CS510 Proposal

Dean Alvarez

Logistic Information

Track: Research Track
Team Members: Dean Alvarez — deana3 at illinois dot edu
Project Coordinator: Dean Alvarez

Proposal Details

- Research Question: We first define a Text Augmented Graph (TAG), a graph that contains textual information on both the nodes and the edges. Formally, we can define a TAG as a graph $G = (V, E)$ where $\forall v_i \in V,\ \exists X_i$ where $X_i = (w_0, ...w_n)$ is the text corresponding to the node v_i. Likewise, $\forall e_{ij} \in E,\ \exists Z_{ij}$ Where $Z_{ij} = (w_0, ...w_m)$ is the text corresponding to the edge e_{ij}.
  Now, for this definition of a TAG, we can define the research question: given a dense TAG and some query Q, how can we find both the most relevant sections of text as well as the most relevant edges. In other words, if we define $G' \subset G$ how can we maximize $P(G'|Q, G)$.

- Significance: This is an interesting question because if we could solve it then we could have a new and richer way to explore document collections. Suppose, for instance, that if you have a question about the history of Alan Turing's computer science proofs, the Wikipedia article which has answers and relevant links, also has a lot of non-relevant information and links. Using this approach, the article's less relevant information and links could be pruned so that the user could better hone in on the relevant information and relevant links.

- Novelty: Since the TAG is a novel idea, so far as I can tell, this work is also novel. That said, there are various similar questions related to document retrieval and graph querying which I will likely pull from to answer this particular question. I think the main complication from prior work will be how to join the edge relevance and the section relevance, as there seems to be plenty of work on text information retrieval on its own and for graphs.

- Approach: I think answering this question is going to require some sort of novel method which joins classical retrieval methods with what edges

would be relevant for some information. Note that I don't think it would be enough to simply model edges as documents and then just retrieval the edges with the highest match as you would a document. The reason for this is that with an edge, you generally want a different sort of information. For instance, you might be interested in links that expand on a particular relevant idea. Likewise, you might be interested in links that contrast a particular viewpoint. Either way, these edges seem like they will have to be treated differently from the nodes.

- Evaluation: We can evaluate the solution to this problem in a few ways. The first, of course, could be a qualitative assessment. Another more qualitative way could be to craft a network with predefined "views" which could be evaluated for accuracy. If we knew which the relevant sections and links were, we could use something like precision and recall in order to quantitatively evaluate the methods.

- Timeline: I expect to first finish gathering a dense TAG to run queries on, I suspect that I will use a Wikipedia subset I find to be especially dense. After this, I will need to implement the method for querying. I already have code for storing a tag, so this project will focus mostly on the search algorithms atop of the TAG (although perhaps some data structure improvements will be needed to support querying efficiently). After this I can focus on developing tools for thorough evaluation.

- Task Division: I'm the only group member so I will be doing everything.