MFDS Project

Aditya Das, Aryan Pandey , Dev Panghate, Abhiram Santosh, Shashank Shekhar Singh

June 2022

1 Introduction

In this project we are to rate the similarity between two passages. Its use case is to automatically grade exams. A template passage will be provided against which our algorithm must compare the answers from different students and yeild a score telling how similar they are. Based on this score, an automatic grading system can be set up. Our approach to solve this problem is given below.

2 Our Approach

For solving this problem, we thought of representing each passage with a vector which would contain the semantic meaning of the passage. On vectorizing the template and the test passage, we could then use any metric of distance to fine the distance between the vectors. This distance on normalizing will represent the similarity score.

For vectorizing the passage, we could either use a Recurrent Neural Network or a Transformer mechanism. These Deep Neural Network based methods were preferred because they could better capture the meaning of the sentence. The problem with Recurrent Neural Networks is that they cannot handle long sentences. The information stored after processing the entire sentence is mostly concentrated towards the end of the passage. Training RNNs are also trickier with long passages.

Hence we decided to go with a transformer based architecture. The advantage of the transformer architecture is the the attention mechanism. This mechanism helps encode the meaning of each token or word with respect to other words and since a word can have many meanings, it is able to capture the right one. Also since it is not a sequential model, the problem of forgetting will not happen with long passages. The only downside is that the computation is of $O(n^2)$ complexity, so the computation power required will be very high if we expect results in reasonable time. But luckily there are plenty of pretrained models available to use.

3 Training Method

Due to the computational power required to train a transformer, we decided to use a pretrained sentence transformer. As its clear from the name, a sentence transformer is used to vectorize a sentence. It does this by calculating a value vector for each word in the sentence and then averages the vector representation of each word. We compute the vector for each senctence in our passage using this transformer and average it to find a normalised vector representation for our passage.

The similarity metric we chose to use is cosine similarity. This is because the output is always constrained in a fixed range [0,1] and the similarity between two passages can be interpreted independently. The formula is given by

$$cos(\vec{a}, \vec{b}) = \frac{\vec{a}.\vec{b}}{|\vec{a}||\vec{b}|} \tag{1}$$

For passages that have a similar meaning, we expect that their vector representations will be closeby in the vector space and so will have a high cosine similarity score. Similarly for passages with different meanings, we expect a low score.

4 Testing and Results Obtained

For testing the model's performance, we manually made our dataset of 10 data points. First we gather a passage of any random domain (eg cooking, sports etc). Then we either manually tweak and rearrange the sentences or use a paraphrasing tool to do the same. By doing so we try to ensure that the rough meaning of the passages remain the same but there are clear distinctions between the writing style. We expect these pairs to have a high cosine similarity score.

For testing passages with different meaning, we just gather two passages across domains and pass it through our algorithm. We expect this score to be low.

Consider the following 10 passages

A1: Tomato are a popular garden crop for their high productively and rich flavors, and learning to grow them well includes understanding how to prune the plants. Fear of tomato pruning mistakes holds many gardeners back from trimming their tomato plants. But this is a task worth learning as properly pruned tomato plants are healthier, bothered by fewer fungal diseases, and more productive. Keep reading to get our in-depth advice on common tomato pruning mistakes to avoid.

A2: Tomatoes are a popular garden crop because of their high yield and quality, and knowing how to trim the plants is an important part of growing them successfully. Many gardeners are hesitant to prune their tomato plants because they are afraid of making mistakes. But it's a skill worth knowing since correctly trimmed tomato plants are healthier, have fewer fungal illnesses, and produce more fruit. Continue reading for more information on how to prevent typical tomato trimming blunders.

B1: There are many reasons to prune tomato plants. Pruning helps the gardener direct growth and control the mature size of the plant, as well as fit more plants into the garden, maximizing space. Tomato pruning to promote airflow can boost plant health and reduce the occurrence of diseases. Removing unnecessary growth also prevents over-crowding and permits lots of sunlight to reach the center of the plant. Pruned plants may crop earlier than unpruned plants and produce larger fruits.

B2: Tomato plants should be pruned for a variety of reasons. Pruning allows the gardener to regulate development and manage the plant's mature size, as well as fit more plants into the garden and maximise space. Tomato pruning that promotes airflow can improve plant health and minimise disease incidence. Removing unwanted growth also reduces overcrowding and allows plenty of light to reach the plant's core. Plants that have been trimmed may produce bigger fruits and crop earlier than those that have not been pruned.

C1: Tomato suckers are sideshoots that grow from the angle where the leaf of a tomato meets the stem. Tomato suckers produce flowers and eventually fruits, but allowing all the suckers to grow on an indeterminate tomato isn't a good idea. It results in a massive, overgrown plant that's difficult to support, but is also more prone to insect problems and plant diseases.

C2: Tomato suckers are sideshoots that sprout from the point where a tomato's leaf joins the stem. Tomato suckers generate flowers and ultimately fruits, but it's not a good idea to let all of them develop on an indeterminate tomato. It produces a gigantic, overgrown plant that is not only difficult to sustain, but also more susceptible to pest and disease issues.

D1: But through the winter I miss the homegrown food and the promise of food to come. I miss green beings emerging from the soil and the smells. My god, a city winter is mostly scentless. There's a reason why they don't make dirty snow, previously frozen dog poop, and idling car exhaust into scented candles. I start to forget about earthy smells, and then when I remember, the longing is so deep that it aches in my belly — the branches and bits of green things, and the hundreds of potted plants I bring indoors just don't seem to cut it.

D2: However, I miss the homegrown food and the promise of more to come throughout the winter. I miss the fragrances and the green things sprouting from the dirt. A metropolitan winter is basically scentless, my goodness. There's a reason why they don't produce fragrant candles out of filthy snow, previously frozen dog excrement, or idle automobile exhaust. I begin to forget about earthy fragrances, and when I do, my stomach hurts with desire - the branches and pieces of greenery, as well as the hundreds of potted plants I bring home, just don't seem to cut it.

E1: It is so much quieter now in my city backyard, which is often full-on during the "gentler" seasons with stress-barking dogs, and stress-barking humans, and machines (a neighbour has an outdoor saw and a day doesn't go by when he isn't sawing for hours at a time). The quiet is nice — a sure sign that I am getting older, or perhaps just more self-aware, and when I go into the garden it is just me, my dog, the occasional bird, and some street noise that I almost don't hear anymore having lived off of major urban streets my entire adult life.

E2: It's much calmer now in my city backyard, which is typically crowded during the "gentler" seasons with stress-barking dogs, stress-barking humans, and machinery (one of my neighbours has an outside saw, and there's seldom a day when he isn't sawing for hours at a time). The calm is pleasant — a solid indicator that I'm getting older, or perhaps just more self-aware — and when I go into the garden, it's just myself, my dog, the occasional bird, and some city noise that I almost don't notice any longer, having spent my whole adult life living off of big metropolitan streets.

The following table summarises the results of similarity across the passages. Note that the passages whose labels start with the same label are expected to be similar

Passage 1	Passage 2	Similarity Score
A1	A2	0.905
B1	B2	0.986
C1	C2	0.959
D1	D2	0.962
E1	E2	0.919
C1	E2	0.321
A1	E2	0.226
B1	E2	0.253
B1	D2	0.311
C1	D2	0.376

To reproduce these results first create individual .txt files for each of the passages. Then run the command "python Group_13.py (path to txt1) (path to txt2)". It will print the similarity score

5 Contributions

Participants are

- Aditya Das ME19B194
- Aryan Pandey NA19B030
- Dev Panghate MM19B059
- Abhiram Santosh CH19B037
- Shashank Shekhar CE20B101

Aditya Das and Dev Panghate are responsible for ideating the model to use and the advantages and disadvantages of different models. Aryan Pandey, Abhiram Santosh and Shashank Shekhar are responsible for writing the code and creating the dataset. Everyone contributed equally for the report creation