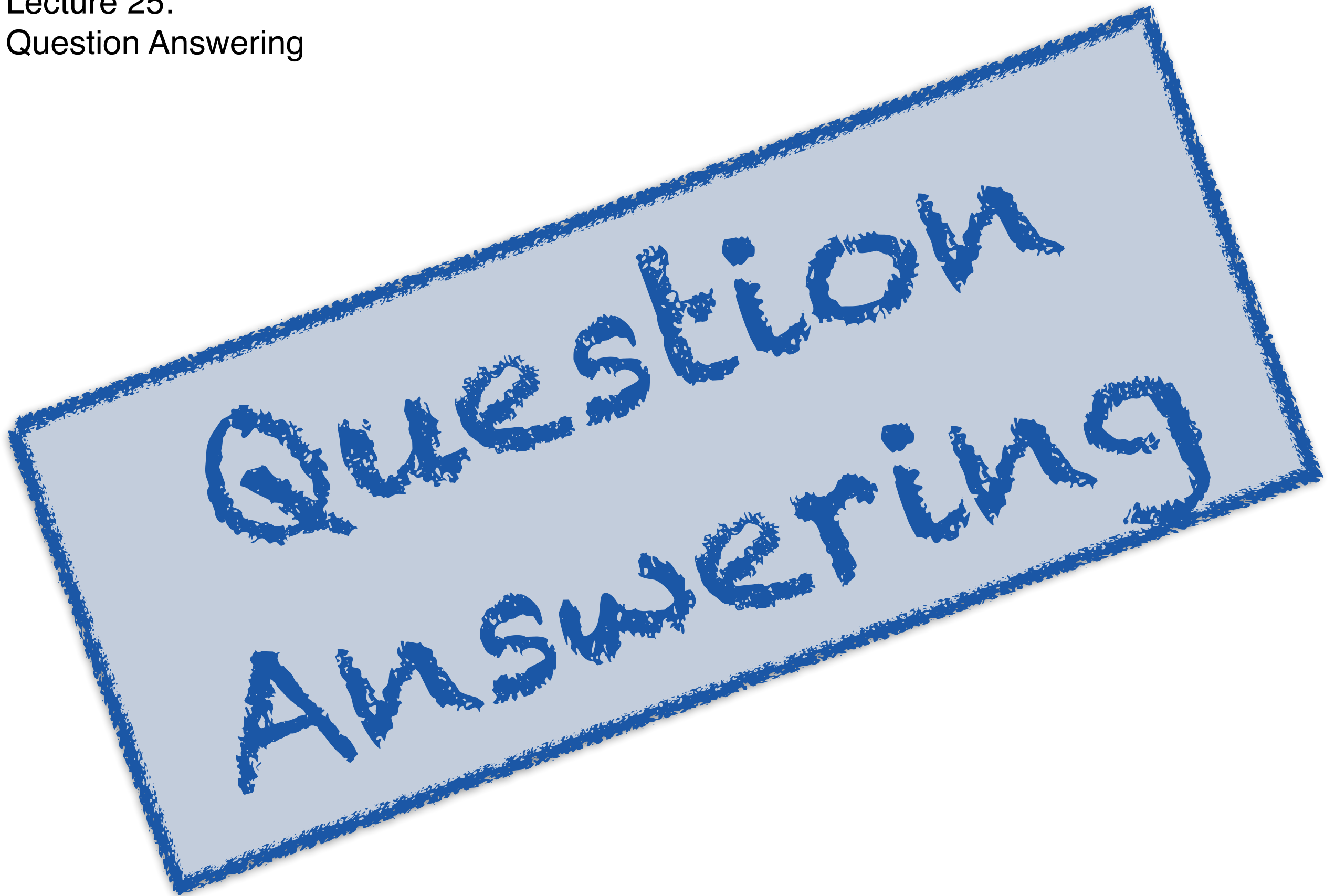


Lecture 25: Question Answering



Question Answering

Question Answering can mean different things:

- Being able to query a knowledge base (e.g. a database of known facts) in natural language. This may require a semantic parser to translate the natural language question into, say, SQL
- Being able to query a collection of documents that is known (or assumed) to contain answers (as short text spans in these documents)
- Being able to answer questions about a single document by returning short text spans in the document that answer these questions (“reading comprehension”)
- Being able to answer knowledge questions about a domain (e.g. take multiple choice exams on science questions)

Reading: Chapter 25



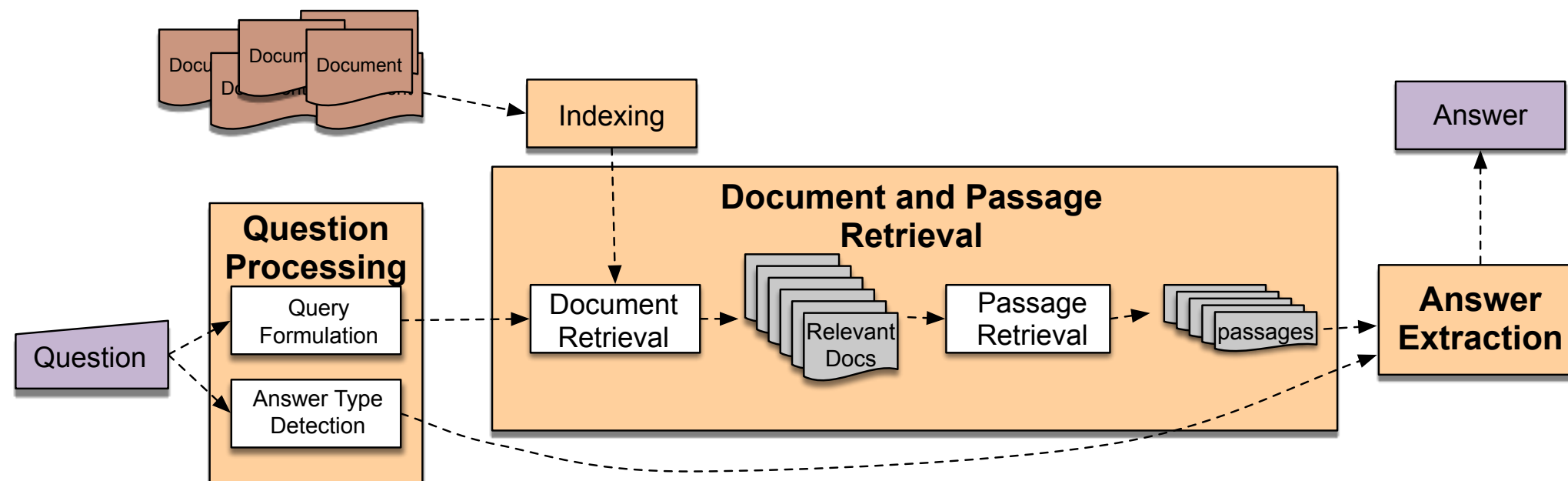
Question Answering (QA) as an Information Retrieval (IR) task

Answer a user's questions by finding a text snippet in a large document collection that contains the answer.

Questions that can be answered in this way are typically about simple “factoids”

Question	Answer
Where is the Louvre Museum located?	in Paris, France
What's the abbreviation for limited partnership?	L.P.
What are the names of Odin's ravens?	Huginn and Muninn
What currency is used in China?	the yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	drums
What's the official language of Algeria?	Arabic
How many pounds are there in a stone?	14

The IR-QA pipeline



Stage 1: Question Processing

- Query Formulation
- Answer Type Detection

Stage 2: Document and Passage Retrieval

- Document Retrieval
- Passage Retrieval

Stage 3: Answer Extraction

Question Processing

We need to get from a **natural language question...**

Which US state capital has the largest population?

...to a **query string** for the IR system:

Query = "US state capital has largest population"

... an **answer type**:

Answer Type = CITY

... and the **focus** (which words in the question are likely to be replaced by the answer):

Focus = "which US state capital"



Answer Type Identification

The answers to many common factoid questions fall into a small number of categories (**answer types**). Knowing the answer type can be very helpful.

In the simplest case, the **question word** alone is sufficient to identify the answer type:

Who... → PERSON

Where... → LOCATION

When... → TIME

But in many cases, one has to consider at least the first noun after the question word, or the verb

Which city... → CITY

How much does ... cost → MONEY

Answer types (Li & Roth '02,'05)

Entities:

animals, body parts, colors, creative works (books, films,...),
currency, diseases/medicine, products,...

Humans:

individuals (who was the first person on the moon?),
descriptions (who was Confucius?), groups, etc.

Locations:

city, country, mountain, state, ...

Descriptions:

definitions (what is X?), manner (how can you do X),...

Numeric:

code (e.g. phone numbers), counts, dates, distances, sizes,
order (ranks of entities), temperatures, speeds, weights, ...



Document and Passage Retrieval

The IR engine returns a ranked list of relevant **documents** from the collection.

Because answers are short snippets, the top n documents are split into shorter **passages** (e.g. paragraphs).

We can filter passages to identify more relevant passages at this stage, e.g. based on how many named entities they contain, how many question words (or n-grams) they contain, the answer type, etc.



Answer Extraction

Given a set of relevant passages, return the span that contains the answer.

Baseline model (for some types of questions)

Run an NER system, and return the entities whose type matches the answer type

More generally, answer extraction can be treated as a **sequence labeling task**

Evaluation: MRR

The **mean reciprocal rank (MRR)** metric is used to evaluate system that return a ranked list of items (here: answer spans):

Q: Where was Elvis born?

Answers:

1. Memphis, Tennessee
2. Tupelo, MS ← **Correct** (rank(Q) = 2)
3. Graceland

Define **rank(Q)** as the highest rank of any correct answer for Q, and **rRank(Q)=1/rank(Q)** when at least one correct answer is returned, and **rRank(Q)=0** when no correct answer is returned

The system's MRR score on a pool of N questions is then defined as the average (mean) reciprocal rank on all questions

$$\text{MRR} = \frac{1}{N} \text{rRank}(Q_i)$$



Reading comprehension as span-extraction QA

Reading comprehension tests often ask children to answer questions based on a short paragraph.

Although reading comprehension can be formulated as a multiple-choice task, or a free answer task (which is difficult to evaluate), the span-extraction perspective requires that answers correspond to text spans



SQuAD

Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, *Dangerously in Love* (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Q: "In what city and state did Beyoncé grow up?"

A: "Houston, Texas"

Q: "What areas did Beyoncé compete in when she was growing up?"

A: "singing and dancing"

Q: "When did Beyoncé release *Dangerously in Love*?"

A: "2003"

Humans were asked to write questions for Wikipedia paragraphs and provide spans as answers.

The best systems outperform humans, even on SQuAd 2.0, which has "unanswerable questions" (no span to be returned)

Leaderboard: <https://rajpurkar.github.io/SQuAD-explorer/>



A BiLSTM-based QA System

Basic architecture: Two biLSTMs (for question and passage):

- The **question LSTM** computes a single question **vector** \mathbf{q}
- The **passage LSTM** predicts **start and end positions of the answer span**, based on two learned classifiers that depend on each passage word's embedding \mathbf{p}_i and on the question vector \mathbf{q}

$$P_{\text{start}}(i) \propto \exp(\mathbf{p}_i \mathbf{W}_s \mathbf{q}_i) \quad P_{\text{end}}(i) \propto \exp(\mathbf{p}_i \mathbf{W}_e \mathbf{q}_i)$$

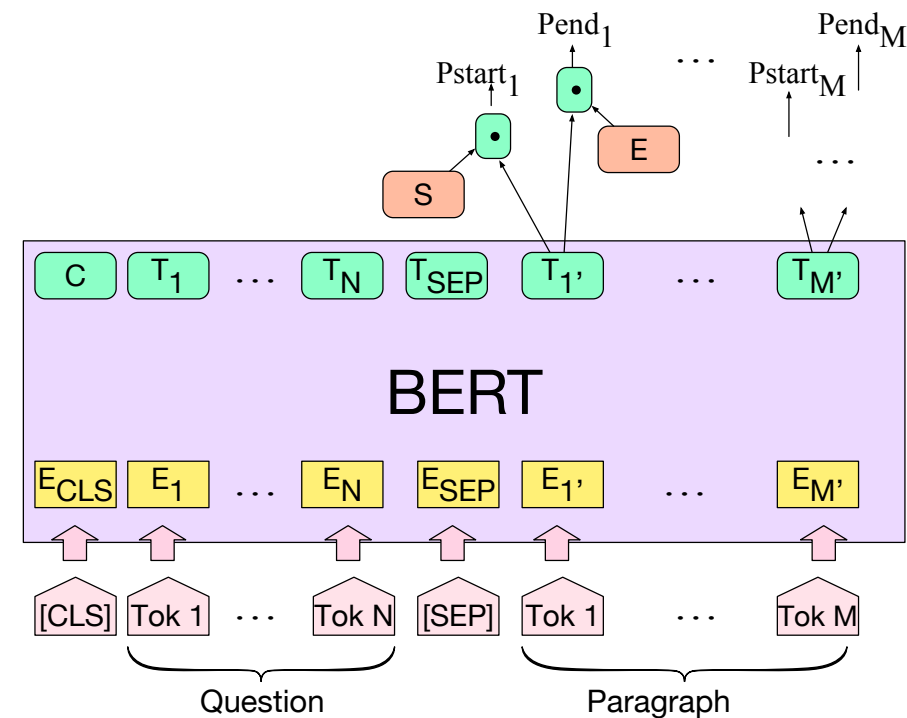
The **question vector** \mathbf{q} is a weighted average of the biLSTM-based embeddings of the question words: $\mathbf{q} = \sum_j b_j \mathbf{q}_j$

Question word weights b_j are given by the normalized, exponentiated dot product of each word embedding with a single, learned, relevance weight vector \mathbf{w}

$$b_j = \exp(\mathbf{w} \cdot \mathbf{q}_j) / \sum_i \exp(\mathbf{w} \cdot \mathbf{q}_i)$$

Passage: Each token is input as an embedding (e.g. GloVe), concatenated with its POS tag/NER label, a 0/1 flag indicating whether it occurs in the question, and possibly an token-specific attention-based embedding of the question

A BERT-based QA system



BERT is a very large pre-trained transformer-based model that provides contextual embeddings

BERT reads the question and passage as a single string, separated by a SEP token.
(this is standard for tasks where BERT has to consider two sequences)

To use BERT for QA:

- define new start and end token embeddings S and E
- fine-tune the output layer, again to predict start and end, e.g. via $P_{\text{start}}(i) \propto \exp(\mathbf{p}_i S)$ $P_{\text{end}}(i) \propto \exp(\mathbf{p}_i E)$

More recent developments in QA

QA is a very active area of research

Retrieval-based QA is often seen as too simplistic (especially when billed as “reading comprehension”)

More recent developments include datasets whose answers require several steps of reasoning (multi-hop QA), as well answers that require commonsense knowledge.

Visual QA: answer questions about an image.



Science exams as testbed for QA

Task: Answer **multiple choice questions** from 8th-grade science exams

1. Which equipment will best separate a mixture of iron filings and black pepper?

(1) magnet (2) filter paper (3) triple-beam balance (4) voltmeter

This requires a lot of **background knowledge** that has to be acquired from somewhere (e.g. textbooks), and reasoning capabilities

https://allenai.org/content/docs/Aristo_Milestone.pdf

