Synergy

by
J. Felix King

Professor Mark Hopkins, Advisor

A thesis submitted in partial fulfillment
of the requirements for the
Degree of Bachelor of Arts with Honors
in Computer Science

Williams College
Williamstown, Massachusetts

December 2, 2024

Chapter 1

Introduction

1.1 Motivation

Machine translation tasks rely heavily on the availability of parallel training data – sentences in one language and their translations in the other. For many langauge pairs, hundreds of thousands of high-quality sentence pairs are available. But there are hundreds of languages which have only been translated into one or two other languages, if any at all. Any translation task involving one of these "low-resource" languages (ASK: can we call languages low resource or just language pairs?) is doomed to produce a low-quality result at best.

[?]

Chapter 2

Background

2.1 Neural Machine Translation and the Transformer Model

Neural networks can be adapted for many different tasks, and those tasks can be categorized under labels which describe the type of input and the type of output. For example, spam detection would fall under "sequence classification." Translation is a sequence-to-sequence task, because the input sequence (e.g. a sentence in French) is being used to generate an output sequence (e.g. an English translation of that sentence). The input and output sequences can and often do differ from one another in length, in script, and in sentence structure.

How would it work to individually translate words in a source sentence and combine them according to the target language's syntax rules? It would often provide a terrible translation, not just because different languages can communicate the same information in more or fewer words than one another; words cannot be translated individually because their meanings depend on their context. The same preposition, "in," may correspond to any of several German prepositions – "in" or "auf" or "bei" – depending on the context. Thus effective translation of a single word in context may require use of an entire sentence or more than one sentence.

something in here about early NMT methods

The Transformer model (2017, cite) innovated on the encoder-decoder structure (cite), and it is still the standard for NMT architecture and other applications. It uses an attention (cite attention paper?) and feedforward layers in place of recurrent components. This new architecture not only has features suitable to many linguistic tasks, but trains much more quickly than the alternatives as it allows for far more parallel computation in training.

In the following section, I'll describe at a high level how a Transformer model translates a sentence.

Consider the English sentence "The man wearing black shoes rode his bicycle."

Step 1: Tokenize

The sentence must first be broken down into small parts for the model to work with. This is

done with the tokenizer, which is often itself a trained component of the model. The tokenizer's vocabulary contains enough words and sub-words to cover the languages the model takes as input. Thus the tokenizer turns the sentence into a list of tokens, which are all elements in the tokenizer's vocabulary. Here is the result of tokenizing the above sentence using the tokenizer in Facebook's NLLB models:

[256047, 1617, 492, 214030, 49154, 203020, 134457, 4414, 330, 163731, 248075, 2]
['eng_Latn', 'The', 'man', 'wearing', 'black', 'shoes', 'rode', 'his', 'bi', 'cycle', '.', '</s>']

The numerical array just desplays the indices of the tokens in the tokenizer's vocabulary. Note '</s>,' which did not appear in the original sentence; it is a special "end of sentence" token which the tokenizer adds to every input.

Step 2: Encode

In its first layer, the encoder deterministically turns each token into a dense vector embedding, then implicitly marks the position of each token in the sentence by adding a unique sum of sinusoids. The embedding vectors corresponding to each token in the vocabulary are learned in training, and the model learns the implicit meaning of the signal added by the positional encoding. The rest of the layers of the encoder repeatedly use self-attention to contextualize each embedding in the sentence. The embedding of a verb might be strongly influenced by the noun it is acting on, or the embedding of an article in a gendered language might depend primarily on the gender of its referent, for example.

The vectors produced by the encoder are high dimensional and therefore impossible to accurately plot. But their components represent the relationship between words and individual tokens. The cosine similarity (angle) between vectors can represent analogous relationships between pairs of words. For example, (insert an example here)