NLP2$_lab8$

thibaut.benefice lucas.pinot victor.litoux hicham.madi

November 23, 2022

1 What is the purpose of subword tokenization used by transformer models?

Subword tokenization attempts to split words into smaller morphemes. For example the word "unbelievably" could be split into the morphems "un", "believ", "abl" and "y" or some variation of these. The idea behind this is mostly two-fold: (a) it enables an overall reduction in the size of vocabularies and also (b) helps recognize components of words that are not in our initial vocabulary once in production/evaluation. (a) is especially true in situations where we have a large initial vocabulary. An example for (b) would be the following: say at training we found the word "unbelievably", in our initial vocabulary, the whole word be memorized. However, if the model sees the word "believe" at inference, it is highly likely it would be considered unknown. In the situation where we use sub word tokenization, "believe" would become "believ" and be recognized by our model and therefore useful in our semantic analysis.

2 When building an encoder-decoder model using an RNN, what is the purpose of adding attention?

In the case of RNN based encoder-decoder models, attention is mainly useful for the decoder step. Because of the way RNN work, whether they are uni or bidirectional, a bias will exist towards certain words of our input irrelevant to their semantic importance. This means that if a word is at a certain position in the input (mostly at the start or the end), its importance to the decoder will increase, even if other words are of higher semantic value. The attention mechanism enables the decoder to access information of all hidden states independantly of the position in the input and gives us information on the relevance of the words to one another.

3 In a transformer model what is the multi head attention used for?

Self-attention is a mechanism in which we try to find how relevant each word of a given input are to one another. However self-attention represents relationships of a single type which can be insufficient to fully represent the semantic meaning of a sentence. Multi-headed attention is basically an upgrade of the self-attention method. Multi-headed attention basically concatenates multiple self-attention layers for the same input and projects those results in single representation space. This process make it possible to embrace multiple types of relationships between words.

4 In a transformer model, what is the purpose of positional embedding?

In languages, the order of the words and their position in a sentence really matters. The meaning of the entire sentence can change if the words are re-ordered. When implementing NLP solutions, recurrent neural networks have an inbuilt mechanism that deals with the order of sequences. The transformer model does not use recurrence or convolution and treats each data point as independent of the other.

The positional information is added to the model explicitly to retain the information regarding the order of words in a sentence. Positional encoding is the scheme through which the knowledge of the order of objects in a sequence is maintained.

5 What are the the purpose of benchmarks?

Benchmarks are basically universally accepted datests/metrics/tests used to evaluate models. These are necessary to compare the performance of different models and trust their results. This also makes it possible to track progress over the years of certain tasks and compare them to human level capacity. This is extremely useful if we wish to automatize tasks that could only be performed by humans so far. However, just as it is when using a dataset for training, there is always the risk that a bias is present in our dataset. Therefore, although benchmarks are a necessary staple to research and improvement of the field, they are also subject to change as some issues may be discovered further down the line. Also, the more models improve, the harsher they should be tested. This means that as the domain progresses, tasks that were deemed to complex initially can and will be used as benchmarks further down the line. A good example of this evolution in benchmarks is how superglue tried to upgrade glue.

6 What are the differences between BERT and GPT?

They are the same in that they are both based on the transformer architecture, but they are fundamentally different in that BERT has just the encoder blocks from the transformer, whilst GPT-2 has just the decoder blocks from the transformer.

GPT-2 works like a traditional language model is that it takes word vectors and input and produces estimates for the probability of the next word as outputs. It is auto-regressive in nature: each token in the sentence has the context of the previous words. Thus GPT-2 works one token at a time. BERT, by contrast, is not auto-regressive. It uses the entire surrounding context all-at-once.

GPT-2 is trained in the standard transformer way, with a batch size of 512, a well-defined sentence length, and a volcabulary size of 50,000. At evaluation time, the model switches to expecting input one word at a time. This is done by temporarily saving the necessary past context vectors as object properties.

BERT gets around this by modifying the task. Whereas GPT-2 learns on the "predict next" task directly, BERT learns on the task "learn the word in a sentence in which 15% of the words are masked out". The masking is a form of regularization; it withholds just enough at preventing the algorithm from cheating through rote memorization. It will also rarely replace a word with a different word.

7 How are zero-shot and few-shots learning different from fine-tuning?

Finetuning means taking weights of a trained neural network and use it as initialization for a new model being trained on data from the same domain (often e.g. images). It is used to speed up the training and overcome small dataset sizes.

There are various strategies, such as training the whole initialized network or "freezing" some of the pre-trained weights (usually whole layers).

The common practice for machine learning applications is to feed as much data as the model can take. This is because in most machine learning applications feeding more data enables the model to predict better. However, few shot learning aims to build accurate machine learning models with less training data.

Zero-shot learning is that at test time, a learner observes samples from classes which were not observed during training, and needs to predict the class that they belong to.

8 In a few paragraphs, explain how the triplet loss is used to train a bi-encoder model for semantic similarity?

Bi encoders are models used mostly in information retrieval. They are composed of two main sections and a head for which the structure changes depending on if we are training or inferring on the model. The first main section takes a query as input (i.e a google research a user inputs), gives its embedding and passes through an encoder (BERT for example). This allows us to obtain a semantic representation of the query. The second main section follows the same steps but applied on the document from which we are trying to extract. Then we apply a function that outputs the semantic similarity of the input query and the document which allows to rank various documents depending on how well they answer our query.

The triplet loss is one of the methods used for training such a model. This method is composed of three elements: an anchor (a query) and a negative and positive. The negative and positive are two possible "answers" that could be given to the query but the positive is the "good" answer while the negative is the "wrong answer". These answers are clearly labelled as such and the objective at training would be to minimize the distance between the anchor's encoded value and the positive which and maximize the anchor's encoded value with the negative.