PROJECT SCHEDULE

Date	Status	Milestone	Remark
10/6/2022	Complete	Created git repository for project	
	Complete	Created basic chrome extension to highlight all links in a Wikipedia article	
	Complete	Created web scraper for page veiws	
	Complete	Incorporated page views into chrome extension	
	Complete	Created web scraper for page redirects	
	Complete	Incorporated page redirects into chrome extension	
	Infeasible	Get extension latency to less than 1 second	Wikipedia has rate limiting for queries, so this is impossible without violating Wikipedia's terms of service
10/20/2022	Complete	Pivot my project idea: use a wikipedia data dump to preprocess all the information about hyperlink quality by generating a graph from the hyperlinked articles.	
	Complete	Download wikipedia data dump	
	Complete	Generate input and output edge lists	
	Complete	Generate redirect list	
	In Progress	Upload graph information to cloud database	This is probably going to be GCP Firestore because I know how to use it and it's cheap (ish)
11/3/2022	Complete	Create parallel version of script to generate output edge list and redirect list from dump	
	Complete	Create script to combine output edge list and redirect list into a full graph	
	Complete	Create script to upload graph information to Firestore	This was too complex and slow so I pivoted from firestore to mongo db
	Complete	Create script to upload graph information to MongoDB	This was also too complex, so I pivoted from mongo db to redis
	Complete	Create script to upload graph information to Redis	
	In Progress	Make a jupyter notebook with some ideas on information retrieval algorithms to rank wikipedia articles in relation to each other	I will continue to experiment with these techniques and evaluate them
11/17/2022	Complete	Finalize information retrieval algorithm	for pages A and B: L2 distance of [intersection(A.inputs, B.inputs)/union(A.inputs, B.inputs), intersection(A.outputs, B.outputs)/union(A.outputs, B.outputs)] is a pretty reliable metric in my manual testing
	Complete	Create Flask API endpoint to access similarity algorithm for a given page ID	
	Complete	Incorporate flask endpoint into chrome extension	Fairly trivial using javascript fetch API
	In Progress	Organize all the graph generation scripts into main repo and add documentation explaining how to use the scripts step by step	I will have this done by December 2nd deadline
	Complete	Create a user interface for the chrome extension	
	Complete	Create some visualizations for the final presentation	L2 metrics and hockey-stick curve for demonstration