March 2, 2022

The goal of the project is to turn a collection of my favorite questions into a question graph based on NLP-identified similarity of questions.

Gathered the top questions, from Tim Ferriss, Daniel Schmachtenberger, and Ben Franklin and placed them into a csv file, which is now over 100 questions. Sufficiently large to be a dataset to be used.

We decided pretty quickly to focus on personal development questions, as they have the most relevance to my life right now. We plan to share the code to the rest of the world so people can plug in their questions too.

The next step for now is to figure out what sentence similarity algorithm we will use. It looks like BERT is the right way to go. Link is [here](https://towardsdatascience.com/bert-for-measuring-text-similarity-eec91c6bf9e1). There is a simple python package you can use called sentence-transformers, which you can use. Explainer link is [here](https://huggingface.co/sentence-transformers). Actual link to the library is [here](https://pypi.org/project/sentence-transformers/). What we will do is download this library, and learn how to use it. Ideally we’ll use a pre-trained transformer and turn our questions into a vector space. We can then either create a KNN graph for Neo4J, or we can make a UMAP from the vector space. I think the Neo4J option sounds a bit sexier. Then we can make it into a website. We could maybe even open source the code so people can make their own question graphs, or add to the code.

March 3, 2022

My goal for today is to make this an emacs project. I’m going to seriously try emacs. This might have everything I need.

We got the graph working! This is fantastic. Now we’re going to try to throw it on Neo4J. We note that we’re doing a hard threshold. In other words, we’re using a Boolean based adjacency matrix. This means that there are some questions that are not connected because they’re not sufficiently related. If I set the threshold too low, you get everything connected to everything (or at least it seems that way from the python notebook display of the igraph object…this might change when we throw it on Neo4J).

March 4, 2022

Now we’re going to throw the graph onto Neo4J by exporting it as an edgelist. But first, we’re going to do some housekeeping. We’re going to set up the virtual environment for the python project. We could do this easily doing visual studio code, but we are experimenting with using emacs as a way to make me better at engineering, the same way manual transmission made me a better driver (so it feels like).

Got the graph! I had made a mistake and had actually made questions that were the LEAST related because I confused cosine similarity with cosine distance. But I found the mistake by means of sanity check.

I placed the graph into Neo4J and I did the usual centrality analysis and clustering (Louvain) analysis. With respect to centrality, I hypothesize that the most central questions are the ones that lead to the others. In other words, if you study the most central questions, you’ll get to the others by inference. With the clustering, I found clusters that have various themes. One cluster was an “emotion” theme and another was a “virtue” theme. Yet another was a “negative sentiment” theme. There were only a small handful of clusters that were sizeable enough to have any sort of possible meaning.

What I intend to do from here:

* Be a little more critical about the input data. The current dataset is nice because we get to see the crossover between, for example, Tim Ferriss’s questions and Daniel Schmachtenberger’s questions (we can program that into Cypher). But we also would like to see exclusively how Tim’s questions are connected and how Daniel’s questions are connected separately.
* Add more data. Tony Robbins and other business coach-like people have fantastic questions that are worth putting in. Note that the general theme for these questions is personal development. We note that other networks of questions might revolve around, for example, unsolved problems in biology or anything else like that.
* Vary the distance metric. We’re doing cosine right now. Maybe Euclidean would be worth it.
* Do a UMAP + plotly interface. It’s worthwhile to see who is near who and how near.

Now how do we put this out in front of people?

* Make a website where the questions are uploaded and a similarity graph is produced. Open source the project so others can contribute code to improve it.
* Upload this to HealthECCO and utilize the community there for the sake of visibility. They’re Neo4J centric. This is probably the best path forward actually.

How do we monetize this?

* I don’t think people will PAY for this, but they might ask me to do similar things for them, paid.
* Add the NLP element to the current graph work I’m doing for Alkahest to connect to abstracts and that kind of thing. Abstract or title analysis for example is a real possibility, now that I know how to use the BERT model.
* Read about other applications of BERT and see if I get any ideas. NLP in general. See how NLP is being applied to biology and bioinformatics in the first place.
* Revisit some of my old PubMed harvesting work, and see if I can add to it at all. CyTOF abstracts network? Test the limits of the BERT model. Be prepared for all possible outcomes in this regard.
* Get this into people’s hands. Tell a couple people now, and then turn it into a website or something. We’ll see what Martin says. I think at least we have to make a repo out of this so Martin can link it.
* See if I can get this into the hands of Tim Ferriss and Tony Robbins. What if I had a chance to meet them? This would be a big hit for pretty much all the podcasters. I think Lex Fridman and Joe Rogan would get a kick out of this too.

Further research:

* Look at BERT-based similarity of twitter posts. What would a more “central” post mean? Would that be a good post?
* Try dumping in one of those 300-question lists from that one question website that I’ve got in my Evernote. What if centrality is a quick way to figure out whether something is a good question? A good question would be one that leads to lots of other questions. I think that’s what my tool does. It figures out what questions are good questions in a set of questions. I think that’s my selling point. Using AI to find out what questions are good questions. That’s how I sell it anyway. I think eventually I’ll have tens of thousands of questions in here and I’ll be able to find the best questions. The most fundamental questions that any human has ever asked. I predict the top ones will be “Why are we here?" and “What’s the meaning of life?”

I’ve shown:

* Martin Preusse
* Marie
* El-Ad
* Uncle Nigel
* Scott Lohr

I think we can rest on our loins right now and get our first round of feedback. Then we can start building it accordingly, and maybe make some sort of blog post about the thing. Medium maybe. I’ll have to ask some people what makes the most sense.

March 7, 2022

We tidied up the script and added the standard igraph pipeline that I do for R, right in python, so I don’t have to move over to Neo4J to do that. I found a Bible dataset from Kaggle, and made a parallel script that does the analysis of the Bible.

From looking at the Bible analysis, what I observe immediately is that longer verses have higher degree and betweenness centrality scores. These verses also are very hard to read because there are a lot of names and “compound” elements. Thus, what I hypothesize is that my script is optimizing for information density. This is good to know.

From here, I’m going to try a couple other disparate datasets, as each dataset gives me a bit more intuition around what the model is actually doing.

Then I have to figure out how I’m going to get this project out into the wild. Maybe I’ll screen share tomorrow.

March 8, 2022

I switched models to a more recent one, called “all-mpnet-base-v2” because the one I was using was deprecated. The model is stringent, so I had to loosen the 0.3 distance threshold. The results are different, and I won’t be able to comment further until I look at what the Bible results give me.

There are many BERT models that go beyond sentences. There is a BioBERT set of models for things like chemical entities that are worth looking into. So there are a lot of directions to go with this but I’m going to stay with sentences for a while before I move to the next thing. This needs to at least become a blog post.

I think the best thing I can do is make the pipeline air tight and add other kinds of datasets and simply do good write-ups after each dataset in order to learn more about the models.

The new model, if properly adjusted, can give one community structure. In other words, the value is not about finding THE central text. It’s about finding the communities and the central text given communities. This is a nice way to add value.

March 9, 2022

I’m trying the question answer model on the questions to see if that produces more relevant results.

I’m thinking right now that I really explore BERT for all it’s worth. We’re dealing with billion training example models pre-trained and open sourced. Maybe we run it on:

* Companies
* NFTs
* Crypto
* Forex
* Stocks
* Startups
* PubMed abstracts
* Papers
* Captions
* Tweets
* Profiles

Just for the sake of it. Lots of potential use cases. Look into BioBERT.

Basically I can take anything and turn it into a CyTOF dataset.

18:42

I am running a PubMed abstracts dataset to see how well the general model does. I anticipate this to take a long time, but that’s ok. I’m thinking what I might do is separate the model embedding step with the rest of it, simply because that’s the bottleneck. Everything else is pretty general.

Until then I have to think about what else I can do.

I want to take aphorisms like those in the Tao Te Ching. Or something like that. I think that would be a good thing.

What I’ve essentially developed is a recommendation engine based on NLP. The killer app here is something like upgrading the current recommendation engines in various companies. For example, Amazon’s book recommendations could be improved embedding its books. Whole text, or chapters or whatever.

Images could be converted into descriptions and then fed into the language model. Or maybe the other way around if image embedding works.

Remember also that the recommendation engines out there actually are bipartite graphs between people and movies (or whatever). Not just similarity graphs. So we still have to think a bit more about this.

I say that for now, we stop at the questions. The self development questions database is perfectly fine. We can stop there. We can make a Neo4J dash tool and be done with it. We can do that by the end of the week and make a blog post about it. That should be fine.