Dialogue generation with transformers

Shivangi Khandekar, Josep Rubió Piqué
UAB, Natural Language Processing

Abstract

- Try to create an AI agent giving meaningful responses on dialogues.
- Use transformer and self-attention architecture instead of RNN.
- Trained with OpenSubtitles dateset.

Dataset

OpenSubtitles is an open corpus with movies subtitles in different languages.

The dataset used to train the agent are the english subtitles from OpenSubtitles:

- $\sim$ 140M utterances
- Dirty dataset. Metadata from the movie, comments not related to dialogues, ...
- Scripts from PolyAI used to clean the dataset
- Trained with just the previous sentence in the context.

```
{
"file_id":  "lines-aaa",
"context":  "They always do.",
"response":  "Hmm, but they don't.",
"context/0":  "He will.",
"context/1":  "What then?",
...
}
```

Attention is all you need

- Traditional **Encoder-Decoder** architecture without RNN or convolutional networks
- Based on attention concept. Learning of Query (**Q**), Key (**K**) and Value(**V**) matrices
- **Stacked transformers** for encoder and decoder and **multi head** attention to *attend to different information*
- **Positional encoding** to give relative position information to tokens. In the paper a sin function added to embedding
- For decoder an extra **masked attention** layer is used to learn the language model. It cannot look forward so it masks attention for not generated tokens

Transformer Architecture

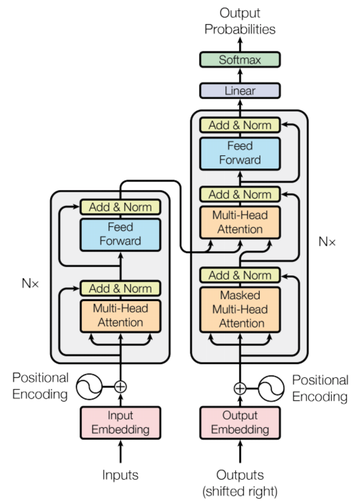
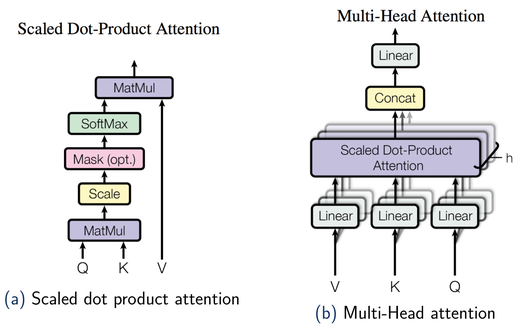


Figure: Transformer architecture



(a) Scaled dot product attention (b) Multi-Head attention

Figure: Attention

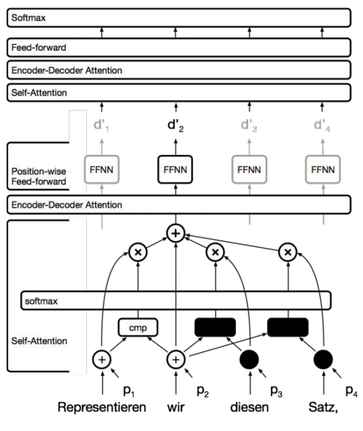


Figure: Decoder architecture

Transformer implementation and outcomes

Implementation based on *Transformer model for language understanding* from TensorFlow.

- Subword tokenizer from TensorFlow Dataset library
- Token embedding using Keras embedding layer
- 4 encoder layers
- 4 decoder layers
- Sublayers and embedding outputs of size 128
- Feed Forward inner layer of size 512
- 8 attention heads

Small train dataset

Trained with 78.000 utterances and 20 EPOCHS. 2 hr in a personal computer:

- **Context:** *Hello I am John.*
- **Predicted response:** *I'm not gonna be a doctor.*
- **Context:** *Who are you?*
- **Predicted response:** *I am a bit much.*
- **Context:** *I want to eat a sandwich*
- **Predicted response:** *I want to be a doctor!*

Big train dataset

Trained with $\sim$ 2M utterances and 10 EPOCHS. 10 hr in the cloud and a GPU:

- **Context:** *My name is John.*
- **Predicted response:** *I'm sorry..... Achi. Achi.....*
- **Context:** *Good morning.*
- **Predicted response:** *Where are you??????????????????*
- **Context:** *I am in my car*
- **Predicted response:** *What the fuck are you talking about???????????*
- **Context:** *I am going to the hospital*
- **Predicted response:** *What the fuck are you talking about???????????*

Big train dataset

Trained with the recommended architecture from the paper. The same but **6** encoder and decoder layers, outputs of size **512** and FF inner layer of size **2048**. The response is the same for all contexts.

Trained with $\sim$ 2M utterances and 3 EPOCHS. 9 hr in the cloud and a GPU:

- **Context:** *My name is John.*
- **Context:** *Good morning.*
- **Context:** *I am in my car*
- **Context:** *I am going to the hospital*
- **Predicted response:** *I don't know what you're talking about. about.with.with.with.about.with.with.with.with.*

Evaluation

The main way to evaluate the results in dialogue generation is with human judgement, comparing the result of the agent with other responses generated from other agents. I cold not evaluate my results this way.

Conclusion & Future Work

- It seems the results keep getting worst as it gains more information.
- Underfitting with less information?
- It seems the model converges to give dull responses, which could make sense from a dialogue prospective if a similar sentence to the context does not exist in the dataset
- The model learns a "language model" that creates responses grammatically correct fast but it does not give meaningful responses in the dialogue.
- Hard to deal with big datasets for time and space constraints.
- Training the agent with a big dataset with the recommended architecture properly.

References

- *Vaswani, Attention is all you need*
- Tensorflow Transformer model for language understanding `https://www.tensorflow.org/tutorials/text/transformer`
- OpenSubtitles corpus `http://opus.nlpl.eu/OpenSubtitles-v2018.php`
- Poly AI *Conversational Datasets* `https://github.com/PolyAI-LDN/conversational-datasets`