

1 DAT640-2023 exam info

Resit Exam

DAT640 - Information Retrieval and Text Mining 2023 Autumn

DATE AND TIME

- Exam starts: 20.11.2023, 09:00
- Exam closes: 20.11.2023, 13:00

You can see how much time you have left on the exam on the top of the screen. Exam answers that are submitted after the time has expired will not be accepted.

AIDS

All aids are permitted. This includes both written and printed material as well as files and programs on your own device.

IMPORTANT CONTACTS

If you need help during the exam, you can call one of the phone numbers below. This applies if you need clarifications from the course responsible or administrative support.

- Course responsible: Petra Galuscakova, tlf. 41 37 40 65
- Administrative support tlf. 51 83 31 26

WITHDRAW DURING THE EXAM

If you wish to withdraw from the exam, you must do so by choosing “deliver blank” in the top right menu and follow the instructions.

HANDING IN

The exam will automatically close for uploading when the time is up.

Note: In case something goes wrong in Inspira, such that you are unable to submit your exam, you must contact administrative support immediately.

QUESTIONS AND GRADING

The exam contains 29 questions in total.

- There are multiple choice questions or sub-questions, where there is -1 point for each wrong answer (no answer is 0 points). These are explicitly indicated.

Total points: 100

Grading (standard scale)

- 0-39: F
- 40-49: E
- 50-59: D
- 60-79: C
- 80-89: B
- 90-100: A

For all computations, provide numbers rounded to 3 digits (e.g., 0.7, 0.25, 0.333).

GOOD LUCK!

If you have any comments about the exam, write them here

Format

B


I


U


x_2


x^2


I_x
































Words: 0

Maximum marks: 0

2 Similarity

$$\mathbf{x} = (1, 0, 0, 1, 1, 0, 1, 1, 0, 1)$$

$$\mathbf{y} = (1, 1, 0, 1, 0, 0, 1, 0, 1, 1)$$

Calculate the similarity of the above two binary vectors. (2x1.5 points)

Jaccard similarity:

Cosine similarity:

Maximum marks: 3

3 Classification

Assume a multiclass classification problem with 5 categories.

Using the one-against-one strategy, how many binary classifiers are needed in total? (3 points)

Answer:

Maximum marks: 3

4 Indexing

Select all statements that are correct regarding the payload of a posting in an inverted index. (3 points)

Select one or more alternatives:

- ☐ Document IDs are stored in the payload
- ☐ Postings with payload require less memory than postings without payload
- ☐ Postings with payload supports more ranking algorithms
- ☐ The payload is not required in a posting

Maximum marks: 3

5 Coding

```

1  from collections import Counter
2  from typing import List, defaultdict
3
4
5  def score_collection(self, query_terms: List[str]):
6      """Scores all documents in the collection using term-at-a-time query
7      processing.
8
9      Args:
10         query_term: Sequence (list) of query terms.
11
12      Returns:
13         Dict with doc_ids as keys and retrieval scores as values.
14         (It may be assumed that documents that are not present in this dict
15         have a retrival score of 0.)
16      """
17      self.scores = defaultdict(float) # Reset scores.
18      query_term_freqs = Counter(query_terms)
19
20      for term, query_freq in query_term_freqs.items():
21          self.score_term(term, query_freq)
22
23      return self.scores
24
25
26  def score_term(self, term: str, query_freq: int):
27      """Scores one query term and updates the accumulated document retrieval
28      scores (`self.scores`).
29
30      Args:
31         term: Query term.
32         query_freq: Frequency (count) of the term in the query.
33      """
34      postings = self.get_postings(term)
35      for doc_id, payload in postings:
36          self.scores[doc_id] += payload * query_freq

```

What would be the time complexity of the `score_collection` method performing term-at-a-time scoring assuming that we have n query terms, m documents, and k as the length of the average posting list? (2 points; -1 if incorrect)

Select one alternative:

- ☐ $O(k*m)$
- ☐ $O(n*m)$
- ☐ $O(n*k*m)$
- ☐ $O(n*k)$

Maximum marks: 2

6 Coding

```
DOCS = {
  1: {"title": "All Along The Watchtower",
      "content": "There must be some way out of here Said the joker to the thief \
There's too much confusion I can't get no relief"
    },
  2: {"title": "Land of Confusion",
      "content": "There's too many men, too many people Making too many problems \
And not much love to go round Can't you see this is a land of confusion?"
    },
  3: {"title": "Nowhere Near",
      "content": "How easy I forget Just how you add to my confusion So I'm out of here \
Cause I know I'm nowhere near What you want, What you want, What your lookin for"
    },
}

# ...

query = {'match_phrase': {'content': "too much confusion"}}
res = es.search(index=INDEX_NAME, body={'query': query})
```

Assume you have an Elasticsearch index with three documents (without any analysis performed). Which of these document IDs will be returned in `res['hits']['hits']`? (2 points)

Select all document IDs that will be returned:

☐ 1☐ 2☐ 3

Maximum marks: 2

7 Retrieval

	doc1	doc2	doc3	doc4
term1	1	1	2	1
term2		2		1
term3	2		1	
term4	4		1	2
term5	1	2	1	

A document-term matrix is given above.

We use a Language Modeling retrieval method with Dirichlet smoothing and the smoothing parameter (μ) set to 6.

Answer the following questions: (2 points each)

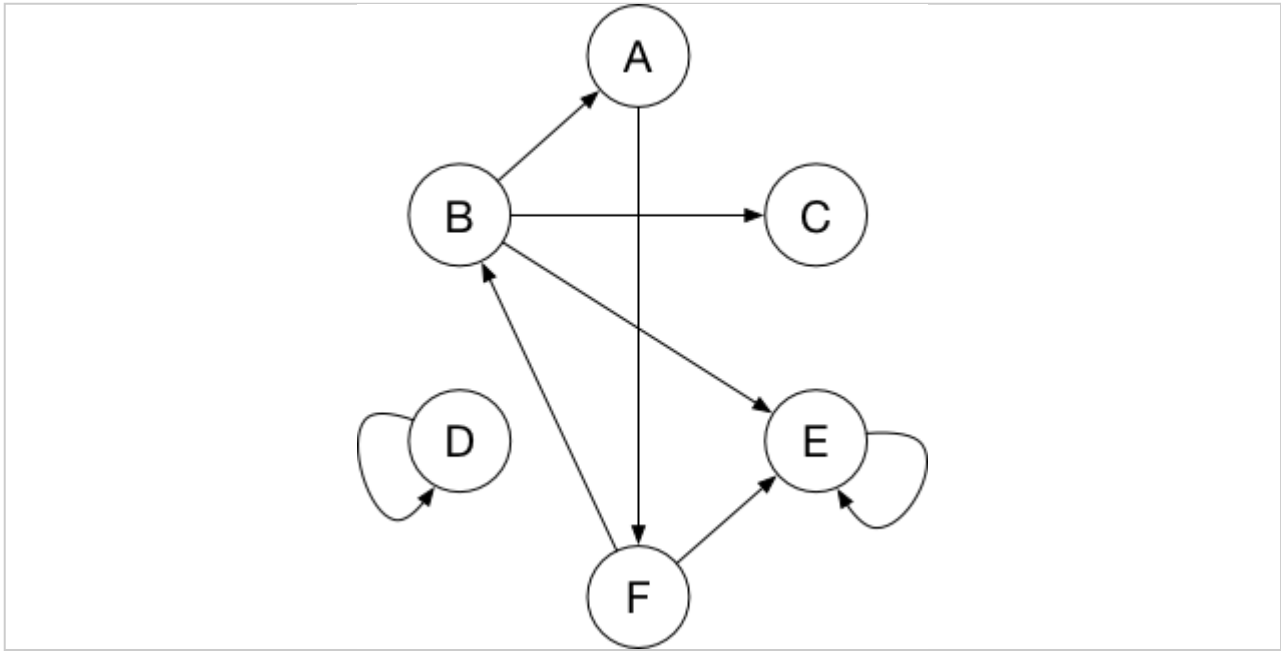
- What is the probability of term2 in the empirical language model of doc2?
- What is the probability of term5 in the background language model?
- What is the probability of term1 in the (smoothed) language model of doc4?
- Which term has the highest probability in the (smoothed) language model of doc2?

Select alternative (term1, term2, term3, term4, term5)

- Which is the top scoring document for the query ``term5 term2"?
(doc1, doc2, doc3, doc4)

Maximum marks: 10

8 PageRank



Compute the PageRank values for the above graph for the first two iterations and provide the required values in the table below. (8x1 point)

The probability of a random jump (i.e., the parameter q) is 0.2.

	Iteration 0	Iteration 1	Iteration 2
A	0.167		
B	0.167	<input type="text"/>	<input type="text"/>
C	0.167	<input type="text"/>	
D	0.167	<input type="text"/>	<input type="text"/>
E	0.167		<input type="text"/>
F	0.167	<input type="text"/>	<input type="text"/>

Maximum marks: 8

9 Retrieval

In learning-to-rank, usually an initial retrieval round is performed to retrieve the top-N documents for the query using a baseline retrieval model (e.g., BM25). Then, those top-N documents are re-ranked using supervised learning. Why is this intermediate step necessary, i.e., why not use supervised learning directly on the entire document set? (3 points)

Fill in your answer here

Format
|
B
I
U
 x_2
 x^2
 I_x
|

|

|
 $\frac{1}{2}$
 $\frac{3}{4}$
|
 Ω

 Σ

Words: 0

Maximum marks: 3

10 Relevance feedback

Which of the following statements is *false*? (2 points; -1 if incorrect)

Select one alternative:

- ☐ Relevance feedback always improves recall
- ☐ The Rocchio algorithm needs a set of annotated documents
- ☐ Implicit feedback is noisier than explicit feedback

Maximum marks: 2

11 Retrieval Evaluation

	Query 1	Query 2
Algorithm A	1, 2, 6, 5, 9, 10, 7, 4, 8, 3	1, 2, 4, 5, 7, 10, 8, 3, 9, 6
Algorithm B	10, 9, 8, 7, 5, 4, 6, 2, 1, 3	1, 3, 2, 4, 5, 6, 8, 7, 10, 9
Ground truth	1, 4, 5	3, 6

The table shows, for two queries, the document rankings produced by ranking two different algorithms along with the list of relevant documents according to the ground truth. We assume that relevance is binary.

Answer the questions below. (5x2 points)

- What is P@5 (precision at rank 5) of Algorithm A on Query 1?
- What is the Average Precision of Algorithm A on Query 1?
- What is the Reciprocal Rank of Algorithm B on Query 2?
- What is the Mean Reciprocal Rank of Algorithm B?
- Which algorithm has higher Mean Average Precision? (Algorithm A, Algorithm B, they have the same)

Maximum marks: 10

12 Statistical significance testing

Select all statements that are correct Student's t-test: (2 points)

Select one or more alternatives:

- ☐ The systems compared follow a normal distribution
- ☐ The test statistic is recorded for several permutations of the systems' outputs
- ☐ Any test statistic can be used

Maximum marks: 2

13 Retrieval evaluation

Which of the following statements about creating assessment pools for retrieval systems is *false*? (3 points)

Select one or more alternatives:

- ☐ Greater pool depth ensures that more of the relevant documents are identified
- ☐ The documents not included in the assessed pool are assumed to be non-relevant
- ☐ Only the top-k documents from each retrieval system (where k is much smaller than the number of documents in the collection) should be chosen
- ☐ The assessors are presented with documents in the order in which they are retrieved by the system

Maximum marks: 3

14 Conversational information access

Which of the following search tasks would be best addressed using a conversational user interface? (2 points)

Select one or more alternatives:

- ☐ Searching for an item with rich attributes that can be individually specified, but are much simpler to provide piecewise
- ☐ Ad-hoc search
- ☐ Planning a vacation where the results consist of a hotel, travel arrangements, restaurant plans, and places to see
- ☐ Memoryless refinement where the user learns the right terms to describe their information need by iterating with a search system but each query is ad-hoc search

Maximum marks: 2

15 Coding

	qid	query	topic_number	turn_number
0	4_1	What was the neolithic revolution?	4	1
1	4_2	When did it start and end?	4	2
2	4_3	Why did it start?	4	3
3	4_4	What did the neolithic invent?	4	4
4	4_5	What tools were used?	4	5
...
248	105_5	Who named the movement?	105	5
249	105_6	What was the US reaction to it?	105	6
250	105_7	Tell me more about the movement of the police ...	105	7
251	105_8	Why were they killed?	105	8
252	105_9	What else motivates the Black Lives Matter mov...	105	9

```

import pandas as pd
from transformers import T5ForConditionalGeneration, T5Tokenizer

SEPARATOR = "|||"
MODEL_NAME = "castorini/t5-base-canard"

# Load the model and tokenizer from HuggingFace
model = T5ForConditionalGeneration.from_pretrained(MODEL_NAME)
tokenizer = T5Tokenizer.from_pretrained(MODEL_NAME)

# Load topics
topics = pd.read_csv("topics.csv")

def create_query_rewrites(topics: pd.DataFrame) -> pd.DataFrame:
    """Create query rewrites.

    Args:
        topics: A dataframe containing the queries for each topic.

    Returns:
        Modified dataframe with the query rewrites.
    """
    rewrites = pd.DataFrame()
    for _, topic in topics.groupby("topic_number"):
        topic.reset_index(inplace=True, drop=True)
        for i, row in topic.iterrows():
            if i == 0:
                topic.at[i, "rewrite"] = row["query"]
                continue

            # TODO: Complete this function so the variable rewrite contains the
            # rewritten query.
            rewrite = ...
            topic.at[i, "rewrite"] = rewrite
        rewrites = pd.concat([rewrites, topic])
    return rewrites

```

Complete the part marked with # TODO in the `create_query_rewrites` method above. (4 points)

This method creates query rewrites for each topic using T5. A query rewrite is based on the previous and current queries of the topic.

A part of the dataset containing the topics and their queries is displayed above. Hint: you should use the methods *encode* and *decode* of the T5 tokenizer and the method *generate* of the T5 model.

NB: You only need to provide the code that goes in place of the # TODO above.

Fill in your answer here

1	
---	--

Maximum marks: 4

16 Entity retrieval

```
import re
from typing import Dict, List

def create_entity_repr(docs: List[str], window_size: int) -> Dict[str, str]:
    """Creates entity representations from mention-annotated documents.

    Example input:

    docs = [
        "first document that mentions [[entity1|entity-one]] alone",
        "2nd document with [[entity2|ABC]] and [[entity1|entity-one]] together",
        "xxx yyy zzz [[entity2|ABC]] aaa bbb ccc ddd [[entity3|ZZZ]] eee fff",
    ]

    By calling this method with window_size=2, the generated output will be

    {
        "entity1": "that mentions entity one-alone ABC and entity-one together",
        "entity2": "document with ABC and entity-one yyy zzz ABC aaa bbb",
        "entity3": "ccc ddd ZZZ eee fff",
    }

    Args:
        documents: List of documents with mention-level entity annotations.
        window_size: Size of the context window (in terms).

    Returns:
        A dictionary indexed with entity IDs, holding the corresponding entity
        representations as values.
    """
    # TODO
```

Write a method that takes a collection of documents that contain mention-level entity annotations and creates term-based entity representations from them. (7 points)

Entities are marked up using wiki-style piped links, i.e., are in `[[entityID|text]]` format. For simplicity, the text component of these links does not contain spaces. (You may assume that the input is syntactically correct with regards to these annotations.)

For each entity that is mentioned in the input documents, create an entity representation document by concatenating the terms within a given windows size around each of its mentions (i.e., take `window_size` terms before and `window_size` terms after its mention as well as the text annotated with the entity itself). Entity-annotated texts inside `[[]]` should be treated as a single term!

You may assume that the input text has been preprocessed and that you can simply split to terms on spaces.

Above, you can see the signature of the method that you need to implement along with an example input and corresponding output. You may assume that the `re` package has been imported. The use of additional packages is not allowed.

NB: You only need to write the body of the method (i.e., that comes in place of `# TODO`) in the input area below.

You need to submit code that runs and produces output in the required format, as this exercise will be graded automatically based on how many of the tests it passes.

Fill in your answer here

1	
---	--

Maximum marks: 7

17 Retrieval

Which of the following statements about the sequential dependence model (SDM) is *false*? (3 points; -1 if incorrect)

Select one alternative:

- ☐ It belongs to the class of linear feature-based models
- ☐ It is a particular Markov random field model
- ☐ The feature functions estimate term/bigram frequencies combined across multiple fields
- ☐ The ranking function is a weighted combination of feature functions

Maximum marks: 3

18 Retrieval

You are given a small collection of documents, $D = \{d_1, d_2, d_3\}$, and a query q , each consisting of a sequence of terms t_i :

$$\begin{aligned} d_1 &= \langle t_1 \ t_2 \ t_3 \ t_4 \ t_5 \ t_6 \rangle \\ d_2 &= \langle t_3 \ t_4 \ t_3 \ t_1 \ t_8 \ t_2 \ t_2 \ t_7 \rangle \\ d_3 &= \langle t_2 \ t_9 \ t_4 \ t_1 \ t_8 \ t_2 \ t_3 \ t_1 \ t_4 \rangle \\ q &= \langle t_4 \ t_3 \rangle \end{aligned}$$

The SDM scoring function:

$$score(d, q) = \lambda_T \sum_{i=1}^n f_T(q_i, d) + \lambda_O \sum_{i=1}^{n-1} f_O(q_i, q_{i+1}, d) + \lambda_U \sum_{i=1}^{n-1} f_U(q_i, q_{i+1}, d) \quad (1)$$

The weights are given as $\lambda_T = 0.85$, $\lambda_O = 0.1$, $\lambda_U = 0.05$.

The specific feature functions are:

Unigram matches:

$$f_T(q_i, d) = \log P(q_i | \theta_d) \quad (2)$$

Ordered bigram matches:

$$f_O(q_i, q_{i+1}, d) = \log \left(\frac{c_o(q_i, q_{i+1}, d) + \mu P_o(q_i, q_{i+1} | D)}{|d| + \mu} \right) \quad (3)$$

Unordered bigram matches:

$$f_U(q_i, q_{i+1}, d) = \log \left(\frac{c_w(q_i, q_{i+1}, d) + \mu P_w(q_i, q_{i+1} | D)}{|d| + \mu} \right), \quad (4)$$

Note the use of the logarithm (base 2) and the use of the Dirichlet smoothing with parameter $\mu = 6$. Also $|d|$ is the length of a document. Use a window of $w = 4$ terms for the unordered bigrams.

What is the value of the unordered bigram feature function for "t4 t3" in document d3?
(3 points, -1 if incorrect)

Select one alternative

- ☐ -1.716
- ☐ 0.3043
- ☐ -2.786

Maximum marks: 3

19 Entity linking

Entity	count
Superman	1000
Superman (comic book)	120
Superman (1978 film)	50
Superman (film series)	27
Superman (1999 video game)	3

The table shows all the different entities and counts from a surface form dictionary for the entry (i.e., surface form) "superman".

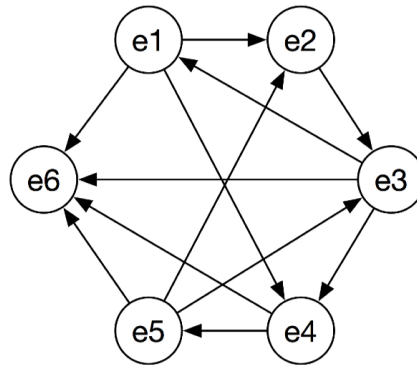
Which entity has a commonness score of 0.1? (2 points; -1 if incorrect)

Select an alternative:

- ☐ Superman
- ☐ Superman (comic book)
- ☐ Superman (1978 film)
- ☐ Superman (film series)
- ☐ Superman (1999 video game)
- ☐ None of them

Maximum marks: 2

20 Entity linking



$$WLM(e, e') = 1 - \frac{\log(\max(|\mathcal{L}_e|, |\mathcal{L}_{e'}|)) - \log(|\mathcal{L}_e \cap \mathcal{L}_{e'}|)}{\log(|\mathcal{E}|) - \log(\min(|\mathcal{L}_e|, |\mathcal{L}_{e'}|))}$$

What is the relatedness (Wikipedia Link-based Similarity) score between entities 3 and 6, based on their incoming links? (3 points)

(Use base 2 for log.)

WLM(e3, e6) = .

Maximum marks: 3

21 Entity-oriented search

Which of the following considerations is *incorrect* for an evaluation measure designed specifically for the *target type identification* task? (3 points, -1 if incorrect)

Select one alternative:

- ☐ Not all types of mistakes are equally bad, near-misses should be graded
- ☐ Hierarchical relationships in the type taxonomy should be taken into account
- ☐ It should not matter whether a wrong answer is located on the same branch than the correct answer or on a different one
- ☐ Multiple correct ground truth types should be supported

Maximum marks: 3

22 Entity linking

Name two main differences between entity linking in queries and entity linking in documents. (2 points)

Fill in your answer here

Format

B


I


U


x_2


x^2


$\frac{1}{x}$
























Σ



Words: 0

Maximum marks: 2

23 Coding

```

1  from typing import List, Tuple
2
3  SkipGrams = Tuple[str, str]
4
5
6  def generate_positive_examples(sentence: str, l: int) -> List[SkipGrams]:
7      """Generates positive examples for the given sentence.
8
9      Args:
10         sentence: Sentence.
11         l: Window size.
12
13     Returns:
14         List of positive examples.
15     """
16     positive_examples = []
17     tokens = sentence.split()
18     for i, token in enumerate(tokens):
19         for j in range(i, i + l + 1):
20             if j < 0 or j >= len(tokens) or i == j:
21                 continue
22             positive_examples.append((token, tokens[j]))
23     return positive_examples

```

The code above is used to create positive examples for a given sentence in order to train a Word2Vec Skip-gram with negative sampling model.

Is there an error in the implementation of the method `generate_positive_examples`, and if yes, what is it? (2 points)

Fill in your answer here

Maximum marks: 2

24 Neural IR

Sentence: "The curious cat explored the mysterious garden under the pale moonlight."

Lexicon: [the, curious, cat, explored, mysterious, garden, under, pale, moonlight]

Window size: 3

Given the sentence, lexicon, and window size above, select the examples that are not valid negative training examples for a Skip-gram model. (2 points)

Select one or more alternatives:

- ☐ (moonlight, the)
- ☐ (cat, moonlight)
- ☐ (garden, curious)
- ☐ (curious, mysterious)

Maximum marks: 2

25 Retrieval

Which of the following statements about neural ranking is/are correct? (2 points)

Select one or more alternatives:

- ☐ The interaction matrix is created using a Learning-to-rank method.
- ☐ Interaction-focused ranking uses Transformers to calculate relationships between queries and a documents.
- ☐ An approximate nearest neighbor index substantially improves the speed of finding document embeddings closest to query embeddings.
- ☐ Approximate Nearest Neighbor algorithms provide exact matches of the nearest neighbors for any given query.

Maximum marks: 2

26 Neural IR

Name two of the main differences between BERT-based word embeddings and word2vec word embeddings. (2 points)

Fill in your answer here

Format
|
B
I
U
 x_2
 x^2
 I_x
|

|

|

|

|

Σ
|

Words: 0

Maximum marks: 2

27 Large Language Models

Create a few-shot prompt to categorize animals into "domestic" and "wild" categories using ChatGPT or similar GPT-based model. (3 points)

Fill in your answer here

Maximum marks: 3

28 Cross-language IR

You would like to create information retrieval system for search in Norwegian documents using English queries. Describe very briefly three possible approaches that you could use. (3 points)

Fill in your answer here

Maximum marks: 3

29 User simulation

Information need

$$C_0 = \begin{bmatrix} \text{type} = \text{hotel} \\ \text{location} = \text{central} \\ \text{price} = \text{cheap} \end{bmatrix}$$

$$R_0 = \begin{bmatrix} \text{name} = \\ \text{website} = \\ \text{address} = \end{bmatrix}$$

Agenda 1

$$\begin{bmatrix} \text{inform}(\text{type} = \text{hotel}) \\ \text{request}(\text{name}) \\ \text{request}(\text{website}) \\ \text{request}(\text{address}) \\ \text{bye}() \end{bmatrix}$$

Agenda 2

$$\begin{bmatrix} \text{inform}(\text{name}) \\ \text{inform}(\text{website}) \\ \text{inform}(\text{address}) \\ \text{request}(\text{type} = \text{hotel}) \\ \text{request}(\text{location} = \text{central}) \\ \text{request}(\text{price} = \text{cheap}) \\ \text{bye}() \end{bmatrix}$$

Agenda 3

$$\begin{bmatrix} \text{inform}(\text{type} = \text{hotel}) \\ \text{inform}(\text{location} = \text{central}) \\ \text{inform}(\text{price} = \text{cheap}) \\ \text{request}(\text{name}) \\ \text{request}(\text{website}) \\ \text{request}(\text{address}) \\ \text{bye}() \end{bmatrix}$$

Given the information need above, how would the agenda be initialized in an agenda-based user simulator? (3 points; -1 if incorrect)

Select one alternative:

- ☐ Agenda 1
- ☐ Agenda 2
- ☐ Agenda 3

Maximum marks: 3

30 Conversational information access

Suggest two different ways of exploiting item reviews by a conversational recommender system. (3 points)

Fill in your answer here

Maximum marks: 3