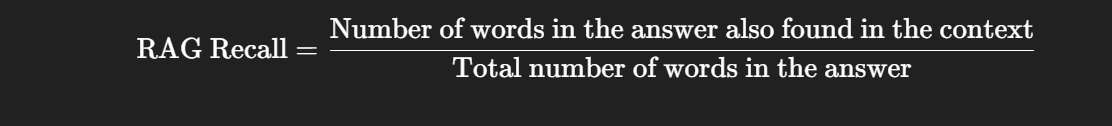
Max Link

I chose the KRAG project for the final. In this project I am using context for RAG and triples for KRAG. Triples are subject, object, and predicate extracted from news articles. I constructed the knowledge graph by extracting these triples with relationship extraction. Relationship extraction is a little different than named entity recognition (NER) because relationship extraction extracts a subject, object, and predicate not just a name. NER just does terms, but relationship extraction lets us understand stories. I used the [global news dataset](https://www.kaggle.com/datasets/everydaycodings/global-news-dataset) for the news articles.

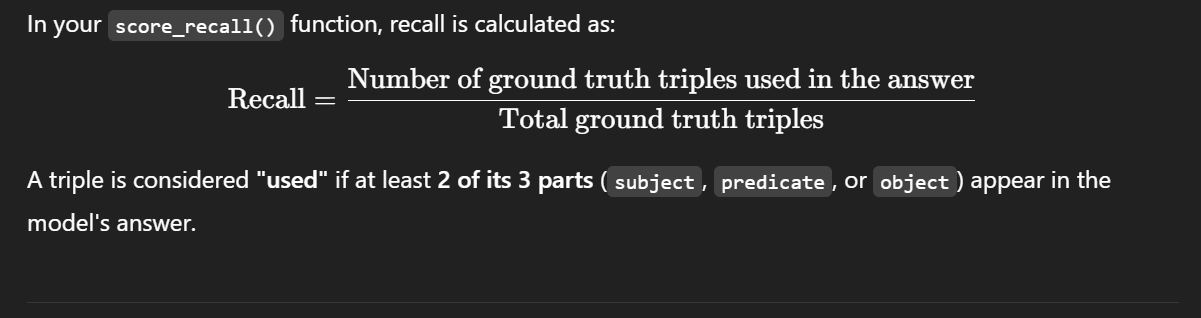
I am calculating recall for RAG and KRAG with these formulas, respectively:

RAG Recall Formula



RAG uses a vectorstore and retrieves tok-K context chunks to answer

KRAG Recall Formula



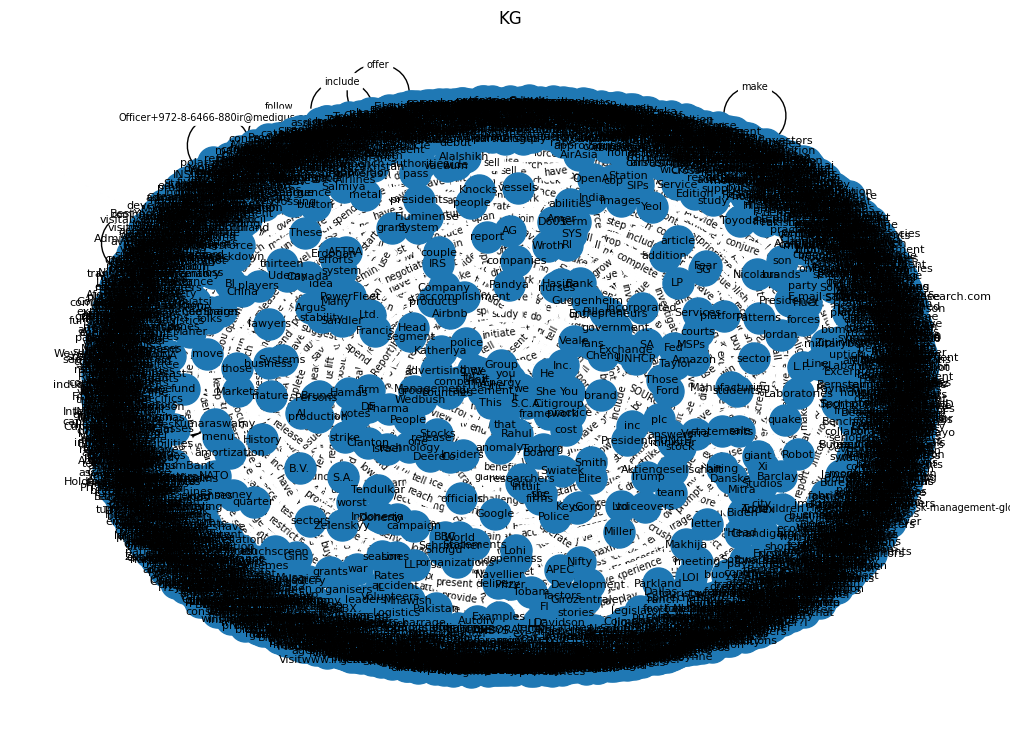
KRAG uses a Knowledge Graph and retrieves top-K triples to answer

The ground truth triples are taken directly from our knowledge graph, and we use Mistral to generate questions about entities in our knowledge graph. This methodology means that we will have a truth to compare our results against for KRAG. Since KRAG does not use context and only uses triples from the knowledge graph (KG), this means we can compare the KG effectiveness with the vectorstore effectiveness.

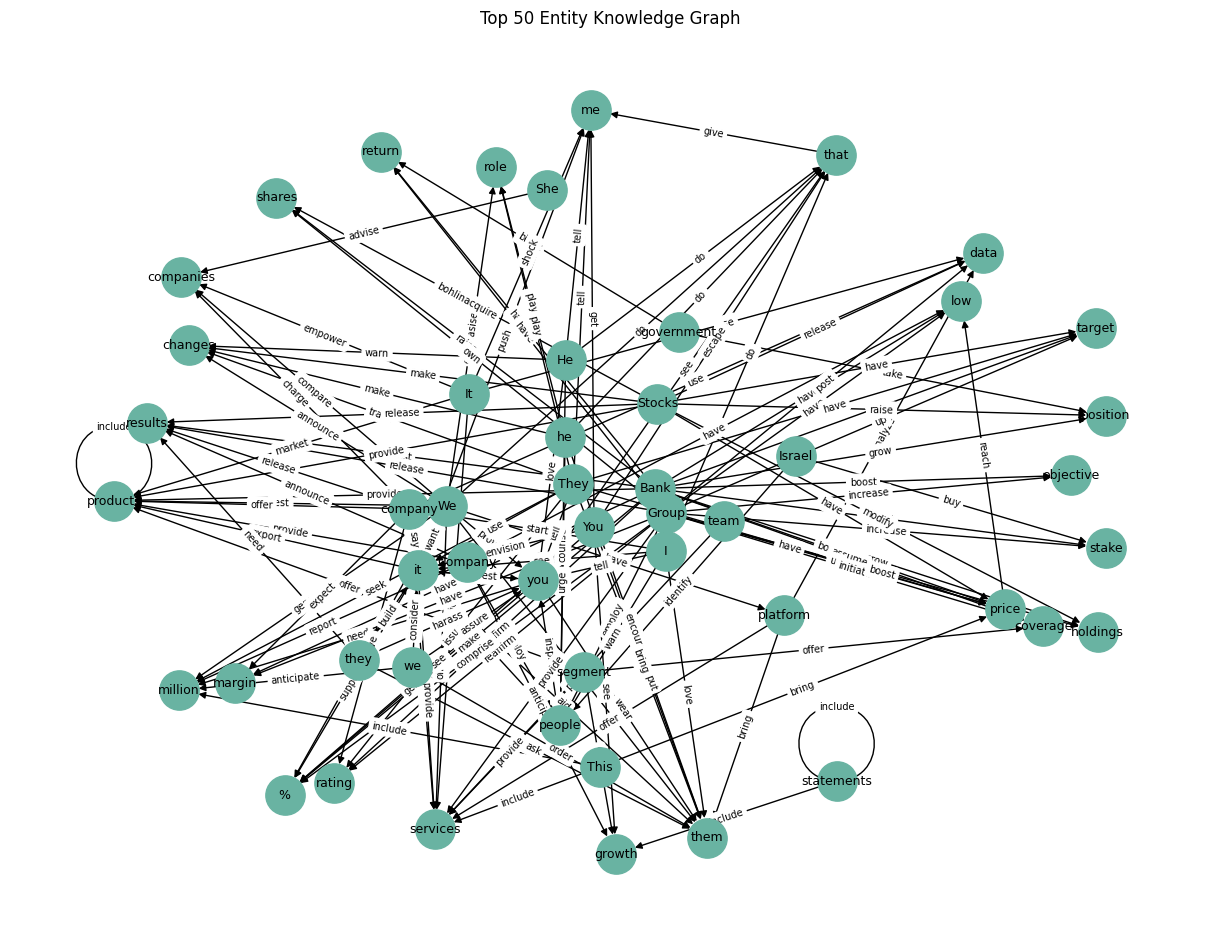
I changed the K value for KRAG and RAG to change how many triples KRAG considered and how much context RAG considered. When the K value is too high, KRAG can get diluted and have a worse recall value. I settled on K = 5 for both KRAG and RAG. When I used 100 articles for the context and triples, KRAG did not noticeably improve over RAG. However, when I used 200 - Max amount of articles, KRAG had a noticeable improvement over RAG.

This suggests that KRAG relying on purely KG without context outperforms RAG that just relies on context.

Here is our knowledge graph:



Here are the top 50 knowledge graph nodes:



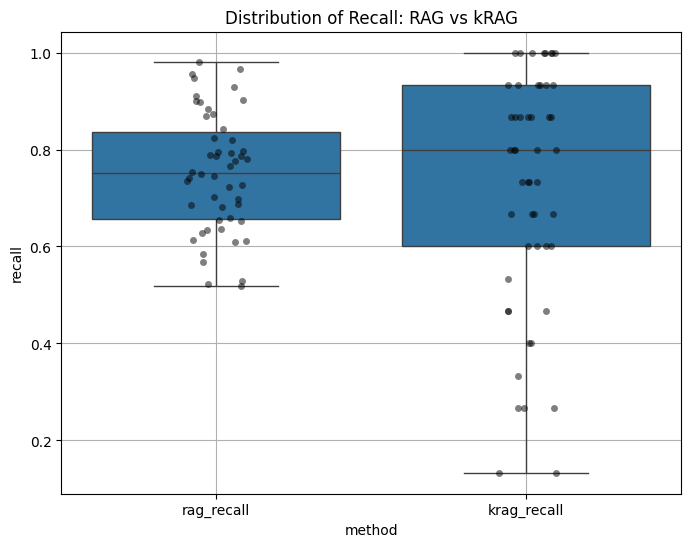
I fed KRAG and RAG 800 articles to use as data for these tests. That was the highest amount I could get to run in a reasonable amount of time.

I got a negative T-stat, which means that the mean of RAG recall is slightly higher than KRAG recall. The p value was way above 0.05 (0.38), which means there was no significant difference between RAG and KRAG in this setup.

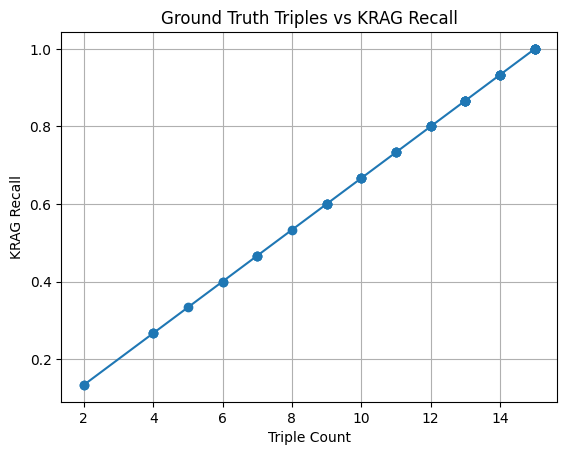
With only 800 articles and basic KG extraction, neither method was superior.

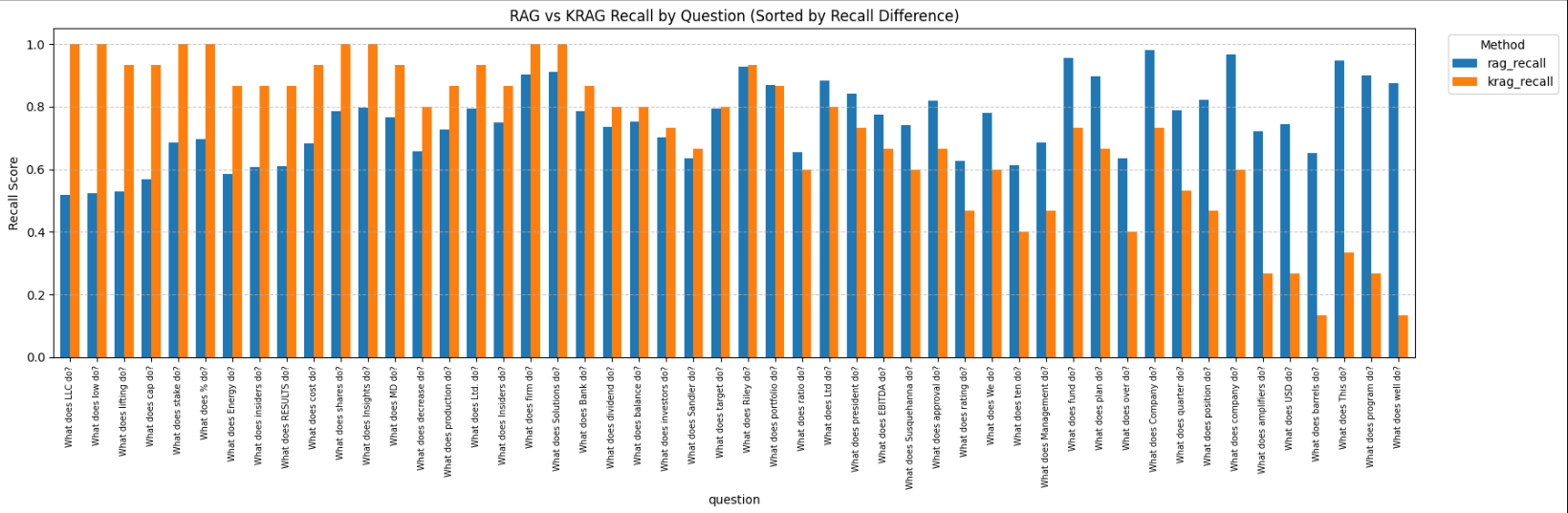
However, KRAG recall increases linearly with how many triples match the question. RAG recall varies, but slightly trends upward with better context match.

RAG recall is slightly tighter and KRAG recall is a bit more spread out.

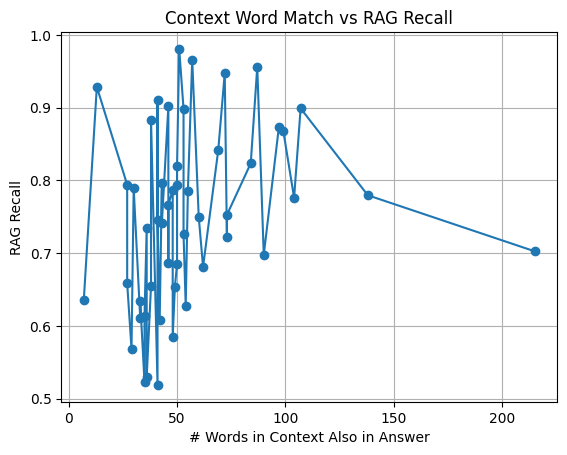


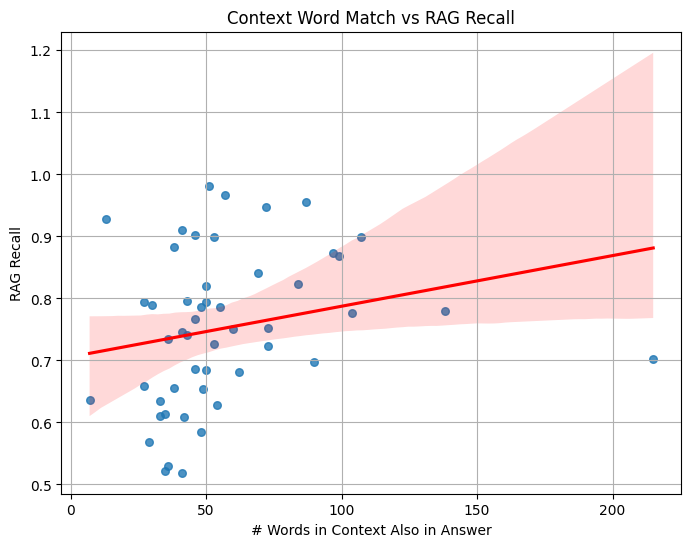
As we can see in the below line graph and bar graph, when the triples do match, KRAG performs very well. KRAG outperforms RAG when the majority of triples found match the words in the answer.





Here is what RAG context match looks like:





Overall, KRAG does better when triples match the answer, but RAG does better when context matches the answer.

I faced some challenges during this experiment. Increasing the dataset size slowed down the evaluation. For KRAG, high K values diluted the precision of triples with noise. The K value should be chosen carefully based on the amount of data and knowledge graph quality to avoid any noise.

The methods used in this experiment can be enhanced in the future by making KRAG a hybrid with context and triples used in combination. The knowledge graph relationship extraction could be improved with NER and relationship extraction used together. Dynamic K selection could be performed based on question complexity. The questions could be fine tuned to better involve relationships between subjects, objects, and predicates.