word frequency and proximity are calculable statistically. But sophisticated language models such as ChatGPT use neural network techniques to infer from a very large corpus of texts the coordinates of any word from the NLPâ??s vocabulary in this multidimensional feature space. As a â??pre-trainingâ?• process, the NLP system extracts automatically its vocabulary of words from its vast training corpus. ChatGPT describes the process like this.

> Once the model has been trained, it generates a high-dimensional vector representation (i.e., a word vector or embedding) for each word in the vocabulary. These word vectors are optimized to capture the semantic and syntactic relationships between words, which allows the model to understand the meaning of words in different contexts. The dimensionality of these vectors can range from a few hundred to several thousand, depending on the architecture and hyperparameters of the model.

(Instead of â??understand the meaning ofâ?• I would say something like â??use.â?•) ChatGPT told me as an example, that after this pre-training the word â??architectureâ?• has a vector with 2,048 floating point numbers. I show some of the values here.

[0.36304411244392395, 0.07940052413988113, 0.35876873111724854, -0.15573939657211304, 0.19009003067016602, 0.39546394324302673, -0.5019744033813477, 0.4967377188205719, -0.10706250393390656, -0.29139858412742615, 0.31247848200798035, -0.05050345811295509, 0.336573392868042, 0.27102079939842224, -0.10432192623662949, 0.2255043385028839, 0.005103361814141512, 0.19811022233963013, 0.21053749346733093, 0.17883440828323364, 0.1537270691394806, 0.20455828380584717, 0.14289109432792664, -0.15890240621566772, -0.19201217532157898, 0.16197709703445435, 0.08864206004190445, -0.33965671038627625, 0.07935832411050797, -0.31531214785575867, -0.08488002479028702, 0.34383600997924805, -0.08132810807228088, -0.21734742832183838, 0.2759551405906677, 0.17740112555027008, -0.3020599489212036, 0.07422214722633362, 0.03535698708820343, -0.06683443462896347, 0.26701384711265564, -0.023962836399555206, 0.031534027725458145, 0.2045361694097519, 0.11502969229221344, -0.06030039873743057, 0.04954819333577156, -0.2791900930404663, 0.22967737901210785, 0.3221096091270447, -0.23879931843280792, -0.2961752119064331, -0.057812690764188766, 0.0276749560983181, -0.1900670087337494, 0.10122703731012344, -0.14712485694885254, -0.012389748148262024, -0.03714437749958038, -0.05026312953233719, â?¦

So, â??architectureâ?• is a point in a space of 2,048 dimensions. Each axis of this space is effectively a weighting attached to a node in a hidden layer of a neural network. There are 2,048 hidden network nodes in this model.

It would be hard for a human being to identify labels for these axes, but they constitute a multidimensional feature space shared across the entire vocabulary. Each word (or token) in the modelâ??s vocabulary is represented by a point in this space. The NLP model subsequently â??fine-tunesâ?• the feature space for a specific context of application, but this pre-training serves as a starting point for the machine learning process. The vocabulary stores each word, token and its feature vector.

Such vector â??embeddingsâ?• are saved in the modelâ??s vocabulary (lexicon) against each word. Though these unique embedding vectors are hard for a human being to analyse and interpret, a simple calculation would show the distances between any two or more words in this feature space. That might indicate how individual words tend to cluster according to: their presence in the training corpus, what words have similar meanings (contexts of use), similarities, represent opinions and emotions (sentiment), and it would serve to classify sections of texts in various ways.

After offline pre-training and fine tuning, these vectors are used during query analysis (inference) to predict, and thereby generate new text.

> When the language model processes a text input, it maps each token in the input to its corresponding vector in the lexicon and uses those vectors to calculate the probabilities of the next word or token in the sequence.

ChatGPT further indicates the advantage of this method in accounting for the remarkably fast response as users communicate with the platform (during inference).

> Once the model has been pre-trained and fine-tuned, the vocabulary of the model stores each word, token, and its corresponding feature vector. This allows the model to quickly look up the feature vector for any given word or token during inference, which is an efficient way of representing and processing text data.

The neural network used during inference has as many output nodes as there are tokens in the modelâ??s vocabulary, in the order of 50,000. There are as many input nodes as there are dimensions for the word vectors (e.g. 2,048).

During inference, the model looks up the vector for each word in the sequence, and outputs a probability that each token in the vocabulary will appear next. That may be the most likely token, or another token sampled on the basis of a pre-set randomising factor. By most accounts, NLP systems donâ??t always select the best next word in a sequence. Thatâ??s to avoid simply repeating the same optimally predicted word sequences during every interaction with the platform.

The concept â??word embeddingsâ?• crops up frequently in explanations of advanced natural language processing models, especially in discussions of â??attention,â?• for later discussion.

Bibliography

- 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 from MidJourney prompted by â??Architecture in multidimensional feature space.â?•

Category

1. Artificial Intelligence

Tags

1. embeddings
2. natural language processing
3. vectors

Date Created
April 15, 2023
Author
rcoyne99