DAT640 2022 autumn, Resit Exam - Answer Key

General approach to grading the essay questions
Even if a solution contains the answer that should give a certain score based on the provided solution key, points will be deducted if the answer (1) has a mix of correct and incorrect statements (-1 point) or (2) is too long or has largely irrelevant content w.r.t. the question (-1 point).

2 Similarity (2x1.5 points)

$$\begin{aligned} x &= (1,0,0,1,1,0,1,1,0,1) \\ y &= (1,1,0,1,0,0,1,0,1,1) \end{aligned}$$

Calculate the similarity of the above two binary vectors.

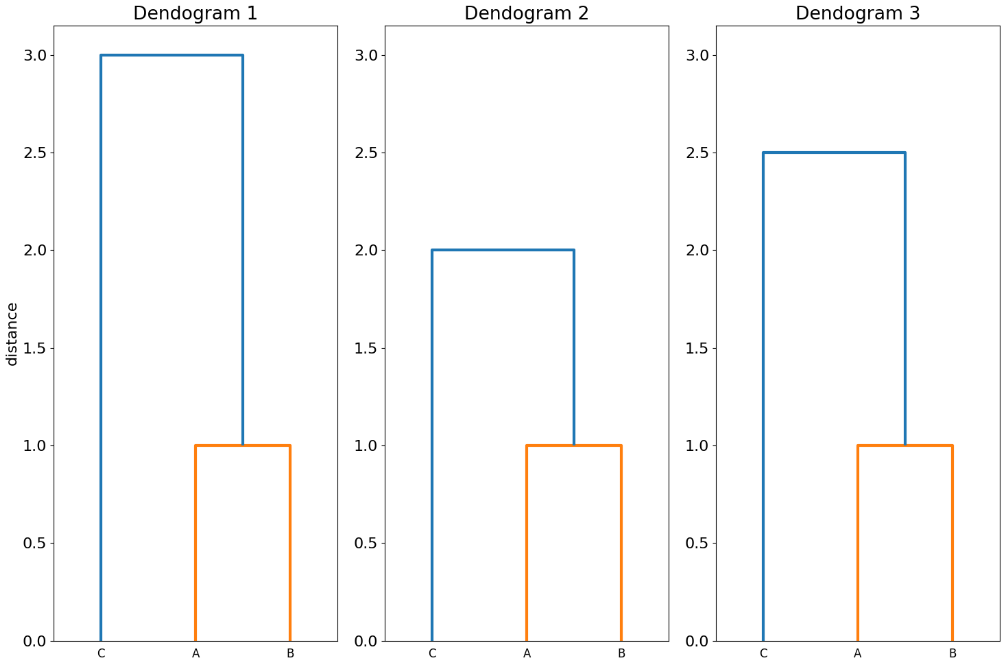
- Jaccard similarity: 0.5
- Cosine similarity: 0.666

3 Classification (3 points)

Assume a multiclass classification problem with 5 categories. Using the one-against-one strategy, how many binary classifiers are needed in total?

Answer: [$\frac{5*4}{2} = \mathbf{10}$]

4 Clustering (3 points; -1 if incorrect)



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

Which dendogram was created using single-link inter-cluster similarity?

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

- ( ) Dendogram 1.
- (X) Dendogram 2.
- ( ) Dendogram 3.

5 Retrieval (5x2 points)

	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: (2p each)

- What is the probability of term2 in the empirical language model of doc2? [0.4]
- What is the probability of term5 in the background language model? [0.182]
- What is the probability of term1 in the (smoothed) language model of doc4? [0.24]
- Which term has the highest probability in the (smoothed) language model of doc2? [term5]
- Which is the top scoring document for the query "term5 term2"? [doc2]

6 Retrieval Evaluation (5x2 points)

	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

Table 1: Retrieval evaluation.

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.

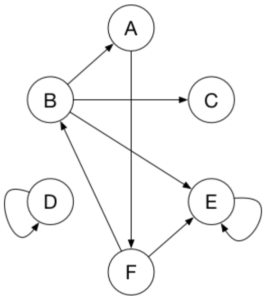
- What is P5 (precision at rank 5) of Algorithm A on Query 1? [0.4]
- What is the Average Precision of Algorithm A on Query 1? [0.625]
- What is the Reciprocal Rank of Algorithm B on Query 2? [0.5]
- What is the Mean Reciprocal Rank of Algorithm B? [0.35]
- Which algorithm has higher Mean Average Precision? [A/B/the same]

7 Retrieval (3 points; -1 if incorrect)

Which of the following statements about the sequential dependence model (SDM) is *false*?

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

8 PageRank (12x1 point)



Compute the PageRank values for the above graph for the first two iterations.
The probability of a random jump (i.e., the parameter q) is 0.2.

	Iteration 0	Iteration 1	Iteration 2
A	0.167	0.1	0.079
B	0.167	0.122	0.122
C	0.167	0.1	0.079
D	0.167	0.188	0.197
E	0.167	0.3	0.394
F	0.167	0.188	0.126

9 Coding (3 points; -1 if incorrect)

```
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?

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

10 Indexing (3 points)

Select all the correct statements regarding the payload of a posting in an inverted index.

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

Scoring:

- 3 points if all correct; otherwise, 1 point for each correct, -0.5 point for each incorrect answer
- For example: 1.5 points if 2 correct and 1 incorrect; 0.5 point if 1 correct and 1 incorrect
- Minimum points is 0

11 Retrieval (3 points)

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:

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

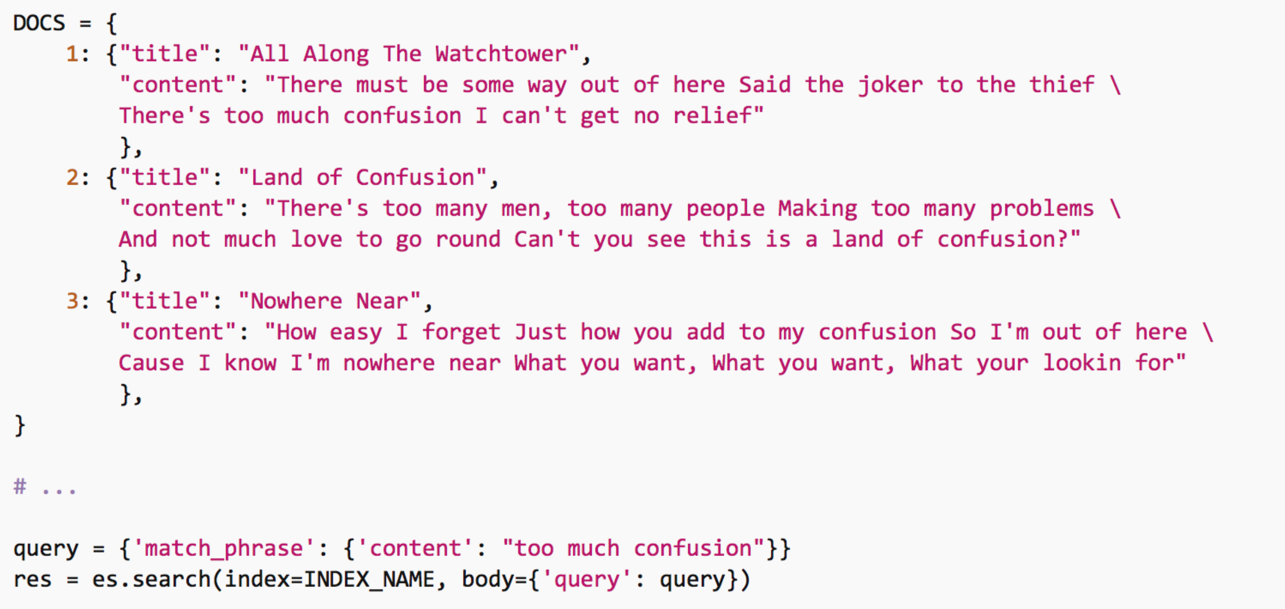
Unordered bigram matches:

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

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?

- -2.876
- -1.716
- 0.3043



```
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})
```

12 Coding (2 points)

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’]?

- [X] 1
- [ ] 2
- [ ] 3

Scoring: 2 points for correct selection, otherwise 0.

13 Coding (2 points; -1 if incorrect)

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?

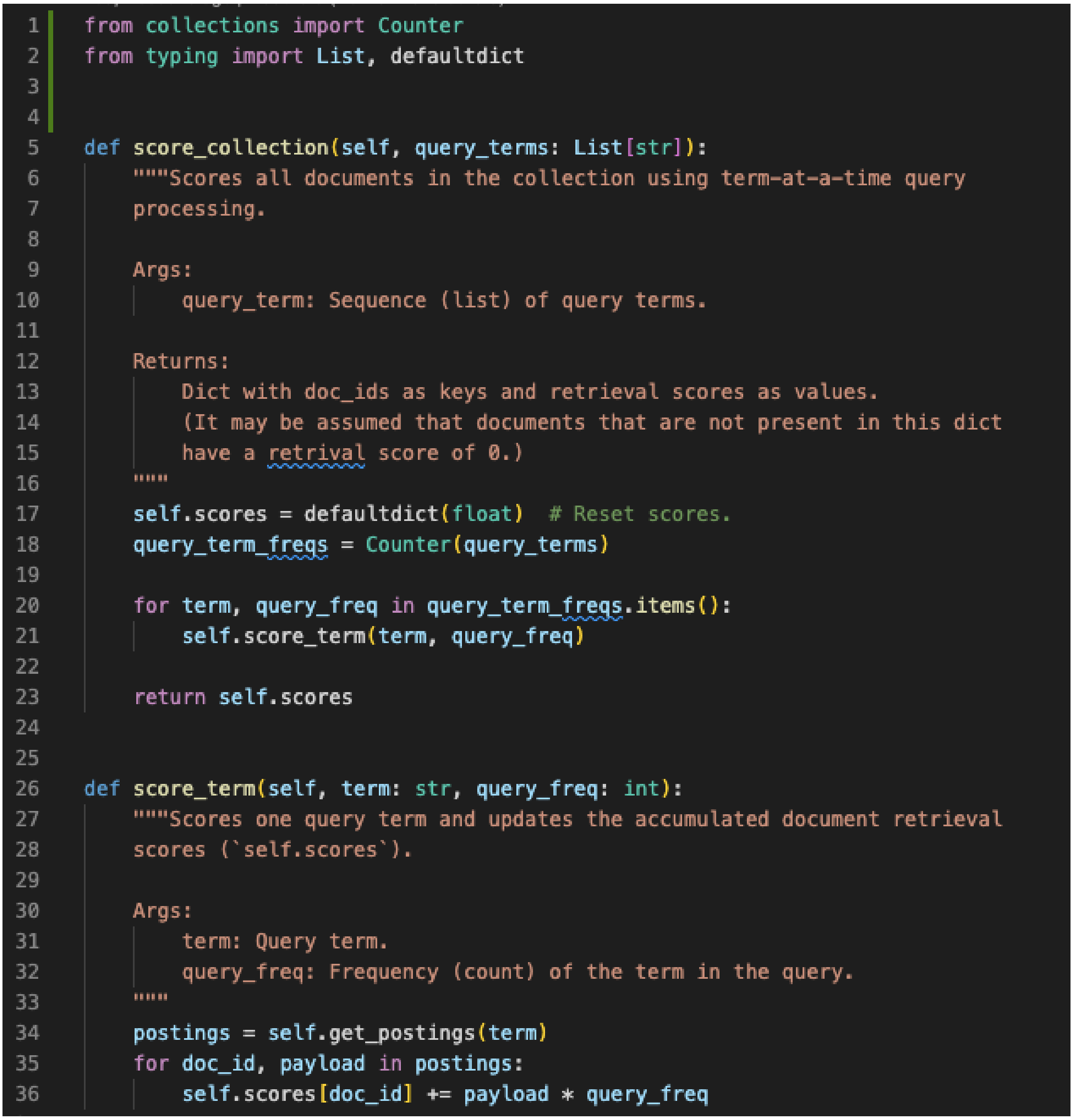
- ( ) $\mathcal{O}(n \cdot m)$
- ( ) $\mathcal{O}(k \cdot m)$
- (X) $\mathcal{O}(n \cdot k)$
- ( ) $\mathcal{O}(n \cdot k \cdot m)$

14 Conversational Search Systems (2 points)

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

- [ ] Ad-hoc search
- [X] 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
- [X] Planning a vacation where the results consist of a hotel, travel arrangements, restaurant plans, and places to see

Scoring:



```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
```

- 2 points if all correct
- 1 point if 2 correct and 1 incorrect
- 0 points otherwise

15 Fairness (2 points; -1 if incorrect)

Is the ranking fair to both groups (women and men)?

- ( ) Yes
- (X) No

16 Retrieval Evaluation (5x2 points)

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

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.

(1), good (2), excellent (3).

	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

Select the correct answers the following questions:

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

17 Conversational information access (2 points)

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

- (1) Contains the value of the frame since the beginning of the conversation
- (2) Passes dialogue act containing intent and slot value pairs for the current utterance
- (3) Answers the questions "What to say?" and "How to say it?"
- (4) Decides what action the system should take next

- [1] ST
- [2] NLU
- [3] NLG
- [4] DP

Scoring: 0.5p for each correct association.

18 Retrieval system design (5 points)

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.

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?

Scoring:

- 2 points for representation of songs (+1 for truly creative ideas beyond term-based representations)
- 2 points for representation of playlist.
- 1 point for describing a mechanism to match the set of songs in a playlist to a candidate song.

Example answer:

- (a)
  - A song could be represented as a multi-field document based on its metadata tags: song title, album, artist, release year.
  - Taking text data from metadata/tags as fields, an inverse index can be constructed.
  - A playlist may be considered as a fielded document with each of its fields consisting of the concatenation of the corresponding fields in the songs it contains.
- (b)
  - With the representation described, the relevance of a candidate song to an initial playlist can be scored based on a fielded term-based model, such as BM25F.

19 Retrieval (3 points)

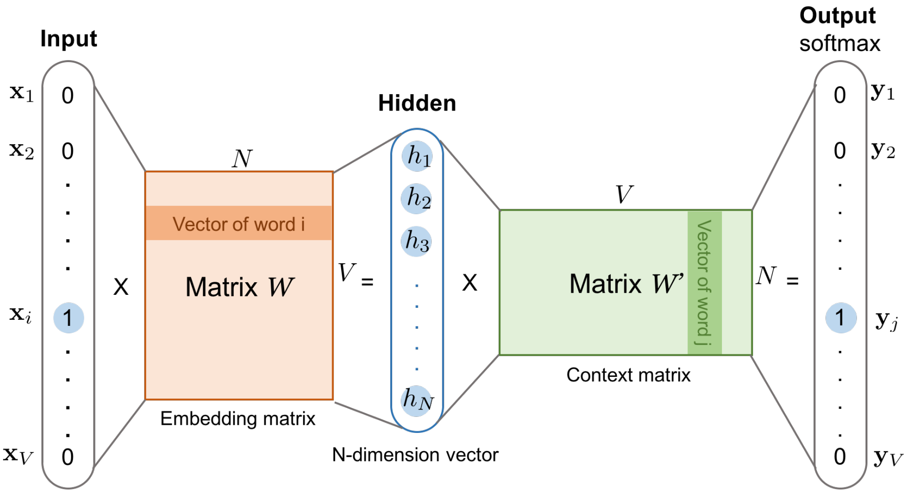


Figure 1: Word2Vec.

The figure shows a Word2Vec algorithm. Which variant is this, and what do the matrices represent?

- The SkipGram variant of Word2Vec, where $\mathbf{W}$ embeds center words and $\mathbf{W}'$ embeds context words.
- The CBOW variant of Word2Vec, where $\mathbf{W}$ embeds center words and $\mathbf{W}'$ embeds context words.
- The SkipGram variant of Word2Vec, where $\mathbf{W}'$ embeds center words and $\mathbf{W}$ embeds context words.
- The CBOW variant of Word2Vec, where $\mathbf{W}'$ embeds center words and $\mathbf{W}$ embeds context words.

20 Neural IR (2 points)

What are the main differences between interaction-focused and representation-focused neural IR systems?

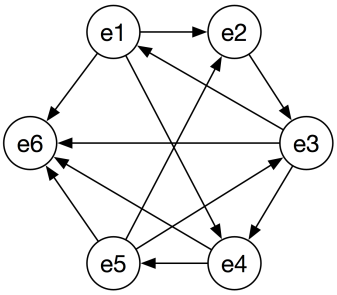
- (1p) explicit construction of interaction between query and documents vs. computing the representations independently
- (1p) identifying relevant documents only from query-dependent sample of documents vs. all documents in a collection
- (1p) unsupervised retrieval vs. supervised encoder model in the first stage

21 Neural IR (2 points)

What are the main differences between BERT-based word embeddings and word2vec word embeddings?

- (1p) global representation vs. local representation
- (1p) average of multiple senses or most common representation vs. context-specific representation
- (1p) one word-one embedding vs. one word-multiple embeddings

22 Entity linking (3 points)



$$WLM(e,e') = 1 - \frac{\log(\max(|\mathcal{L}_e|,|\mathcal{L}_{e'}|)) - \log(|\mathcal{L}_e \cap \mathcal{L}_{e'}|)}{\log(|\mathcal{E}|) - \log(\min(|\mathcal{L}_e|,|\mathcal{L}_{e'}|))} \tag{5}$$

- $\mathcal{L}_e$ is the set of entities that link to e
- $|\mathcal{E}|$ is the total number of entities

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

(Use base 2 for log.)

WLM(e3, e6) = [-0.26]

23 Entity linking (2 points; -1 if incorrect)

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?**

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

24 Statistical significance testing (2 points)

Select all statements that are correct for Student’s t-test:

- [ ] Any test statistic can be used
- [X] The systems compared follow a normal distribution
- [ ] The test statistic is recorded for several permutations of the systems’ outputs

Scoring:

- 2 points if all correct
- 0.5 point if correct answer and one incorrect answer are selected
- 0 points otherwise

25 Retrieval evaluation (3 points)

Which of the following statements about creating assessment pools for retrieval systems is false?

- [ ] 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 documents not included in the assessed pool are assumed to be non-relevant
- [X] The assessors are presented with documents in the order in which they are retrieved by the system
- [ ] Greater pool depth ensures that more of the relevant documents are identified

Scoring:

- 3 points if only the correct answer is selected
- 1 point if the correct answer and one incorrect is selected
- 0 point otherwise

26 Retrieval (3 points)

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?

Learning-to-rank involves computing features that are based on the query, i.e., cannot be pre-computed (2p); therefore, scoring all document in a collection is infeasible (as in computationally too expensive) at retrieval time (1p).

27 Relevance feedback (2 points; -1 if incorrect)

Which of the following statements is *false*?

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