NLP and Information Retrieval
Sample Exam 1

Generated by ChatGPT

Instructions

- Answer all questions.
- Show all calculations where required.
- Justify your answers clearly in transfer questions.

Task 1: Tokenization & Subword Methods

1a) (Knowledge)
Explain the difference between **word-level tokenization** and **subword tokenization** (e.g., Byte-Pair Encoding). Why is subword tokenization widely used in modern NLP?

1b) (Understanding)
Given the word `"unbelievable"`, provide a possible BPE tokenization output (you can invent merges). Briefly explain why this split might help handle out-of-vocabulary words.

Task 2: Text Classification Foundations

2a) (Knowledge)
Name three possible **text classification** tasks and briefly describe the feature extraction steps you might perform in a **traditional** (non-neural) pipeline (e.g., TF-IDF vectors).

2b) (Transfer)
A social media company wants to **detect hate speech** vs. **harmless content**. Given a small labeled dataset of tweets, would you choose a simple Naïve Bayes classifier or a large Transformer-based classifier (like BERT)? Justify your choice by weighing **time/resources** vs. **potential accuracy**.

Task 3: IR Basics & BM25

3a) (Knowledge)
Briefly define the **BM25** scoring function. What are the roles of term frequency (TF), inverse document frequency (IDF), and document length in BM25?

3b) (Do Stuff / Short Calculation)
Suppose we have 3 documents, each containing the word "computer" as follows:

- Document A: 3 occurrences
- Document B: 0 occurrences
- Document C: 5 occurrences

Given a query "computer," explain how BM25 assigns higher scores to documents with more term occurrences. You do **not** need to calculate an exact numeric score—just outline how increased term frequency and shorter document length would boost BM25.

Task 4: True or False (Conceptual)

Mark each statement **True (T) or False (F)** and provide **one sentence of explanation**:

1. Transformers rely on recurrent connections to handle long-range dependencies.
2. Dense retrieval uses approximate nearest neighbor search to scale to large document collections.
3. Instruction tuning only works for small, domain-specific language models.
4. N-gram language models struggle with zero probabilities for unseen word sequences.
5. In a two-stage IR pipeline, the first stage must always be BM25.

Task 5: Neural Re-Ranking & BERT Fine-Tuning

5a) (Understanding)
Explain how a **neural re-ranking** approach (e.g., BERT re-ranker) works in combination with a **first-stage** retriever. Why might this two-stage approach outperform using only BM25 or only a single dense retrieval model?

5b) (Transfer)
You have a **legal document retrieval** system where correctness is critical. Describe a scenario where **re-ranking** with a carefully fine-tuned BERT model is crucial, and highlight potential **drawbacks** (e.g., inference speed) of adding this re-ranker.

Task 6: Dense Retrieval (Calculation / "Do Stuff")

6a) (Knowledge)
What are the **three major phases** in building a dense retrieval system? Name **one** advantage dense retrieval has over BM25.

6b) (Short Calculation)
You have a query vector $\mathbf{q} = [1.0, 0.5]$ and two document vectors:

- $\mathbf{d}_1 = [0.9, 0.4]$
- $\mathbf{d}_2 = [0.1, 0.2]$

Using **dot product** as the similarity function, compute:

- $\mathbf{q} \cdot \mathbf{d}_1$
- $\mathbf{q} \cdot \mathbf{d}_2$

Which document is **more relevant** under a dot-product dense retrieval scheme?

Task 7: Ranking Evaluation Metrics

7a) (Knowledge)
Define **Precision@k** and **Recall** in the context of Information Retrieval. Provide a quick example of each.

7b) (Calculation)
A search system returns 5 results for a query. Among these results, the relevant ones appear at ranks [1, 4]. There are 3 relevant documents in total for this query.

- Calculate **Precision@5**.
- Calculate **Recall**.
- Which metric is higher in this scenario, and why might that matter?

Task 8: Adapters & LoRA (Parameter-Efficient Fine-Tuning)

8a) (Understanding)
Explain the core idea behind **adapter layers** and **LoRA**. How do they reduce the number of trainable parameters compared to fully fine-tuning a large model?

8b) (Transfer)
You need to fine-tune a **GPT-like** model for a specialized task but have limited GPU memory. Would you choose **adapter layers** or **LoRA**, and why?

Task 9: Instruction Tuning & Prompting

9a) (Knowledge)
What is **instruction tuning**, and how does it differ from using **few-shot prompting**?

9b) (Transfer)
You need an LLM to follow instructions better. Outline a **dataset** you would create for instruction tuning and explain its benefits.

Task 10: RLHF & Model Alignment

10a) (Understanding)
How does **Reinforcement Learning from Human Feedback (RLHF)** help align an LLM?

Task 11: Long Context Handling

11a) (Knowledge)
Explain one approach to handling **long input sequences** in Transformers.