Assignment 03 - Internal Data Storage, Underlying Models (I)

Performance summary

✓ Assessed

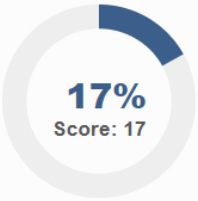
Success status	Score	Attempts
Undefined	17 of 100 points	1

Results

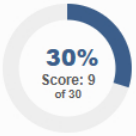
Course	**Web Retrieval SoSe 2025** ID: 4853531344 / 109696966067240
Test	**Assignment 03** ID: 4562682013

This are your test results

Duration	1h 48m 20s 5/10/2025, 3:01 PM - 5/15/2025, 1:02 PM
Answered	9 of 9 questions (100%)
Your score	17 of 100 points (17%)

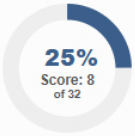


1. Knowledge Tasks (30 points) 4



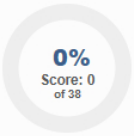
go to section ›

Practice Tasks (32 points) 3



go to section ›

Programming task (38 points) 2



go to section ›

1. Knowledge Tasks (30 points) 9 of 30 points (30%)

True/False (10 points)

Status	Answered	
Your score	9 / 10	90%

Response

Which of the following statements is correct? (unanswered/right/wrong)

Please note:

- Maximum Overall Score -> 10 points
- Minimum Overall Score -> 0 points
- Incorrect Answer -> -0.5 points
- Unanswered -> 0 points

Unanswered	Right	Wrong	
☐	☑	☐	Wildcard queries can use the asterisk (*) symbol to represent one or more missing characters.
☐	☐	☑	Document tokenization involves splitting text into paragraphs.
☐	☐	☑	Accuracy is typically the most accurate measure of IR system effectiveness
☐	☑	☐	The process of normalization in text processing includes converting all characters to lowercase.

☐	☑	☐	Precision at k (P@k) is influenced by the choice of the cutoff value k.
☐	☑	☐	Accuracy is not considered a good metric because high accuracy can be achieved by a skill-free model that only predicts the majority class in imbalanced datasets.
☐	☑	☐	F-Measure is a weighted average of precision and recall.
☐	☐	☑	Every term, included stopwords, in a document is indexed in a term-document matrix.
☐	☑	☐	Automatic query formulation methods can outperform original, manually formulated queries in terms of search effectiveness.
☐	☐	☑	The Boolean retrieval model uses ranking.
☐	☑	☐	Stemming is a process that reduces words to their root form.
☐	☑	☐	Accuracy is an evaluation metric that gives the fraction of predictions that the model is correct
☐	☐	☑	Manual refinement of automatically formulated queries cannot improve their effectiveness.
☐	☑	☐	Phrase queries can be handled using a biword index.
☐	☐	☑	Positional indexes cannot be used to support proximity queries.
☐	☑	☐	The objective method for query formulation uses a statistical approach to identify relevant terms.
☐	☑	☐	A term-document matrix is sparse because most terms do not appear in most documents.
☐	☑	☐	In Boolean queries, the OR operator is used to select documents that contain at least one of the specified terms.
☐	☑	☐	The size of a T-D-Matrix depends basically on the size of corpus
☐	☑	☐	Precision is a measure that calculates the fraction of retrieved documents that are relevant.

▼ Solution

Which of the following statements is correct? (unanswered/right/wrong)

Please note:

- Maximum Overall Score -> 10 points
- Minimum Overall Score -> 0 points
- Incorrect Answer -> -0.5 points
- Unanswered -> 0 points

Unanswered	Right	Wrong	
☐	☑	☐	Wildcard queries can use the asterisk (*) symbol to represent one or more missing characters.
☐	☐	☑	Document tokenization involves splitting text into paragraphs.
☐	☐	☑	Accuracy is typically the most accurate measure of IR system effectiveness
☐	☑	☐	The process of normalization in text processing includes converting all characters to lowercase.
☐	☑	☐	Precision at k (P@k) is influenced by the choice of the cutoff value k.
☐	☑	☐	Accuracy is not considered a good metric because high accuracy can be achieved by a skill-free model that only predicts the majority class in imbalanced datasets.
☐	☑	☐	F-Measure is a weighted average of precision and recall.
☐	☐	☑	Every term, included stopwords, in a document is indexed in a term-document matrix.
☐	☐	☑	Automatic query formulation methods can outperform original, manually formulated queries in terms of search effectiveness.
☐	☐	☑	The Boolean retrieval model uses ranking.
☐	☑	☐	Stemming is a process that reduces words to their root form.
☐	☑	☐	Accuracy is an evaluation metric that gives the fraction of predictions that the model is correct
☐	☐	☑	Manual refinement of automatically formulated queries cannot improve their effectiveness.

☐	☑	☐	Phrase queries can be handled using a biword index.
☐	☐	☑	Positional indexes cannot be used to support proximity queries.
☐	☑	☐	The objective method for query formulation uses a statistical approach to identify relevant terms.
☐	☑	☐	A term-document matrix is sparse because most terms do not appear in most documents.
☐	☑	☐	In Boolean queries, the OR operator is used to select documents that contain at least one of the specified terms.
☐	☑	☐	The size of a T-D-Matrix depends basically on the size of corpus
☐	☑	☐	Precision is a measure that calculates the fraction of retrieved documents that are relevant.

Precision vs. Recall (5 points)

Status	Answered	
Your score	0 / 5	0%

Response

Write the main difference between Precision and Recall. Give at least one example where Precision is preferred over Recall and vice versa.

```
Measuring Precision examines how many of the predicted positives are true correct (minimizing false positives) while Recall determines how many of the actual positives are being captured (minimizing false negatives).

When identifying fraud, Precision matters greatly and can be costly when dealing with false positives. For example, Spam detection—high email precision guarantees fewer false alarms, while redesigning spam algorithms prevents false positives while letting junk emails through, misclassifying them as legitimate emails.

Where ensuring that no positive case goes undetected, Recall becomes extremely important. For example, Medical diagnosis—High recall provides assurance that fewer missed cases exist but reinforces the need for additional follow-up checks, contingent on the absence of detection.

Trade-off: More precise definitions tend to be less forgiving of false negatives, increasing recall while lowering precision and becoming overly lenient. But the problem controls this decision:

Precision > Recall: Fraud detection, spam filtering, and legal sentencing
Recall > Precision: Disease screening, disaster prediction, search & rescue (false negatives risk lives).

In practice, the F1-score (harmonic mean of precision and recall) helps balance both when neither can be sacrificed entirely. Understanding this trade-off ensures better model decisions in real-world applications.
```

194 words (max. 200)

▸ Solution

Automatic Query Formulation (5 points)

Status	Answered	
Your score	0 / 5	0%

Response

Reflect on the advancements made in automatic query formulation in the medical domain. What are the key limitations of current methodologies, and how do these limitations affect the potential for future improvements? Consider both the conceptual and objective methods in your analysis.

```
Advancements:
Recent progress in natural language processing (NLP) and machine learning (ML) have enhanced automatic query formulation in medicine which include:

1. Semantic Understanding - Models like BERT and BioBERT understand better medical terms, synonyms, and the context relationships which are intertwined.
2. Query Expansion - Such techniques make use of (ontologies UMLS, SNOMED CT) which provide a framework of terms to improve recall.
3. Tailored/Personalized Content - Queries are tailored by AI depending on user roles and their past interactions like if they are clinicians or researchers.
4. Structured Query Generation - Converting natural language to structured form that a computer can process and understand like EHRs SQL is query language and SPARQL is for biomedical knowledge graphs.

Key Limitations :

1. Challenges of a Conceptual Nature

Ambiguity & Complexity - Medical languages are often polysemous. One word can have multiple meanings.
```

"Cold" can refer to a symptom and also a disease.
Domain Specific Omissions - General models may fail to understand, and, therefore, incorporate useful implicit clinical knowledge into a query.

2. Subjective objective (Technical) Challenges

Data Sparcity & Bias - Underrrepresented populations in models for rare diseases lead to poor generalization for most models.
Lack of Interoperability - Query portability is limited due to EHR systems and inconsistent terminology flows with heterogeneous
Explainability Gaps - Black-box models lack transparency of critical decisions which put trust at risk.

Impact on Future Enhancements:

Conceptual: Developing knowledge representations such as hybrid symbolic-AI techniques can resolve ambiguity.
Technical: Bias and robustness issues could be alleviated with federated learning and multimodal data integration.
Ethical & Regulatory: Increased clinical validation trust necessitates stricter regulation for reliability.

In summary, Safeguarding, scaling, and system interpretability will ensures reasoned collaboration. Significant progress has been made, but these issues still require cooperation from various fields.

276 words

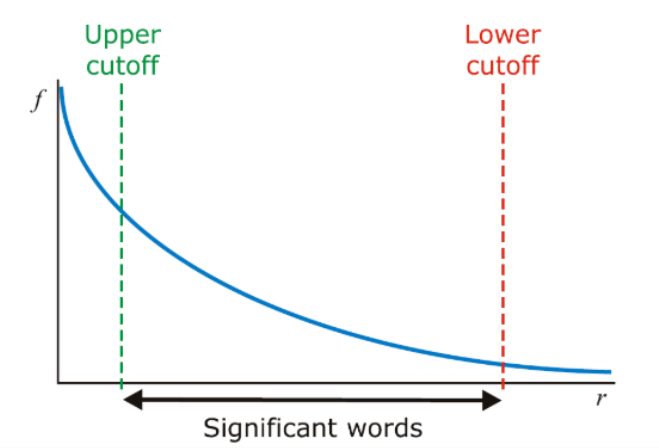
▸ Solution

Word Frequencies (10 points)

Status	Answered	
Your score	0 / 10	0%

Response

The graph depicting the distribution of word frequencies suggests that both very common and very rare words are typically filtered out in information retrieval systems to improve indexing efficiency and search result relevance.



Write an essay that addresses the following points:

1. Discuss the rationale behind setting upper and lower cutoffs in word frequency when designing an indexing system for a search engine. How do these cutoffs potentially affect the search engine's performance?
2. Analyze the potential impact of excluding very common words (stop words) and very rare words from the index. Consider both the benefits (e.g., improved processing speed, reduced storage requirements) and the drawbacks (e.g., loss of context, missing rare but important terms).

In order to maximize the effectiveness of relevance scoring in search engine indexing systems, setting upper and lower bounds for word frequencies is essential. The upper bound excludes common words referred to as stop words, such as ‘the’, ‘and’, or ‘is’, which repeatedly occur in all documents but do not help significantly in distinguishing relevant documents from noise. Their removal leads to a smaller index, faster query processing, and less cluttered results. However, overly aggressive filtering can damage some queries like, “to be or not to be,” where the stop words do hold meaning. The lower bound omits extremely uncommon words, which are specialized terms or typos, appearing in so few documents that they possess no statistical relevance. This inclusion prevents the index from inflating and irrelevant computations, but does run the risk of missing critical unique keywords, especially in domain specific searches like medical and legal.

The benefits rendered with the removal of stop words and rare words boost processing speed alongside storage space, while the narrowed focus improves search relevance. The specific mid-frequency terms that discriminate documents best are selected. Nonetheless, narrowing context relevance risk is introduced, especially with phrase based queries alongside the loss of critical unique keywords for niche searches. In solving the problems, contemporary systems utilize flexible approaches, as in the case of dynamic boundaries, or specific stop lists of the domain, which all remain adaptable. A case would be medical search engines that will store repeated words like "COVID" because of their relevance to the field. Also, these balance efficiency with comprehensiveness where infrequent terms are indexed but not prioritized unless specifically searched for. As described, the careful application of word frequency cutoffs ensures balanced efficiency and accuracy of searches, and intelligent context-sensitive filters will most likely be integrated in the future.

298 words

▸ Solution

go back to overview

Practice Tasks (32 points) 8 of 32 points (25%)

Evaluation Metrics (16 points)

Status	Answered	
Your score	8 / 16	50%

Response

- In order to answer the questions please download the following .csv file: **Iris Retrieval Dataset**
- **Tip**: It will help you to answer the questions in the given order.
- For this task, you will have to load the provided .csv which contains two columns: "retrieved" data, and "gold standard". The provided file is an output of a Machine Learning classifier that was trained on the "iris" dataset that contains 3 different types of iris flowers (Setosa, Versicolor, Virginica). The "retrieved" data column corresponds to the predicted value of the classifier and the "gold standard" to the actual/original class.
- Example:

retrieved	gold standard
setosa	versicolor
versicolor	versicolor
setosa	virginica

Here the first data instance was classified as setosa, although it belongs to the versicolor class.

- **Confusion Matrix (5 points)**

  Complete the following confusion matrix from the dataset provided

	Setosa	Versicolor	Virginica
Setosa			
Versicolor			
Virginica			

- **Precision (3 points)**
- Calculate the precision of each of the following classes.

  Note: the result should have three decimal digits, e.g., 0.685.

  Setosa precision: 1.0

  Versicolor precision: 0.968

  Virginica precision: 0.912

- **Recall (3 points)**
- Calculate the recall of each of the following classes.

  Note: the result should have three decimal digits, e.g., 0.685.

  Setosa recall: 1.0

  Versicolor recall: 0.909

  Virginica recall: 0.969

- **F-score (3 points)**
- Calculate the f1-score of each of the following classes.

  Note: the result should have three decimal digits, e.g., 0.685.

  Setosa f1-score: 1.0

  Versicolor f1-score: 0.938

  Virginica f1-score: 0.939

- **Accuracy (3 points)**
- Calculate the accuracy of the retrieval system.

  Note: the result should have three decimal digits, e.g., 0.685.

  Accuracy: 0.971

Solution

- In order to answer the questions please download the following .csv file: **Iris Retrieval Dataset**
- **Tip**: It will help you to answer the questions in the given order.
- For this task, you will have to load the provided .csv which contains two columns: "retrieved" data, and "gold standard". The provided file is an output of a Machine Learning classifier that was trained on the "iris" dataset that contains 3 different types of iris flowers (Setosa, Versicolor, Virginica). The "retrieved" data column corresponds to the predicted value of the classifier and the

gold standard to the actual/original class.

- Example:

retrieved	gold standard
setosa	versicolor
versicolor	versicolor
setosa	virginica

Here the first data instance was classified as setosa, although it belongs to the versicolor class.

- **Confusion Matrix (5 points)**

Complete the following confusion matrix from the dataset provided

	Setosa	Versicolor	Virginica
Setosa	40.0	0.0	0.0
Versicolor	0.0	30.0	3.0
Virginica	0.0	1.0	31.0

- **Precision (3 points)**
- Calculate the precision of each of the following classes.

Note: the result should have three decimal digits, e.g., 0.685.

Setosa precision: 1.0

Versicolor precision: 0.968

Virginica precision: 0.912

- **Recall (3 points)**
- Calculate the recall of each of the following classes.

Note: the result should have three decimal digits, e.g., 0.685.

Setosa recall: 1.0

Versicolor recall: 0.909

Virginica recall: 0.969

- **F-score (3 points)**
- Calculate the f1-score of each of the following classes.

Note: the result should have three decimal digits, e.g., 0.685.

Setosa f1-score: 1.0

Versicolor f1-score: 0.937

Virginica f1-score: 0.939

- **Accuracy (3 points)**
- Calculate the accuracy of the retrieval system.

Note: the result should have three decimal digits, e.g., 0.685.

Accuracy: 0.962

T-D Matrix (8 points)

Status	Answered		
Your score	0 / 8		0%

Response

Suppose that we have the following document collection:

Doc1: food prices rise expectation

Doc2: food prices rise in winter

Doc3: increase in food prices in winter

Doc4: winter food prices increase

Write down the term-document incidence matrix for this document collection.

Term	Doc1	Doc2	Doc3	Doc4
expectation	1	0	0	0
food	1	1	1	1
increase	0	0	1	1
in	0	1	1	0
prices	1	1	1	1

```
rise            1    1    0    0
winter          0    1    1    1
```

12 words

▸ Solution

Inverted Index (8 points)

Status	Answered		
Your score	0 / 8		0%

Response

Suppose that we have the following document collection:

Doc1: food prices rise expectation

Doc2: food prices rise in winter

Doc3: increase in food prices in winter

Doc4: winter food prices increase

Write down the inverted index (use -> for drawing the posting lists), that would be built for this document collection.

For writing the arrows of the posting lists, you can just use the symbols.

```
expectation -> Doc1
food        -> Doc1 -> Doc2 -> Doc3 -> Doc4
increase    -> Doc3 -> Doc4
in          -> Doc2 -> Doc3
prices      -> Doc1 -> Doc2 -> Doc3 -> Doc4
rise        -> Doc1 -> Doc2
winter      -> Doc2 -> Doc3 -> Doc4
```

25 words

▸ Solution

go back to overview

Programming task (38 points) 0 of 38 points (0%)

T-D Incidence Matrix (20 points)

Status	Answered		
Your score	0 / 20		0%

Response

Suppose that you have the following lists:

d1 = ["python", "anaconda", "jupyter", "programming", "learn", "computer"]

d2 = ["game", "are", "world", "programming", "learn", "learn", "are"]

d3 = ["learn", "are", "world", "programming"]

each representing a document. Create a function in python that takes as arguments the three lists and returns the T-D matrix of the above document collection, e.g.

my_TD_matrix(d1, d2, d3).

```
def my_TD_matrix(d1, d2, d3):
    combined_docs = d1 + d2 + d3
    vocab = set(combined_docs)

    documents = [d1, d2, d3]

    term_doc_matrix = []
    for word in vocab:
        presence = []
        for document in documents:
            presence.append(1 if word in document else 0)
        term_doc_matrix.append((word, presence))

    for term, vector in term_doc_matrix:
        print(f"{term}: {vector}")

d1 = ["python", "anaconda", "jupyter", "programming", "learn", "computer"]
d2 = ["game", "are", "world", "programming", "learn", "learn", "are"]
d3 = ["learn", "are", "world", "programming"]

my_TD_matrix(d1, d2, d3)
```

69 words

▸ Solution

Boolean Retrieval (18 points)

Status	Answered

Your score	0 / 18	0%

Response

Describe how you could use the function of the previous question, so that you can query the documents of the document collection that contain two terms, e.g. python AND learn. Give either the code or pseudocode.

```
def my_TD_matrix(d1, d2, d3):
    combined_docs = d1 + d2 + d3
    vocab = set(combined_docs)

    documents = [d1, d2, d3]
    doc_names = ["d1", "d2", "d3"]

    term_doc_dict = {}

    for word in vocab:
        presence = []
        for document in documents:
            presence.append(1 if word in document else 0)
        term_doc_dict[word] = presence

    return term_doc_dict, doc_names


def boolean_retrieval(term1, term2, d1, d2, d3):
    td_dict, doc_names = my_TD_matrix(d1, d2, d3)

    if term1 not in td_dict or term2 not in td_dict:
        print("One or both terms not found in the document collection.")
        return []

    result_vector = [
        t1 & t2 for t1, t2 in zip(td_dict[term1], td_dict[term2])
    ]
    result_docs = [
        doc_names[i] for i, val in enumerate(result_vector) if val == 1
    ]

    return result_docs


d1 = ["python", "anaconda", "jupyter", "programming", "learn", "computer"]
d2 = ["game", "are", "world", "programming", "learn", "learn", "are"]
d3 = ["learn", "are", "world", "programming"]

results = boolean_retrieval("python", "learn", d1, d2, d3)
print("Documents containing 'python AND learn':", results)
```

135 words

▸ Solution

‹ go back to overview

Test execution

Information

Availability: Expired at 5/15/2025, 1:59 PM

Max. attempts: Unlimited

Results of this test are visible to administrators and tutors of this course.

Start test

▸ Change log

Go to top