A3 Q3: What dataset to use?

- Q602 What dataset to use?
  - *Only IMDB dataset.* ***Follow the A3 handout****.*
  - *(the README.md says something different)*

A3 Q4.1 How to load the checkpoint

- Q4.1 requires loading the checkpoints. The checkpoints have GPU tensors. Some methods:
  - Option 1: Launch a Python script on GPU using a slurm job (srun or sbatch). Paste the results into LLM.ipynb.
  - Option 2: Run a jupyter server on teach.cs. Forward a port to visit it.
  - Option 3: Modify the classifier.py script. Save the checkpoint after moving the model to CPU.

A3 Q4.2 yes/no or positive/negative

- Piazza Q622, Q593, Q592: “This movie review is”. Query the tokens “yes”/”no” or “positive”/”negative”?
  - Should be ”positive/negative”. That is a typo in the handout.
  - but since the handout is already “yes/no”, **querying yes/no is accepted as well**.
  - We encourage you to try multiple prompt formats and see if the model outputs make sense.

UNIVERSITY OF TORONTO

Information Retrieval

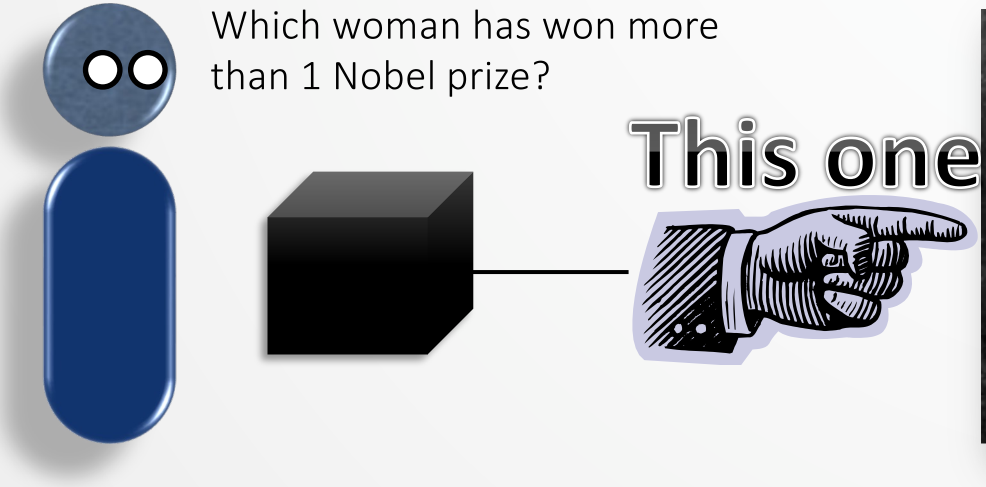
CSC401/2511 – Natural Language Computing – Winter 2023

Lecture 11

University of Toronto

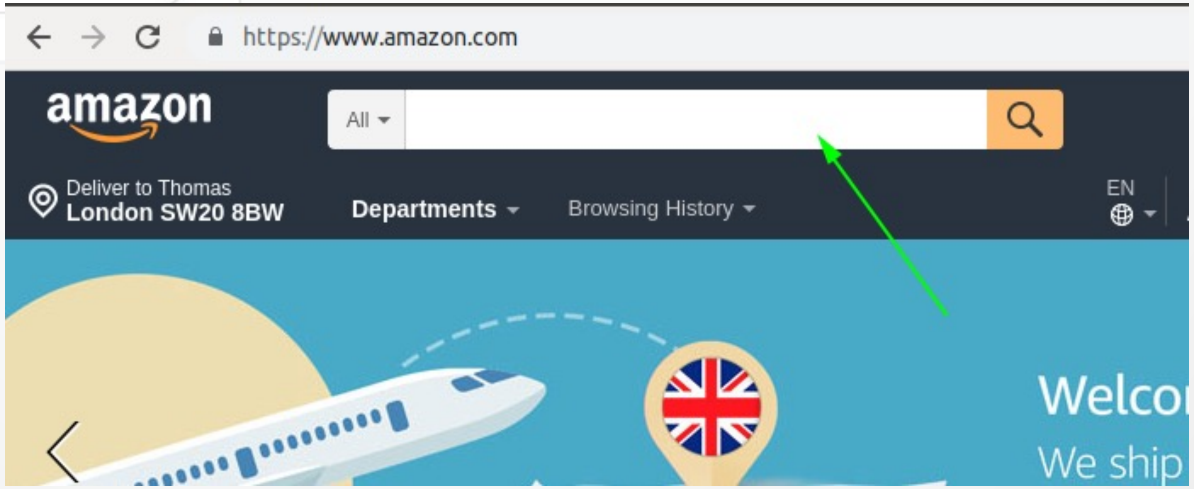
What is Information Retrieval?

Given a query, search for the most relevant document among a knowledge base.



(Marie Curie)

Search Engines are IR systems



Information Retrieval system

Given a query, search for the most relevant document among a knowledge base.

- Three key problems here:
  - How to represent the query?
  - How to store a knowledge base?
  - How to search efficiently and accurately?
- The problems are closely related. We will look at some popular approaches.

Scenario 1: SQL

- Structured Query Language (SQL) query
- How to represent the query?
  SQL queries.
- How to store a knowledge base?
  Tabular entries with predefined schemas.
- How to search efficiently and accurately?
  Compile and execute the SQL queries.

Scenario 2: Max-similarity search

- Find the document that is the most similar to the query.
- How to represent the query?

  Query is just another text-based document.
- How to store a knowledge base?

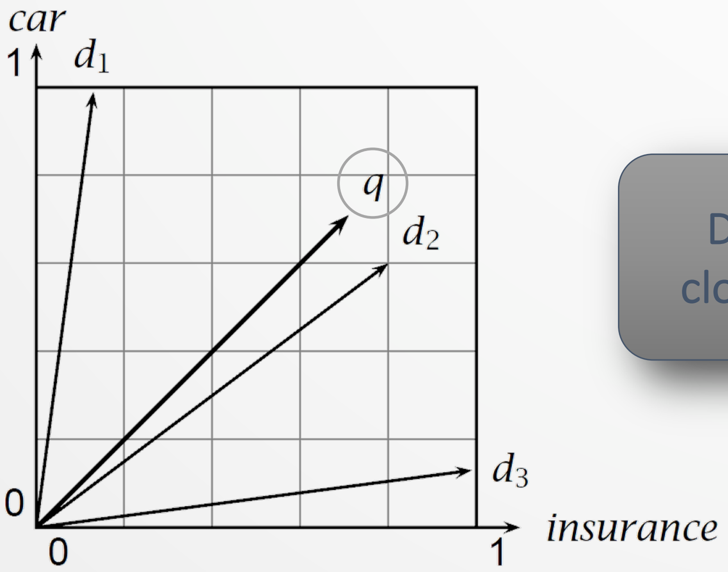
  Vectorized documents.
- How to search efficiently and accurately?

  Compute the **similarity score** between the query and each document. Return the document with the highest similarity score.

UNIVERSITY OF TORONTO

Similarity score

- If the query and the available documents can be represented by vectors, we can determine **similarity** according to their **cosine distance**.
  - Vectors that are **near** each other (within a certain **angular radius**) are considered relevant.



Document $\boldsymbol{d_2}$ is closest to query $\boldsymbol{q}$.

Vectorization: *tf.idf*

- *tf.idf* is a traditional method to vectorize the documents.
- It starts by weighting *words* in the *documents*.

  - **Term frequency, tf_{ij}:** number of occurrences of word w_i in document d_j.
  - **Document frequency, df_i:** number of documents in which w_i appears.
  - **Collection frequency, cf_i:** total occurrences of w_i in the collection.

Term frequency

- **Higher** values of $\boldsymbol{tf}_{ij}$ (for contentful words) suggest that word w_i is a **good** indicator of the content of document d_j.
  - When considering the relevance of a document d_j to a keyword w_i, tf_{ij} should be **maximized**.

- We often **dampen** tf_{ij} to temper these comparisons.
  - $tf_{dampen} = 1 + \log(tf)$, if $tf > 0$.

UNIVERSITY OF TORONTO

Document frequency

- The **document frequency**, df_i, is the number of documents in which w_i appears.
  - **Meaningful** words may occur repeatedly in a related document, but **functional** (or less meaningful) words may be distributed evenly over all documents.

Word	Collection frequency	Document frequency
kernel	10,440	3997
try	10,422	8760

  - E.g., *kernel* occurs about as often as *try* in total, but it occurs in fewer documents – it is a more **specific** concept.

Inverse document frequency

- Very specific words, w_i, would give **smaller** values of df_i.
- To maximize specificity, the **inverse document frequency** is

$$idf_i = \log\left(\frac{D}{df_i}\right)$$

where D is the total number of documents
and we scale with log (why? next slide)

- This measure gives **full** weight to words that occur in 1 document, and **zero** weight to words that occur in all documents.

UNIVERSITY OF TORONTO

Inverse document frequency

- The probability of a document containing word i is:

$$\frac{df_i}{D}$$

  "A document containing word i" is an event.

  Small p: this event is more surprising.

  Therefore, more information

- idf_i is the amount of information provided by observing the event.

UNIVERSITY OF TORONTO

tf.idf vectorization of a document

- We combine the **term frequency** and the **inverse document frequency** to give us a joint measure of **relatedness** between words and documents:

$$tf.idf\left(w_i, d_j\right) = \begin{cases} \left(1 + \log(tf_{ij})\right) \log \dfrac{D}{df_i} & \text{if } tf_{ij} \geq 1 \\ 0 & \text{if } tf_{ij} = 0 \end{cases}$$

- The j^{th}document is therefore represented by a vector:

$$[tf.idf\left(w_1, d_j\right),$$
$$tf.idf\left(w_2, d_j\right),$$
$$\ldots,$$
$$tf.idf(w_{|W|}, d_j)]$$

UNIVERSITY OF TORONTO

Aside: BM25

- BM25 is a baseline algorithm of IR.
- Given query $Q = [q_1, q_2, \dots, q_n]$, BM25 computes a similarity score for document d_j as:

$$\text{Score}(Q) = \sum_{i=1}^{n} \log \frac{D}{df_i} \times g\left(tf\left(q_i, d_j\right); k_1, b\right)$$

$g(\cdot)$ is an engineered function that has hyperparameters k_1 and b

No need to know the details of $g(\cdot)$

Scenario 3: Semantic Doc2Vec

- IR setting: Also using max-similarity search.
- The idea of word2vec can be applied as well.
- Goal: train a document encoder E.
- Design optimization goals for E so that:
  - If d_A and d_B are close to each other, then $\mathrm{sim}\langle E(d_A), E(d_B)\rangle$ should be large.
  - If d_A and d_B are far from each other, then $\mathrm{sim}\langle E(d_A), E(d_B)\rangle$ should be small.
- The definitions of closeness vary from algorithm to algorithm.

UNIVERSITY OF TORONTO

Semantic Doc2Vec

- Example: How does the Contriever paper define the closeness?
  - Positive samples d_+ for a document is **augmented** following some heuristics.
  - Negative samples d_- are **randomly sampled** from within the batches.

- A contrastive loss objective is:

$$L(q, d_+, d_-) = \frac{e^{\mathrm{sim}\langle E(q), E(d_+)\rangle/\tau}}{e^{\mathrm{sim}\langle E(q), E(d_+)\rangle/\tau} + \sum_i e^{\mathrm{sim}\langle E(q), E(d_-)\rangle/\tau}}$$

where τ is the temperature of the softmax.

Evaluating the retrieval systems

- Some commonly used metrics include:
  - Precision
  - Recall
  - F-score
  - Precision @ k

Precision and Recall

- **Precision**: $\frac{N_{\text{relevant \& retrieved}}}{N_{\text{retrieved}}}$
  - Among all retrieved documents, how many are relevant?
  - Precision in machine learning: $\frac{TP}{P}$
- **Recall**: $\frac{N_{\text{relevant \& retrieved}}}{N_{\text{relevant}}}$
  - Among all relevant documents, how many are retrieved?
  - Recall in machine learning: $\frac{TP}{T}$
- Note: Precision and recall has some tradeoff.

UNIVERSITY OF TORONTO

F-score

- **F-score** is the weighted harmonic mean of precision and recall:
  - $F = \frac{1}{\alpha\frac{1}{p}+(1-\alpha)\frac{1}{\mathrm{r}}}$
- Where p is precision, r is recall, and $\alpha \in [0,1]$.
- Notes:
  - When $\alpha = \frac{1}{2}$, we have $F_1 = \frac{2pr}{p+r}$
  - If either of precision or recall is 0 (i.e., true positive count $TP = 0$), then F is arbitrarily set to 0.

UNIVERSITY OF TORONTO

Precision at k

- Modern IR systems usually do not just give one result.
  - Even if the 1^{st} result is not relevant, the 2^{nd}, etc. results could be relevant too.
- People sometimes measure the **precision at k (P@k)**:
  - Among the top k results, how many of them are relevant?
- **P@k** has some potential problems:
  - The 1^{st}, 2^{nd}, …, k^{th} locations have no differences.
  - If there are less than k relevant results, then even the best system can't get **P@k**=1.

Lecture review questions

By the end of this lecture, you should be able to:

- Describe the procedure of max-similarity search.
- Describe the tf.idf vectorization.
- Describe a contrastive objective function of a semantic doc2vec method.
- Identify some evaluation metrics for IR systems and describe the trade-offs between these metrics.

Anonymous feedback form: https://forms.gle/W3i6AHaE4uRx2FAJA

UNIVERSITY OF TORONTO

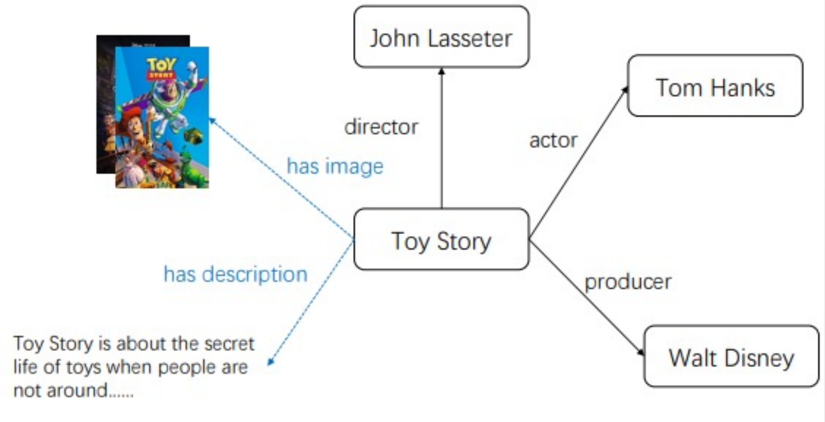
Appendix: Recent challenges of IR

- Structured, relational data
- Multi-modal data

Structured relational data

- Plain texts are **unstructured**.
- Many modern IR systems use **structured** data.
  - E.g., docs vectorized to the same dimensions.
  - E.g., relational data.
- Benefits & challenges of structured data.

{“name”: “Toy Story”,
“director”: “John Lasseter”,
… }



UNIVERSITY OF TORONTO

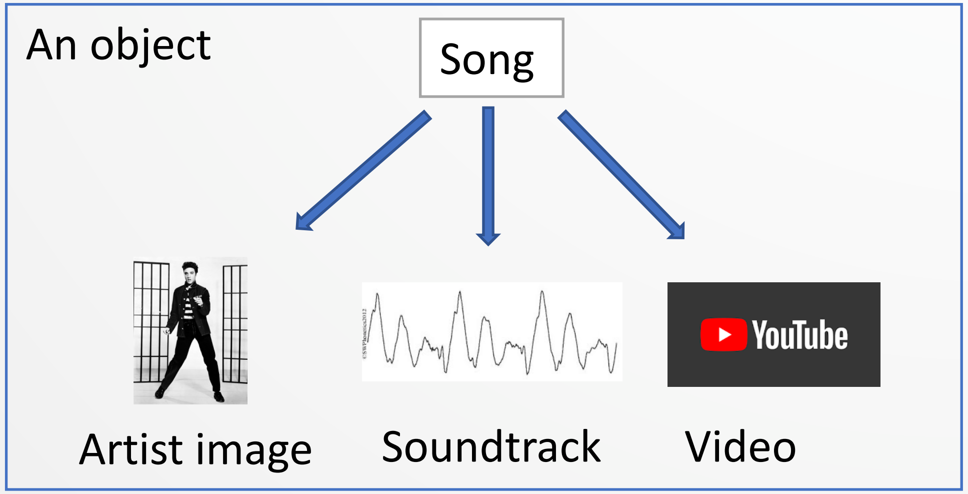
Storing structured data

- Saving each complex object as a database entry is one option.

- We can also store (or embed) the $\{R, S, T\}$ triplets.
  - R is the **relation** (e.g., "has-director") between:
  - the **source** S (e.g., "Toy Story") and
  - the **target** T (e.g., "John Lasseter")

Multimodal data

- Most modern IR systems are **multimodal**.
- The objects contain more than texts.
  - Images, sounds, even videos are stored too.
  - Choosing the right schemas is very important!



UNIVERSITY OF TORONTO