DAIS Qualifying Examination

Spring 2017 (March 9, 2017)

Department of Computer Science
University of Illinois at Urbana-Champaign

Time Limit: 180 minutes

Please read the following instructions carefully before starting the test:

- The examination has 10 questions, including one basic concept question and 9 topic questions. You are required to answer the basic concept question and any **5** of the 9 topic questions of your choice. If you answer more than five topic questions, the committee will randomly select five to grade.
- The basic concept question has **9** subquestions. You are required to answer all of these subquestions.
- The 9 topic questions are distributed by areas as follows:
  - **Database Systems**: 3 questions
  - **Data Mining**: 3 questions
  - **Information Retrieval**: 3 questions
- Use a **separate booklet** for each question that you answer. That is, you will use a total of six booklets (1 for the basic concept question and the rest for the 5 chosen topic questions).
- To ensure the fairness of grading, all the exams will be kept anonymous. Your answer sheets should thus only bear the assigned ID but *no names.*
- Please make your answers clear and succinct; you will lose credit for verbose, convoluted, or confusing answers. *Succinctness does count!*
- The questions have been designed to avoid any ambiguity. However, in case you are not sure how to interpret a question, please assume one interpretation, write down your assumption, and solve the problem accordingly.
- On average, you will have 30 minutes for each of the six questions that you have chosen to answer. So plan your time accordingly, and keep in mind that some questions/subquestions may demand more writing than others. You may find it useful to periodically assess your time usage, and adjust your time allocation dynamically as needed to avoid running out of time on questions that you could have answered.

Required Question (Basic Concepts): Problem 0

You are required to all of the following subquestions. Each subquestion is worth 1 point. Please focus on the major point and keep your answer concise; in general, an answer is expected to be no more than two sentences.

(1) (*Query Language*) How is a query language different from a programming language?

(2) (*Indexing*) To support indexing of *arbitrary* data types beyond the basic ones (e.g., integers, strings)– such as indexing *sets* of values– identify two key issues.

(2) (*Query Processing*)
Join is expensive to process. Name two techniques for speeding up join processing and briefly explain how each of them helps.

Note: Please start with a new answer sheet.

Short answers: The answer of each of the following questions is expected to be no more than two lines (i.e., no explanation nor reasoning is needed)

(4) (*Data mining algorithms*)

1. *name two algorithms that classify high-dimensional data effectively*, and
2. *name two algorithms that mine sequential patterns effectively.*

(5) (*Data mining concepts*) Use one or two sentences to distinguish the following pairs of concepts or methods:

1. *AdaBoost* vs. *random forest*, and
2. *weakly supervised classification* vs. *distantly supervised classification.*

(6) (*Selection of data mining methods*) Name or outline one data mining method that best fits each of the following tasks:

1. *effectively and incrementally cluster data streams*, and
2. *find oil spilled regions in the ocean.*

Note: Please start with a new answer sheet.

(7) (*TF-IDF*)

In the vector space retrieval model, we often use TF-IDF weighting to compute a vector for each document. Consider a paper about information retrieval models in the collection of all the literature articles on computer science topics in the ACM Digital Library. What kind of words would have the highest TF-IDF weights in the vector representing such a paper? Given an example of a word that has a high TF-IDF weight, and another example of a word with low TF-IDF weight. (Use no more than 3 sentences to answer the question.)

(8) (*Inverted index*)

What impact would it have on the inverted index if we remove all the stop words (i.e., common words such as “the” and “a” in English)? (Use no more than 3 sentences to answer the question.)

(9) (*Evaluation*) Let $R = (+, +, +, +, -, -, -, -)$ be the relevance status of 8 documents in a ranked list of retrieval results with “+” indicating a relevant document and “-” a non-relevant document. (Thus, the top four documents are relevant, while the bottom four are non-relevant). Suppose there are in total 10 relevant documents in the whole collection. Compute the precision, recall, and average precision. It’s sufficient to give an expression; there is no need to reduce the expression to a value.

Database Systems: Problem 1

The paper “A Relational Model of Data for Large Shared Data Banks” proposes the relational model for data management.

Part 1

What is a *data model*? Why is it important for a database system? (*1 point*)

Part 2

In the relation model, we often view a relation as a *table*, with attributes being *columns* and tuples being *rows*. With this view, there apparently exists some “duality” between columns and rows: For instance, in terms of relational algebra, one can think of *projection* as reduction on columns while *selection* on rows. Similarly, one can think of *join* as merging the columns of two tables, while *union* the rows. In these senses, rows and columns seem to be somehow “symmetric” in the relational model.

However, in the relational model, rows and columns also possess different properties, which clearly distinguish them (and make them asymmetric). This problem asks you to observe such characteristics: Give *two* properties that contrast rows and columns in the relational model. (*3 points*)

Part 3

In this paper, the proposal of the relational model is motivated by the need for *data independence*, which Codd defines as the capabilities that *data representation characteristics can be changed without logically impairing application programs.* In particular, the paper identifies three dependencies: *Ordering Dependence. Indexing Dependence*, and *Access Path Dependence.*

For each of these dependencies, explain what it is, and how the relational model actually eliminates it. (*3 point*)

According to Codd’s definition, do you think the relational model achieves *full* data independence? Explain why or why not. Use examples to explain, and be concrete and specific in your arguments. (*3 points*)

Database Systems: Problem 2

The "R-Trees" paper presents an index structure for spatial searching.

Part 1

R-tree aims at indexing multi-dimensional data. The author claims that *"structures using one-dimensional ordering of key values, such as B-trees and ISAM indexes, do not work"* for such data.

Let's examine this claim.

a) We ask you to attempt to use B-trees to do this multi-dimensional indexing. Describe how we can use B-trees to "simulate" such indexing, to support queries that find "nearest neighbors" of a given point. (*2 points*)

b) Why is this not an ideal solution? Identify the limitations in your design. (*2 points*)

Part 2

R-tree is, at least conceptually, a generalization of B-tree.

a) Can you use R-tree to do what B-tree can do, say, to index integers? Explain why or why not. (*1 points*)

b) Identify the similarities and differences between R-tree and B-tree. (*2 points*)

c) Identify how R-tree actually *generalizes* B-tree? (*3 points*)

Database Systems: Problem 3

The "Pig Latin" paper introduces a procedural query language.

Part 1

What distinguishes a procedural query language like Pig Latin from a declarative one like SQL? (*1 points*)

Part 2

Query optimization has been applied to SQL query processing.

The paper claims that similar query optimization can be applied to Pig Latin queries. We want to examine this argument and compare optimization for Pig Latin vs. SQL.

a) Can you identify the aspects of Pig Latin, which may make query optimization significantly *easier*? (*2 points*)

b) To contrast, can you also identify the aspects of Pig Latin, which may make query optimization significantly *harder*? (*3 points*)

Part 3

For each of the hard issues you identified in the above (2b), can you speculate/suggest a potential solution? (*4 points*)

Data Mining: Problem 1

Social media (e.g., tweets) may generate a lot of geo-coded social communications that provide rich information for data mining

Part 1

(a) With massive geo-coded tweets, one may be able to mine unusual events happening in a local region (e.g., a city center). Outline an efficient method that mines such unusual events from geo-coded tweets effectively. (*2.5 points*)

(b) In reality, only a small portion of tweets are geo-coded. Outline a method that may utilize both geo-coded and non-geo-coded tweets to effective discovery of local unusual events. Explain why your method may lead to better performance than using only geo-coded tweets. (*2.5 points*)

Part 2

(a) Outline an efficient method that may use geo-coded tweets to discover periodic events in a local region. (*2.5 points*)

(b) In many cases, tweets are short and contain lots of abbreviations or acronyms. On the other hand, news and other formal media may contain sufficient information for newcomers to understand. Discuss how news and tweets may complement each other and outline a mining method that may realize an effective joint mining. (*2.5 points*)

Data Mining: Problem 2

A biological network can be considered as a *heterogeneous information network*. It may contain information about genes connected with proteins, diseases, drugs, and biological pathways. Suppose one would like to construct such a network from a set of research papers from a biological literature database, such as PubMed Central, which contains a good set of research papers.

Part 1

(a) Discuss how such a heterogeneous biological information network can be constructed from research literature. (*2.5 points*)

(b) Discuss how to define *similar genes* effectively by exploring the concepts of *meta-path* and *embedding*, and outline an efficient mechanism to implement it in large heterogeneous networks. (*2.5 points*)

Part 2

(a) If one wants to cluster genes and proteins, based on their relationships with diseases, outline an efficient method to implement it. (*2.5 points*)

(b) Discuss how to design an effective method to predict new functions of a drug based on the information provided in such a heterogeneous information network. (*2.5 points*)

Data Mining: Problem 3

Consider a large, connected, attributed graph $G = (V, E)$, where each node $v \in V$ takes on a discrete value $a_v \in A$. The attribute A takes on k distinct attributes values. Each attribute value k appears in the graph with probability p_k where $\sum_k p_k = 1$. Assume that the structure of the network and the underlying attribute distribution are both unknown. Our initial goal is to estimate the attribute distribution (i.e. p_k). The network operator disallows random access to the network, but will allow a random walk starting from a single seed node. The network operator provides us with a seed node, and we assume that network operator chooses the seed node uniformly at random from the network. We start the random walk from this seed node s_0.

During the random walk, at each step, after inspecting the attribute value of the current node, we update a histogram storing the distribution of attribute values by incrementing count of the bin of the attribute value corresponding to the current node. We pick the next node to visit by picking a neighbor uniformly at random from the neighbors of the current node. We end the walk after $N = O(|V|)$ steps.

(a) Provide an estimate of the distribution of attribute values by normalizing the histogram counts. (*3 points*)

(b) Identify any causes of bias in your estimate in part (a). (*3 points*)

(c) Assume you can't change your random walk (i.e. the next node that you visit is always picked uniformly at random from the neighborhood of the current node). Suggest a way to develop an unbiased estimate of the attribute distribution. (*4 points*)

You may make any assumption needed to address the question as long as you provide a justification.

Information Retrieval: Problem 1. Evaluation

[Carterette 15] refers to the following paper:

Ben Carterette, Ashraf Bah, and Mustafa Zengin. 2015. Dynamic Test Collections for Retrieval Evaluation. In Proceedings of the 2015 International Conference on The Theory of Information Retrieval (ICTIR 15).

Part 1

a In [Carterette 15], the authors mentioned there are three widely-used models for retrieval evaluation. What are these three models? Briefly explain how each works (use no more than two sentences for each model). (*2 points*)

b What is the main motivation for the authors of [Carterette 15] to propose a new way to evaluate a retrieval system? (Use no more than 3 sentences to answer this question.) (*1 point*)

Part 2

a In [Carterette 15], the authors proposed a dynamic test collection framework that can simulate the behavior of users of a search engine. What specific user behaviors are simulated by this framework? (*2 points*)

b What kind of retrieval systems can be evaluated using such a dynamic test collection method, but cannot be evaluated easily using any existing evaluation model? (Use no more than 3 sentences to answer this question) (*1 point*)

c Briefly explain the steps to be followed in order to evaluate a retrieval system using the proposed dynamic test collection framework. (Use no more than 3 sentences.) (*1 point*)

Part 3

One of the challenges in implementing the dynamic test collection framework is how to build a "click simulator." In a real search engine application system such as Google or Bing, we would be able to log many users' search activities over time. Such log data can potentially be used to build a more accurate click simulator than the simulation techniques proposed by the authors of [Carterrette 15]. Suggest some specific ideas for building a click simulator by leveraging the search log data. Try to use mathematical formulas to describe your ideas if you can. (*3 points*)

Information Retrieval: Problem 2. Cross-Lingual Information Retrieval

[Vulic et al. 13] refers to the following paper:

Ivan Vulic, Wim Smet, and Marie-Francine Moens. 2013. Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora. Inf. Retr. 16, 3 (June 2013), 331-368.

Part 1

a What is cross-language information retrieval? What is a comparable corpus? (*1 point*)

b What is the "distributional hypothesis" proposed by Harris in 1954? How did the work [Vulic et al. 13] apply the distributional hypothesis to perform cross-language lexicon extraction? (Use no more than 5 sentences to answer this whole question about distributional hypothesis.) (*2 points*)

Part 2

a The work [Vulic et al. 13] used an extension of the standard LDA model, which is called bilingual LDA (BiLDA). What are the key differences between BiLDA and the standard LDA? (*2 points*)

b In Cross-Language Information Retrieval, a key challenge is to compute $p(q_i|D_J)$, where q_i is a query term in the source language and D_J is a document in the target language. Explain how the output from BiLDA can be used to compute $p(q_i|D_J)$ by providing a rough formula. (*2 points*)

Part 3

Consider performing cross-language information retrieval by using a bilingual dictionary (e.g., a Chinese-to-English dictionary). Suppose our source language is Chinese and target language is English. It is often the case that a non-ambiguous word in a source language may be translated into an ambiguous word in the target language. For example, the word in Chinese referring to the money sense of "bank" (i.e., financial institution) is not ambiguous, but if we translate the Chinese word into the English word "bank," it would introduce ambiguity because in English, "bank" is ambiguous and can also mean "edge of a river." Can you suggest one or multiple methods for solving this problem? (Hint: pseudo-relevance feedback may help.)

(*3 points*)

Information Retrieval: Problem 3. Word Embedding

[Kenter and de Rijke 15] refers to the following paper:

Tom Kenter and Maarten de Rijke. 2015. Short Text Similarity with Word Embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM 15).

Part 1

a The work [Kenter and de Rijke 15] mentioned that multiple ways to compute short text similarity were proposed in the existing work. Can you name at least three of these methods? (*1 point*)

b The work [Kenter and de Rijke 15] studied how to use word embedding to improve computation of short text similarity. From computation perspective, what is the input and what is the output of an word embedding algorithm? (*1 point*)

c Given two short text strings, the work [Kenter and de Rijke 15] used supervised learning to compute the similarity of the two text strings, which requires training data. What does the training data look like? (*1 point*)

Part 2

a In [Kenter and de Rijke 15], the authors compute the semantic text similarity feature $f_{sts}(s_l, s_s)$ by using a formula similar to the BM25 retrieval function which involves a sum over all terms in one short text. If we interpret the formula as a retrieval function, we may view this similarity function as computing the similarity between a query and a document. With this retrieval view, how did the authors determine which of the two short texts is the "query" and which is the "document"? How did the authors ensure that the similarity function is symmetric? (*2 points*)

b When using word embedding to compute similarity of short text, the authors of [Kenter and de Rijke 15] mentioned the problem of out-of-vocabulary (OOV)? Explain why OOV would be a problem for computing similarity of short text. How did the authors solve the problem? (*2 points*)

Part 3

The method proposed in [Kenter and de Rijke 15] is meant for computing similarity of short text. How do you think about applying the method to compute similarity between two very long text documents (e.g., two news articles)? Do you expect it to be more effective than a regular method such as using TF-IDF weighting to represent two documents as vectors and then computing the cosine similarity of the two vectors? Why? (*3 points*)