Udacity “AWS ML Engineer nanodegree”
Capstone Project Proposal
Denis Serbin

- ## Domain background

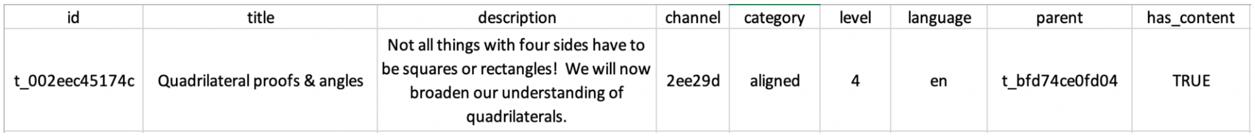
My capstone project is motivated by the Kaggle competition “Learning Equality - Curriculum Recommendations”, which focuses on building efficient ML models that could match educational content (files and videos in all kinds of formats) to curriculum (K-12) topics. Both content and topics have text descriptions in various languages, so every successful model is going to have a substantial NLP component in it.

The problem addressed in the competition is very important in education. I am exposed to it myself since I’m a university professor (mathematics). Together with my colleagues, we’ve been trying to deal with the problem of matching content (lecture notes, videos, and problem formulations) with topics in various math courses (calculus, linear algebra, differential equations, etc.). The matching was mostly done manually, it was very tedious and time consuming. Now, using powerful pretrained language processing models, I believe it is possible to automate the matching process.

- ## Problem statement

Input:

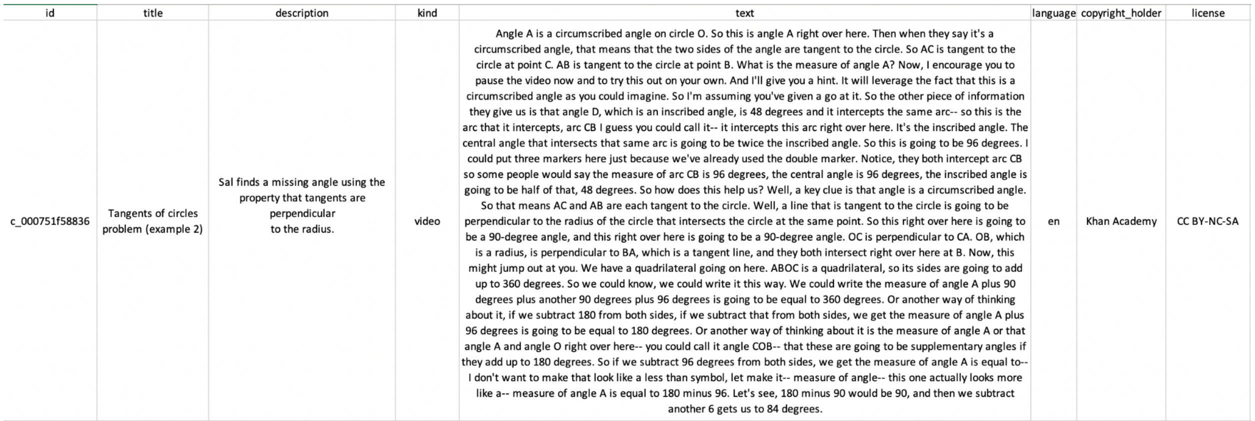
1. A list of topics from K-12 curriculum.
   Each topic has an id and several description fields (see below):



id	title	description	channel	category	level	language	parent	has_content
t_002eec45174c	Quadrilateral proofs & angles	Not all things with four sides have to be squares or rectangles! We will now broaden our understanding of quadrilaterals.	2ee29d	aligned	4	en	t_bfd74ce0fd04	TRUE

   All topics are organized in a tree so that each topic belongs to a branch of the tree and it “knows” its parent.

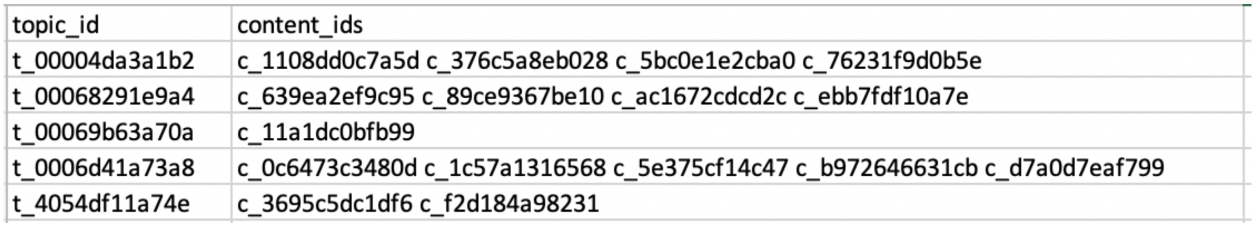
2. A list of content items (files in various formats).
   Every content item also has an id and several description fields:



id	title	description	kind	text	language	copyright_holder	license
c_000751f58836	Tangents of circles problem (example 2)	Sal finds a missing angle using the property that tangents are perpendicular to the radius.	video	Angle A is a circumscribed angle on circle O. So this is angle A right over here. Then when they say it's a circumscribed angle, that means that the two sides of the angle are tangent to the circle. So AC is tangent to the circle at point C. AB is tangent to the circle at point B. What is the measure of angle A? Now, I encourage you to pause the video now and to try this out on your own. And I'll give you a hint. It will leverage the fact that this is a circumscribed angle as you could imagine. So I'm assuming you've given a go at it. So the other piece of information they give us is that angle D, which is an inscribed angle, is 48 degrees and it intercepts the same arc-- so this is the arc that it intercepts, arc CB I guess you could call it-- it intercepts this arc right over here. It's the inscribed angle. The central angle that intersects that same arc is going to be twice the inscribed angle. So this is going to be 96 degrees. I could put three markers here just because we've already used the double marker. Notice, they both intercept arc CB so some people would say the measure of arc CB is 96 degrees, the central angle is 96 degrees, the inscribed angle is going to be half of that, 48 degrees. So how does this help us? Well, a key clue is that angle is a circumscribed angle. So that means AC and AB are each tangent to the circle. Well, a line that is tangent to the circle is going to be perpendicular to the radius of the circle that intersects the circle at the same point. So this right over here is going to be a 90-degree angle, and this right over here is going to be a 90-degree angle. OC is perpendicular to CA. OB, which is a radius, is perpendicular to BA, which is a tangent line, and they both intersect right over here at B. Now, this might jump out at you. We have a quadrilateral going on here. ABOC is a quadrilateral, so its sides are going to add up to 360 degrees. So we could know, we could write it this way. We could write the measure of angle A plus 90 degrees plus another 90 degrees plus 96 degrees is going to be equal to 360 degrees. Or another way of thinking about it, if we subtract 180 from both sides, if we subtract that from both sides, we get the measure of angle A plus 96 degrees is going to be equal to 180 degrees. Or another way of thinking about it is the measure of angle A or that angle A and angle O right over here-- you could call it angle COB-- that these are going to be supplementary angles if they add up to 180 degrees. So if we subtract 96 degrees from both sides, we get the measure of angle A is equal to-- I don't want to make that look like a less than symbol, let make it-- measure of angle-- this one actually looks more like a-- measure of angle A is equal to 180 minus 96. Let's see, 180 minus 90 would be 90, and then we subtract another 6 gets us to 84 degrees.	en	Khan Academy	CC BY-NC-SA

The set of content items is not structured.

Output: an alignment of the given content items with the given topics (see below).



topic_id	content_ids
t_00004da3a1b2	c_1108dd0c7a5d c_376c5a8eb028 c_5bc0e1e2cba0 c_76231f9d0b5e
t_00068291e9a4	c_639ea2ef9c95 c_89ce9367be10 c_ac1672cdcd2c c_ebb7fdf10a7e
t_00069b63a70a	c_11a1dc0bfb99
t_0006d41a73a8	c_0c6473c3480d c_1c57a1316568 c_5e375cf14c47 c_b972646631cb c_d7a0d7eaf799
t_4054df11a74e	c_3695c5dc1df6 c_f2d184a98231

One topic can be aligned with several content items. At the same time, a content item can be aligned with multiple topics as well

- Solution statement

A solution to the stated problem obviously exists: the curriculum alignment can be done manually, by a team of experts. In particular, one of the data files for the competition (called “correlations.csv”) contains the alignment of the topics and content items in the training set. This file basically provides labels for the training set.

A desirable solution would be an ML model that, given the input described above, produces an output with accuracy close to the manual solution that could be produced by human experts.

I am planning to train my model(s) within the provided Kaggle environment. This makes sense since they allow to use free but limited GPU acceleration. If Kaggle resources are not enough, then I am going to use my AWS account as the platform.

In my model I am going to apply knowledge transfer. I am planning to use a pre-trained SentenceTransformer model (for example, ‘paraphrase-multilingual-mpnet-base-v2’) that maps sentences into a high-dimensional vector space. The model can be fine-tuned to better fit the context of the problem (we already know connections between given topics and content items, so the pre-trained model can be improved using the connections).

Datasets and inputs

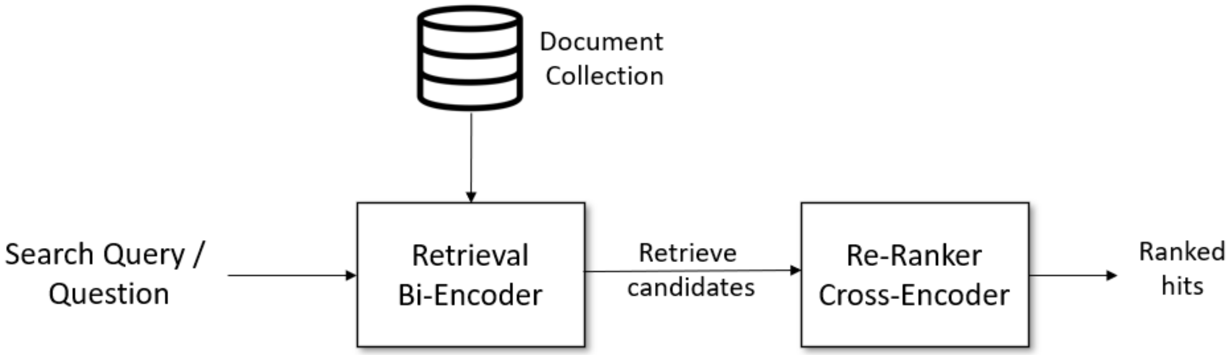
The training dataset is given in three files:

1. “topics.csv” - curriculum topics given in the form shown above,
2. “content.csv” - content items with descriptions (a sample item is shown above),
3. “correlations.csv” - an alignment of the topics with the content items.

All the files can be downloaded from the competition page.

Benchmark model

The stated problem can be viewed as an *information retrieval problem.* Namely, given a query (a topic in our case), one has to find all relevant documents from the list (given content items). Typically, the process can be described as follows (the image is copied from https://www.sbert.net/examples/applications/retrieve_rerank/README.html):



In particular, one could finetune a pre-trained SentenceTransformer model to embed the given topics and content items (using their description features) into vectors and then perform matching based on the distance of the corresponding points in the vector space.

I am not aware of a specific benchmark model that could be immediately applied to the stated problem. Since the problem comes from an active Kaggle competition, a publicly available model (with a low competition score, of course) by one of the competitors could serve as a benchmark model. For example, this one https://www.kaggle.com/code/takamichitoda/lecr-simple-unsupervised-baseline

- Evaluation metrics

The competition "Learning Equality - Curriculum Recommendations" uses the F2 metric

$$F2 \ = \ \frac{5 \cdot Precision \cdot Recall}{4 \cdot Precision + Recall}$$

where

$$Precision \ = \ \frac{true\,positives}{true\,positives + false\,positives}$$

$$Recall \ = \ \frac{true\,positives}{true\,positives + false\,negatives}$$

I am going to use the same metric.

- Project design

As I mentioned above, I would like to try the "Retrieve and Re-rank" approach outlined in https://www.sbert.net/examples/applications/retrieve_rerank/README.html
I am going to

- finetune a pre-trained SentenceTransformer model to embed the given topics and content items into vectors,
- then for each topic (viewed now as a point in the vector space) create a list of all "close" content items (also viewed as points in the same space), and
- choose some number of the closest content items from the list.