My grades for **Final Exam - Main Slot**

Q1

1 / 1

When we upload pics/clips/songs... to social media, **what specific mechanism** to we use, to help others find our content when they search?

Why do we need this specific mechanism?

Tags. Because it helps us convert images, sounds and other information into **text keywords** to represent the content we upload, so as to match the search content.

Q2

1 / 1

What algorithms causes/leads to/results in/is implicated in... **'filter bubbles'**?

How does it lead to this?

Content-based Recommender Systems. For example, if a user has watched a lot of action movies, a content-based recommender system might recommend other action movies. This can lead to the user being exposed to a narrow range of content because the system will only recommend items that are similar to what the user has already seen.

Q3

1 / 1

What TWO other items can a search engine serve us (eg. via 'snippets'), in addition to what gets served already? Name each item, and briefly state why it would be of use to us.

---

1. **Direct Answers**. Users can quickly see the answer to the question by directly asking the question, without having to click into the webpage to find it.
2. **Rating**. When users search for places such as restaurants, they can intuitively understand the ratings and save time.

Q4

1 / 1

As you know, genAI (generative AI) is so-called because it can generate content (text, images, video, audio, more).

HOW will this adversely affect search in the (near, even) future? Explain carefully (don't write a vague answer!).

---

1. **Fake or Misleading Content**: GenAI(e.g. Deepfake) can be used to generate convincing and realistic content that is intentionally misleading or fraudulent. For example, genAI could be used to generate fake news articles or social media posts that appear to be legitimate but are actually fabricated.
2. **Security Issue**: GenAI(e.g. Chatbots) are vulnerable to hacking or cyberattacks, if we reveal our privacy to them, even if the military uses these GenAI, personal secrets and state secrets may be leaked.

Q5

1 / 1

ChatGPT (for example) is said to "hallucinate" sometimes (or a lot of times, depending on the type of questions) - an unfortunate term (because only minds can hallucinate!) used by companies who serve this kind of AI products (eg. https://fortune.com/2023/04/17/google-ceo-sundar-pichai-artificial-intelligence-bard-hallucinations-unsolved/). This means that the bot provides an incorrect answer (which we can verify using our own knowledge or experience or by doing a good old search!).

WHAT mechanism (in the algorithm) causes this to happen? Please be specific.

The mechanism is that when we input a sentence to ChatGPT, it will look for the keywords of the query and input this vector into its vocabulary. At this time, the model will **find the term that is most likely to be the next word**, and switch the center of the search range to the predicted next word. And so on to get the final answer for us. The issue with this mechanism is that the generated sentences **do not rely on the factual**

basis but on the vector similarity in the vocabulary. This often results in **no factual relationship between sentences, just a combination of random facts**, leading to the hallucination.

Q6

1 / 1

We typically write code (eg for your HWs #2 to #5) to make use of IR algorithms.

An alternative way is to use 'nodes' (a node is a box-like representation that encapsulates a specific task by executing that task's code) and WIRE them up visually, like so (see for example, https://www.google.com/search?q=rapidminer+dataflow&tbm=isch):

WHAT would be TWO specific (and different from each other) advantages of switching to this way of working (using nodes, as opposed to coding)?

---

1. **Visualization**: Nodes and their connections can be represented visually, making it easier to understand and debug complex systems. This can be particularly helpful in IR tasks, where large amounts of data and complex algorithms can make traditional code difficult to follow and understand.
2. **Modularity**: Nodes can be modular, meaning that they can be reused and combined in different ways to create different systems. This can save time and effort in IR tasks, where many algorithms require similar preprocessing or postprocessing steps.

Q7

0 / 1

Consider the diagram below (the bottom part is simply a slightly zoomed-in portion of the top):

What algorithm did we study, that results in such a collection of polygons?

Why is each polygon convex?

---

1. **Rocchio algorithm**.
2. The Rocchio algorithm produces convex polygons because it uses a **positive weighting factor for the query document** and **negative weighting factors for the non-relevant documents**. This means that the vectors of the relevant documents will be **pulled towards the vector of the query document**, while **the vectors of the non-relevant documents will be pushed away**. This results in a convex polygon, with the query document at the center.

Q8

1 / 1

We have revisited this diagram (at https://bytes.usc.edu/cs572/s23-sear-chhh/home/index.html) on and off, many times:

Now that the course is over (after you get through this exam, lol), how does it summarize (encapsulate) the course? **Pick four specific and different IR tasks** we studied during the course (including during the 'Assorted topics' pair of lectures), and explain (in a line or two) each, in terms of the three pieces of our diagram.

---

1. **Information Retrieval**: The user submits a query to the search agent, which retrieves relevant documents or information from a vocabulary of content.
2. **Document Classification**: The user provides a document to the classification agent, which assigns one or more labels or categories to the document based on its content.
3. **Recommender Systems**: The user interacts with the recommender agent, which generates personalized recommendations for content based on the user's past behavior or preferences.
4. **Question Answering**: The user asks a question to the question-answering

agent, which retrieves relevant information from a vocabulary of content and provides an answer to the user.

Q9

4 / 4

1+1=2 points: **How** do recommendation engines (REs) work?

1+1=2 points: **What** are two different uses for them when we search?

1.1. **Collaborative Filtering**: This algorithm identifies users who have similar preferences, based on their past behavior, and then suggests items that those users have liked or interacted with. User-based filtering suggests items based on the preferences of users who are similar to the current user, while item-based filtering suggests items that are similar to those that the user has already shown an interest in.
1.2. **Content-Based Recommendation**: This algorithm suggests items that are similar to the items that the user has already shown an interest in.
2.1. **Discovery**: REs can be used to suggest new items that the user might be interested in, based on their past behavior or preferences. For example, if a user has been browsing for shoes, a RE could suggest other shoes that are similar in style or brand.

2.2. **Personalization**: REs can be used to personalize search results, by showing items that are more relevant to the user's interests or preferences. For example, if a user is searching for movies, a RE could rank the results based on the user's past movie ratings or viewing history, showing the most relevant movies at the top of the list.

Q10

1 / 1

What is 'vector similarity search'? Rather than Googling on ChatGPT-ing, answer, based on what we covered.

For **what two different** IR tasks are vector DBs useful? Name, and explain briefly why we couldn't do them without vector DBs.

---

1. Map the traditional database (e.g. table) to vector space (each column becomes a dimension). In this way, each row, that is, each instance, will be represented as a vector. Then it can be easily calculated with cosine similarity on the vector space similarity between different instances.

2.1. **Content-Based Image Retrieval**: In this task, a user submits an image as a query, and the system retrieves all images from a database that are most similar to the query image. Without vector DB, it is difficult for us to accurately describe an image, but vectors can represent them as vectors of visual features.
2.2. **Semantic Textual Similarity**: In this task, the goal is to determine the degree of similarity between two documents. Without vector DB, it is difficult for us to perform high-

dimensional data (every word is a dimension), but vector DBs can represent the documents as vectors of word embeddings, and then indexed for efficient retrieval based on similarity.

Q11

1 / 1

Summarize ANY TWO of your HWs #2 through #5 - **WHAT was the algorithm** underlying, **WHAT task** did it help accomplish?

---

1. HW3. I used the **inverted index** in this assignment. It helps me quickly search documents where a certain term appears.
2. HW5. I used the **vector DB** and **t2v-transformers** in this assignment. They help me convert documents into vectors and search for documents with similar topics to the query, even if the term of the query does not appear in these documents.

Q12

1 / 1

What does 'OPL' stand for, in OPL stack?

What is its main use? Again, stick to what we covered, rather than searching online!

---

1. OPL stands for **OpenAI** + **Pinecone** + **Langchain**.
2. It is like LLM + Langchain + Vector DB. We can train a gpt-like generative AI that meets our own needs based on this pipeline. For example, we can connect LLM to a SQL database so that LLM automatically converts the text language into SQL commands and fetches the corresponding results from our database.

Q13

1 / 1

How do 'RDF triples' make search better? Explain in a few lines.

Using RDF triples, search engines can better understand the relationships between entities and concepts in the data. For example, a user searches for "restaurants in Paris." A search engine that understands RDF triples can use this query to find all entities related to "restaurants" and "Paris" in its RDF triple database.

Q14

4 / 4

Name four algorithms we looked at, for IR tasks, that rely on iteration or recursion. For each, **explain briefly, how** the iteration or recursion helps (ie. what changes during each run).

---

1. **PageRank**: The algorithm is iterative because it constantly updates each **page's rank** based on the ranks of the pages that link to it.
2. **k-means**: The algorithm starts by randomly assigning each document to a cluster, and then iteratively reassigns documents to the cluster whose center (or mean) is closest, hence the name k-means. The **centroid of the clusters** is recalculated on each iteration.
3. **Rocchio**: When a new document is input, Rocchio first calculates its distance to the centroid of each cluster, so as to attribute it to the nearest cluster, and then the **centroid of the clusters** is recalculated on each iteration.
4. **HITS**: The algorithm updates the **authority** and **hub** scores for each page in each iteration based on the scores from the previous iteration.

Q15

4 / 4

Name, and very briefly discuss 4 ML-based algorithms we looked (towards the end of the course!), for IR tasks.

---

1. **Vision Transformers**: It divides the input image into several fixed-sized small blocks (called tokens). Then, each image block is converted into a one-dimensional vector, and these vectors are used as the input sequence of the transformer model. In IR, it can be used to learn local and global information in images and to recognize images (like a CNN).
2. **Faiss algorithm(an ANN)**: First divide all the vectors in the database into multiple clusters through the K-means clustering algorithm, each cluster has its own corresponding index. In the database search, first, find the target cluster corresponding to the query (comparing the distance between the query and each centroid), and then do an exhaustive comparison in this cluster **to reduce the calculation time**.
3. **t-SNE**: It converts high-dimensional space distance into probability to embed high-dimensional data for visualization in a low-dimensional space of two or three dimensions. In IR, t-SNE could be used to visualize document collections based on their vector space representations.

4. **NeRF**: It can restore 3D objects through 2D color and object feature information. In IR, NeRF could be used in AR games or generate immersive views (e.g. Google Map).

Q16

4 / 4

Consider the following summary of ML/DS algorithms (https://bytes.usc.edu/cs572/s23-sear-chhh/extras/docs/KNIME_ML_cheatsheet.pdf):

Of the various algorithms listed above, **pick FOUR** that are useful in IR [we studied them], and explain briefly how each works: **what** does the algorithm do, **what IR task** does it help with.

---

1. **k-Nearest Neighbors (kNN)**: It can find the 'k' documents closest to the input query and realize the classification of the query. In IR, it can be used for document classification or recommendation systems.
2. **k-Means**: It is an unsupervised learning algorithm that partitions n data points into 'k' clusters. In IR, k-Means can be used for document clustering, where each cluster might represent a topic.
3. **Support Vector Machine (SVM)**: It tries to find a hyperplane in an N-dimensional space (N being the number of features) that distinctly classifies the data points. In IR, SVMs can be used for tasks such as document classification (e.g., spam vs. not-spam emails).
4. **Decision Tree**: It's a flowchart-like structure where each internal node

represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. In IR, decision trees can be used for tasks like text categorization. For example, given a set of documents, a decision tree might be trained to classify them into different categories based on the occurrence of certain keywords.

Q17

1 / 1

TikTok's recommendation engine uses a specific data structure, to optimize how it works. **What** is the name of the data structure? In your own words, **how** does it work (you can explain a high level, no need for specifics)?

---

1. **Monolith**.
2. When user upload a video, it is first processed and compressed on the user's device, then processed by the Monolith system, which consists of a series of modular components that handle different tasks such as transcoding, filtering, and encoding. Next, the video is stored in TikTok's content delivery network (CDN), which is designed to efficiently distribute videos to users all around the world. Finally, When a user requests a video, TikTok's servers use a variety of techniques to determine the most relevant videos to display, including personalized recommendations based on the user's past behavior. The selected videos are then served to the user from the CDN.

Q18

1 / 1

Given two vectors (eg like shown below), **what two** 'similarity measures' can we calculate ?

What do vectors have to do, with IR in the first place?!

---

1. **Cosine Similarity**, **Euclidean Distance**
2. In IR, the document and query are first converted into vectors by extracting keywords.