1 DAT640-2022 resit exam info

Resit Exam
DAT640 - Information Retrieval and Text Mining
2022 Autumn

DATE AND TIME

- Exam starts: 13.02.2023, 09:00
- Exam closes: 13.02.2023, 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: Krisztian Balog, tlf. 41 54 86 63
- 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 Inspera, such that you are unable to submit your exam, you must contact administrative support immediately.

QUESTIONS AND GRADING
The exam contains 26 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 Ω Σ

Words: 0

Maximum marks: 0

2 Indexing

You are given an excerpt from an inverted index. Select all statements that apply for this kind of index. *(3 points)*

Select one or more alternatives:

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

Maximum marks: 3

3 Relevance feedback

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

Select one alternative:

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

Maximum marks: 2

4 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
- ☐ The assessors are presented with documents in the order in which they are retrieved by the system
- ☐ 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

---

Maximum marks: 3

5 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:

- ☐ Ad-hoc search
- ☐ Planning a vacation where the results consist of a hotel, travel arrangements, restaurant plans, and places to see
- ☐ Searching for an item with rich attributes that can be individually specified, but are much simpler to provide piecewise
- ☐ 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

6 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 (mu) 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''? Select alternative (doc1, doc2, doc3, doc4)

Maximum marks: 10

7 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

8 Retrieval system design

Suppose you are preparing a music playlist using a music streaming service for the next social gathering you are attending. You already have a few songs added to your playlist, and the service will recommend some songs based on your initial selection.

Describe how you would design a retrieval system that takes a sequence of songs as input and retrieves a ranked list of recommended songs. *(5 points)*

Specifically, describe

- (a) How would you represent a song? (What associated metadata would you leverage?)
- (b) How would you score (rank) songs based on this input?

Fill in your answer here

Format

Words: 0

Maximum marks: 5

9 Coding

```python
from collections import Counter
from typing import List, defaultdict


def score_collection(self, query_terms: List[str]):
    """Scores all documents in the collection using term-at-a-time query
    processing.

    Args:
        query_term: Sequence (list) of query terms.

    Returns:
        Dict with doc_ids as keys and retrieval scores as values.
        (It may be assumed that documents that are not present in this dict
        have a retrival score of 0.)
    """
    self.scores = defaultdict(float)  # Reset scores.
    query_term_freqs = Counter(query_terms)

    for term, query_freq in query_term_freqs.items():
        self.score_term(term, query_freq)

    return self.scores


def score_term(self, term: str, query_freq: int):
    """Scores one query term and updates the accumulated document retrieval
    scores (`self.scores`).

    Args:
        term: Query term.
        query_freq: Frequency (count) of the term in the query.
    """
    postings = self.get_postings(term)
    for doc_id, payload in postings:
        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*k)
- O(n*m)
- O(n*k*m)

Maximum marks: 2

10 Retrieval

	System rankings			Ground truth		
Query	**System A**	**System B**	**System C**	Excellent (3)	Good (2)	Poor (1)
Q1	1, 2, 3, 4, 5	4, 5, 2, 3, 1	2, 3, 1, 4, 5	1	2, 3	
Q2	1, 3, 2, 4, 5	5, 4, 1, 2, 3	2, 4, 1, 3, 5		2	4
Q3	4, 2, 3, 1, 5	3, 5, 2, 4, 1	4, 3, 5, 1, 2	1	5	4

The table above contains the rankings generated by three systems (A, B, C) on three queries (Q1, Q2, Q3), along with the corresponding ground truth labels. The relevance grades are as follows: non-relevant (0), poor (1), good (2), excellent (3).

Select the correct answers the following questions (5x2 points)

	System A	System B	System C
Which system has the highest NDCG@5 score for Q1?	○	○	○
Which system has the highest NDCG@5 score for Q2?	○	○	○
Which system has the highest NDCG@5 score for Q3?	○	○	○
Which system has the highest (average) NDCG@5 score across all queries?	○	○	○
Which system has the lowest (average) NDCG@5 score across all queries?	○	○	○

Maximum marks: 10

11 Fairness

Rank	Group
1	Woman
2	Man
3	Woman
4	Woman
5	Woman
6	Man
7	Man
8	Man
9	Woman
10	Woman

Table 2: Ranking of candidates for a job.

Is the ranking fair to both groups (women and men)? *(2 points; -1 if incorrect)*

Select one alternative:

- Yes
- No

Maximum marks: 2

12 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

13 Neural IR

What are the main differences between interaction-focused and representation-focused neural IR systems? *(2 points)*

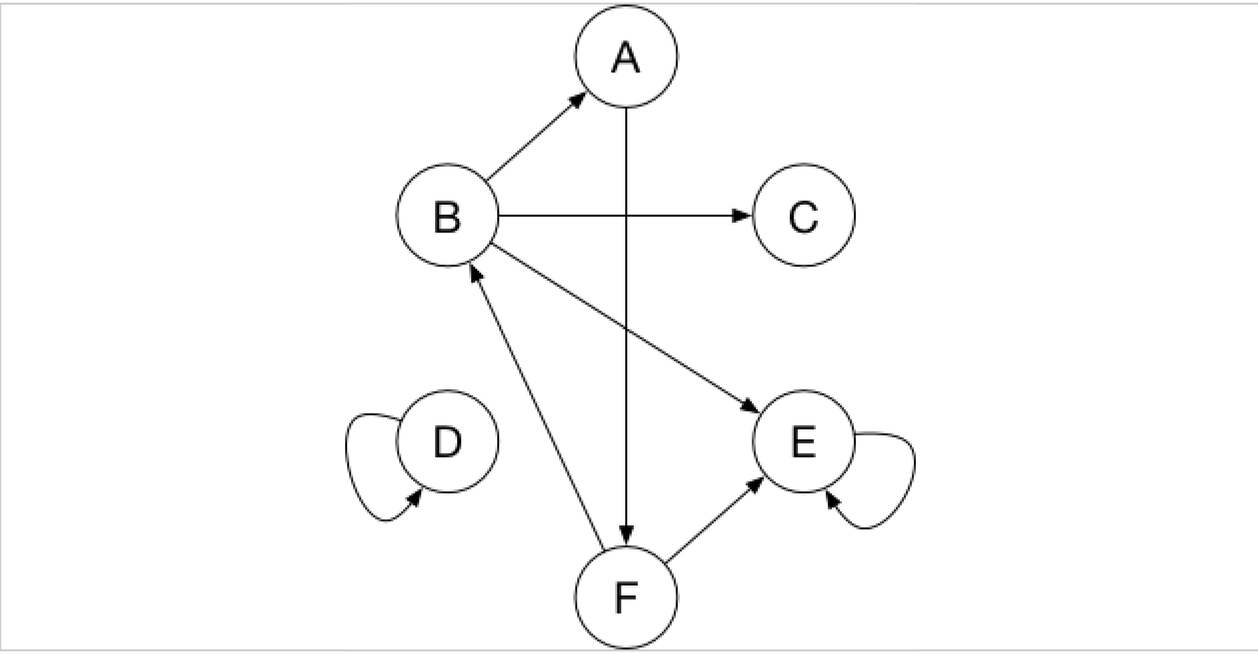
Fill in your answer here

Format

Words: 0

Maximum marks: 2

14 PageRank



Compute the PageRank values for the above graph for the first two iterations. *(12x1 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		
C	0.167		
D	0.167		
E	0.167		
F	0.167		

Maximum marks: 12

15 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? Select alternative (Algorithm A, Algorithm B, they have the same)

Maximum marks: 10

16 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

17 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

18 Coding

```
from pprint import pprint
from elasticsearch import Elasticsearch

es = Elasticsearch()
tv = es.termvectors(index="toy_index", doc_type="_doc", id=3, fields="content", term_statistics=True)
pprint(tv["term_vectors"]["content"]["field_statistics"])
```

Assume you have an Elasticsearch instance running locally with a toy collection indexed as in the exercises done during lectures. Then you run the above Python code.

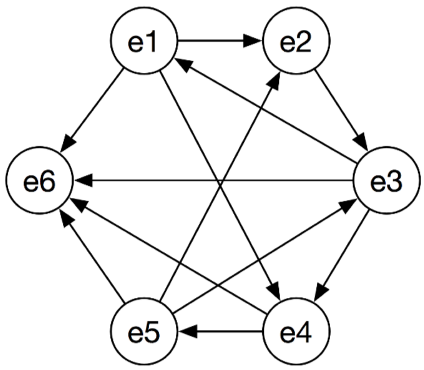
Which of the following outputs would you then expect to see printed on your monitor? *(3 points; -1 if incorrect)*

Select one alternative:

- {'doc_count': 12, 'sum_doc_freq': 1478, 'sum_ttf': 2198}
- {'doc_count': 14, 'sum_doc_freq': 1441, 'sum_idf': 2199}
- {'term_count': 13, 'sum_doc_freq': 1470, 'sum_ttf': 2197}
- {'term_count': 11, 'sum_term_freq': 1474, 'sum_idf': 2194}

Maximum marks: 3

19 Entity linking



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

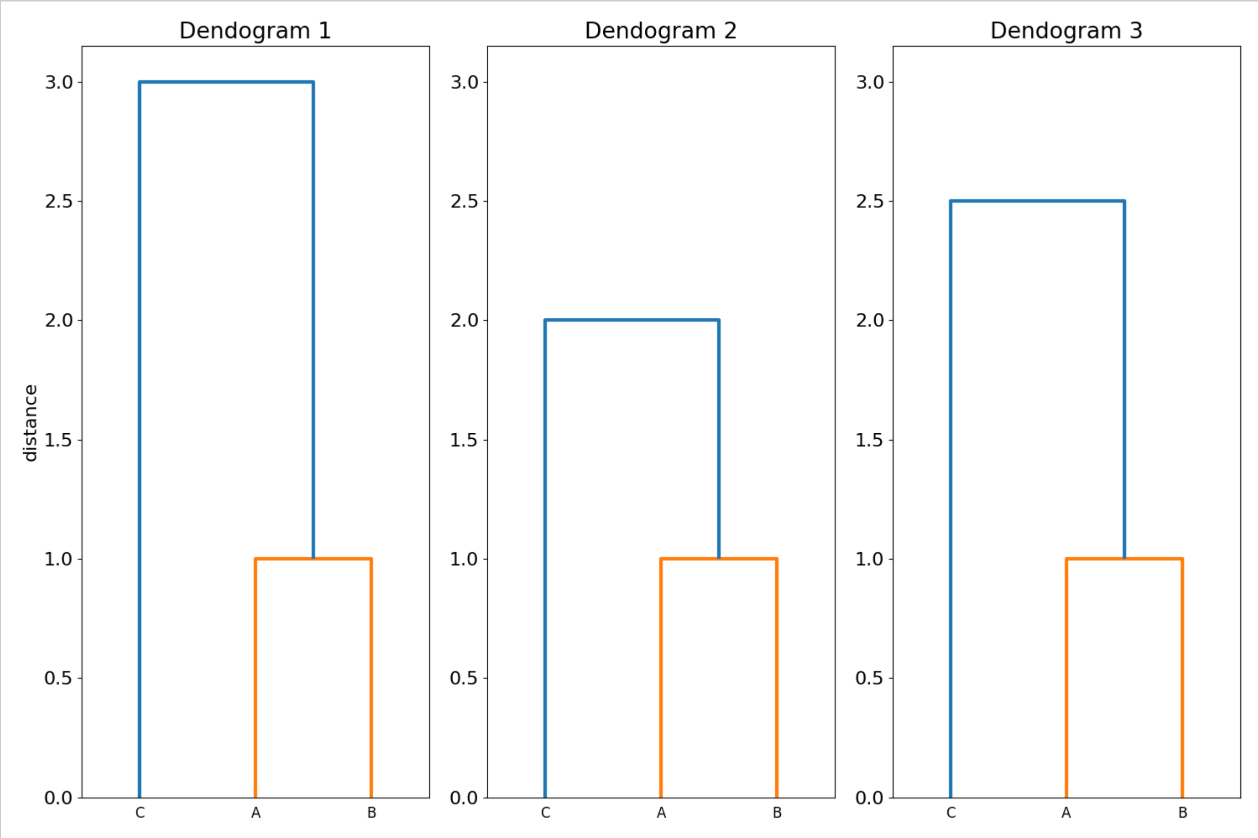
What is the relatedness (Wikipedia Link-based Similarity) score between entities 3 and 6, based on their incoming links? *(3 points)*

(Use base 2 for log.)

WLM(e3, e6) = ☐ .

Maximum marks: 3

20 Clustering



You are given a distance matrix and three dendograms obtained by agglomerative hierarchical clustering with different inter-cluster similarity measures.

	A	B	C
A	0	1	2
B	1	0	3
C	2	3	0

Which dendogram was created using single-link inter-cluster similarity? *(3 points; -1 if incorrect)*

Select one alternative:

- Dendogram 1
- Dendogram 3
- Dendogram 2

Maximum marks: 3

21 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
- ☐ Any test statistic can be used
- ☐ The test statistic is recorded for several permutations of the systems' outputs

Maximum marks: 2

22 Retrieval

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

Select one alternative:

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

Maximum marks: 3

23 Conversational information access

Connect the given statements with the dialogue system components. (You are expected to know what the acronyms stand for.) *(2 points)*

Please match the values:

	NLU	DP	NLG	ST
Decides what action the system should take next	○	○	○	○
Passes dialogue act containing intent and slot value pairs for the current utterance	○	○	○	○
Contains the value of the frame since the beginning of the conversation	○	○	○	○
Answers the questions 'What to say?' and 'How to say it?'	○	○	○	○

Maximum marks: 2

24 Neural IR

What are the main differences between BERT-based word embeddings and word2vec word embeddings? *(2 points)*

Fill in your answer here

Format

Words: 0

Maximum marks: 2

25 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

Words: 0

Maximum marks: 3

26 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:

$d_1 = \langle t_1\ t_2\ t_3\ t_4\ t_5\ t_6 \rangle$
$d_2 = \langle t_3\ t_4\ t_3\ t_1\ t_8\ t_2\ t_2\ t_7 \rangle$
$d_3 = \langle t_2\ t_9\ t_4\ t_1\ t_8\ t_2\ t_3\ t_1\ t_4 \rangle$
$q = \langle t_4\ t_3 \rangle$

The SDM scoring function:

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

The weights are given as $\lambda_T = 0.85$, $\lambda_O = 0.1$, $\lambda_U = 0.05$.
The specific feature functions are:
Unigram matches:

$$f_T(q_i, d) = \log P(q_i|\theta_d) \tag{2}$$

Ordered bigram matches:

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

Unordered bigram matches:

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

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.

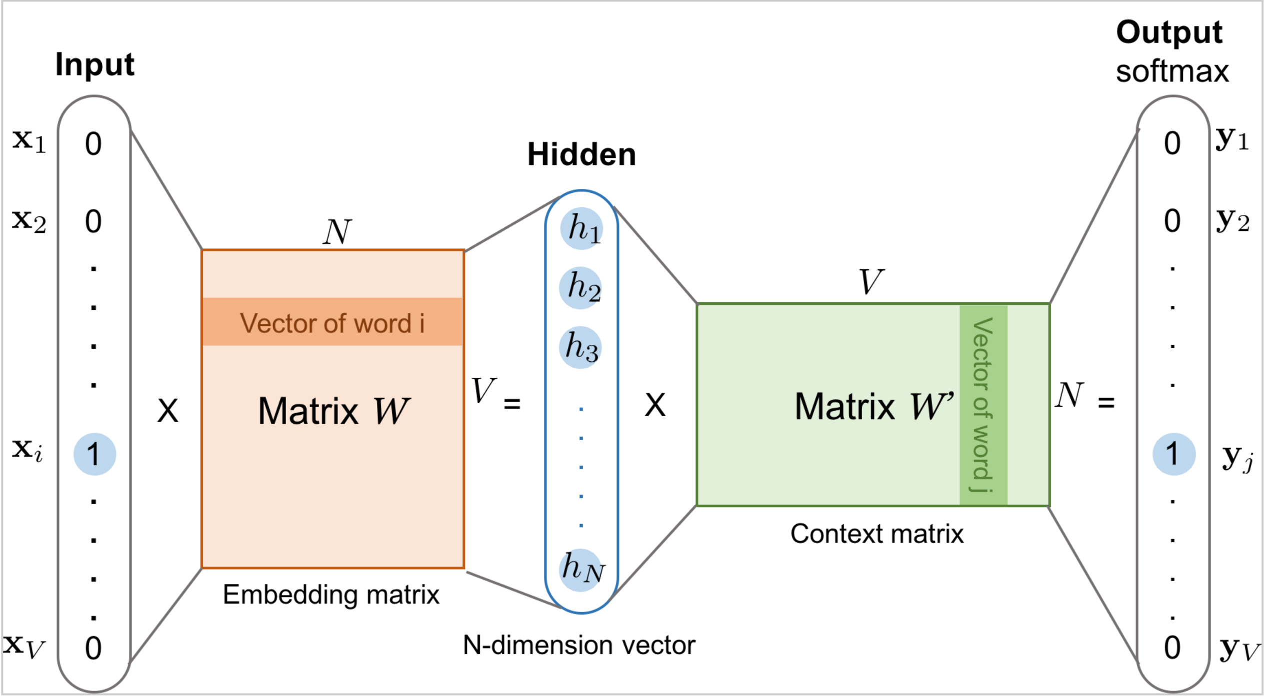
What is the value of the unordered bigram feature function for "t4 t3" in document d3?
(3 points, -1 if incorrect)

Select one alternative

- 0.3043
- -2.786
- -1.716

Maximum marks: 3

27 Retrieval



The figure shows a Word2Vec algorithm. Which variant is this, and what do the matrices represent? *(3 points; -1 if incorrect)*

Select one alternative:

- The CBOW variant of Word2Vec, where **W'** embeds center words and **W** embeds context words.
- The SkipGram variant of Word2Vec, where **W'** embeds center words and **W** embeds context words.
- The CBOW variant of Word2Vec, where **W** embeds center words and **W'** embeds context words.
- The SkipGram variant of Word2Vec, where **W** embeds center words and **W'** embeds context words.

Maximum marks: 3