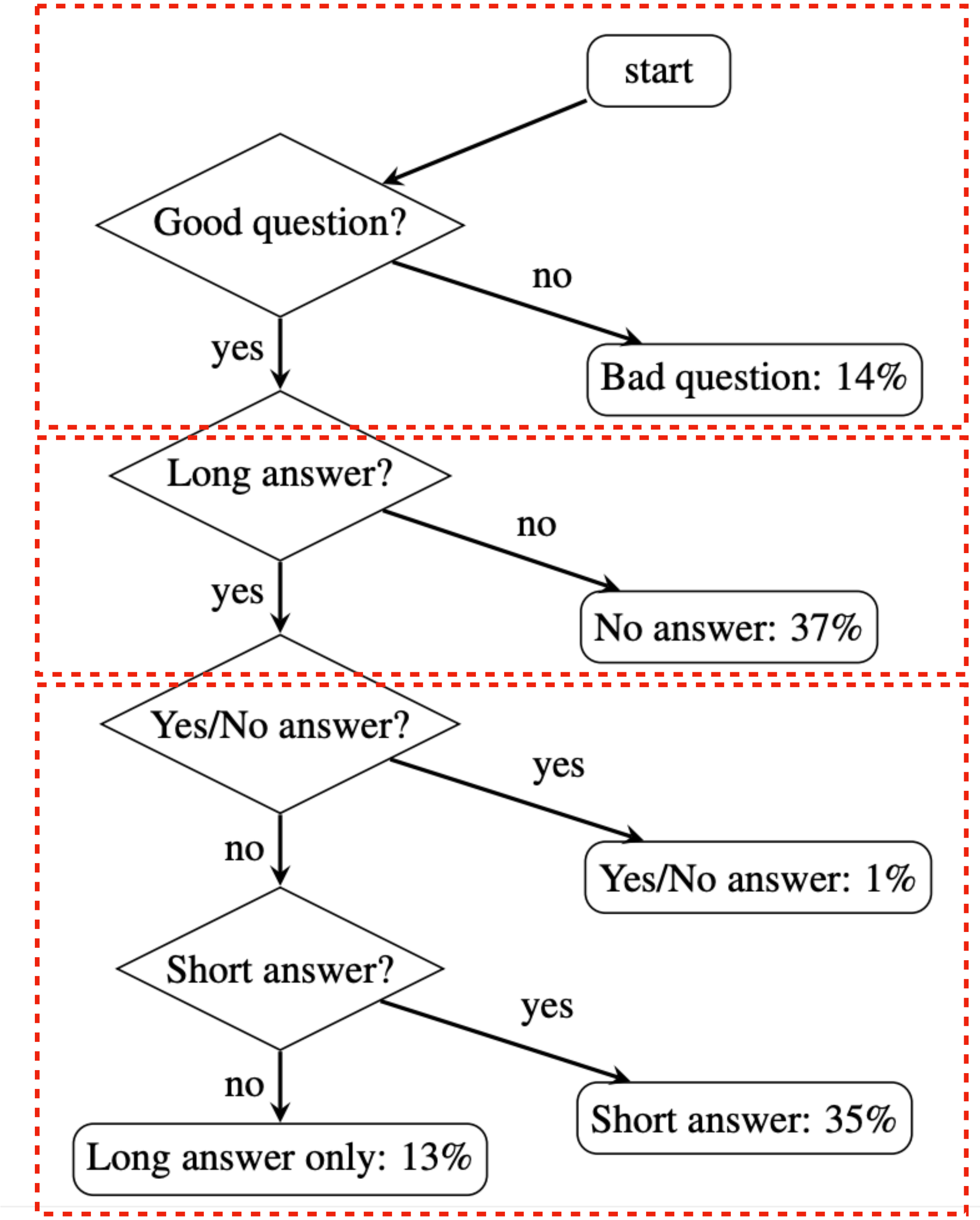
Kaggle Competition: NQA 2020/1/7 Discussion

Presenter: Lin
Team Member: Ye and Lee.

Coverage outline

- Introduction to NQA
    - Issue discussion
    - What to do for the issues?
- Architecture prototyping
    - If the architecture solve the issues?
- Other technical issues
- Milestone/Timeline Discussion

Introduction to NQA collection process

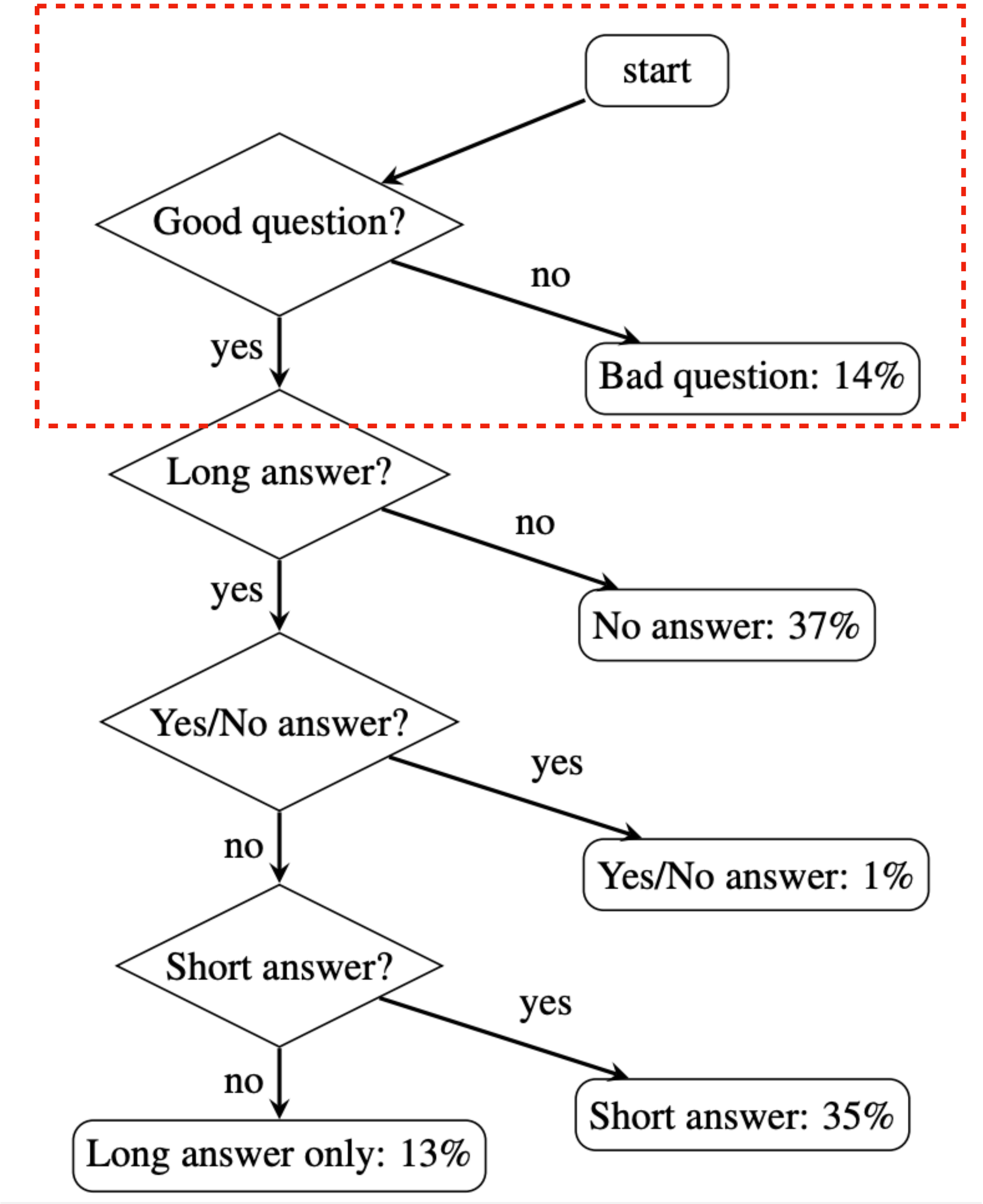


Question Identification

Long Answer Identification

Short Answer Identification

Question Identification



What is bad question?

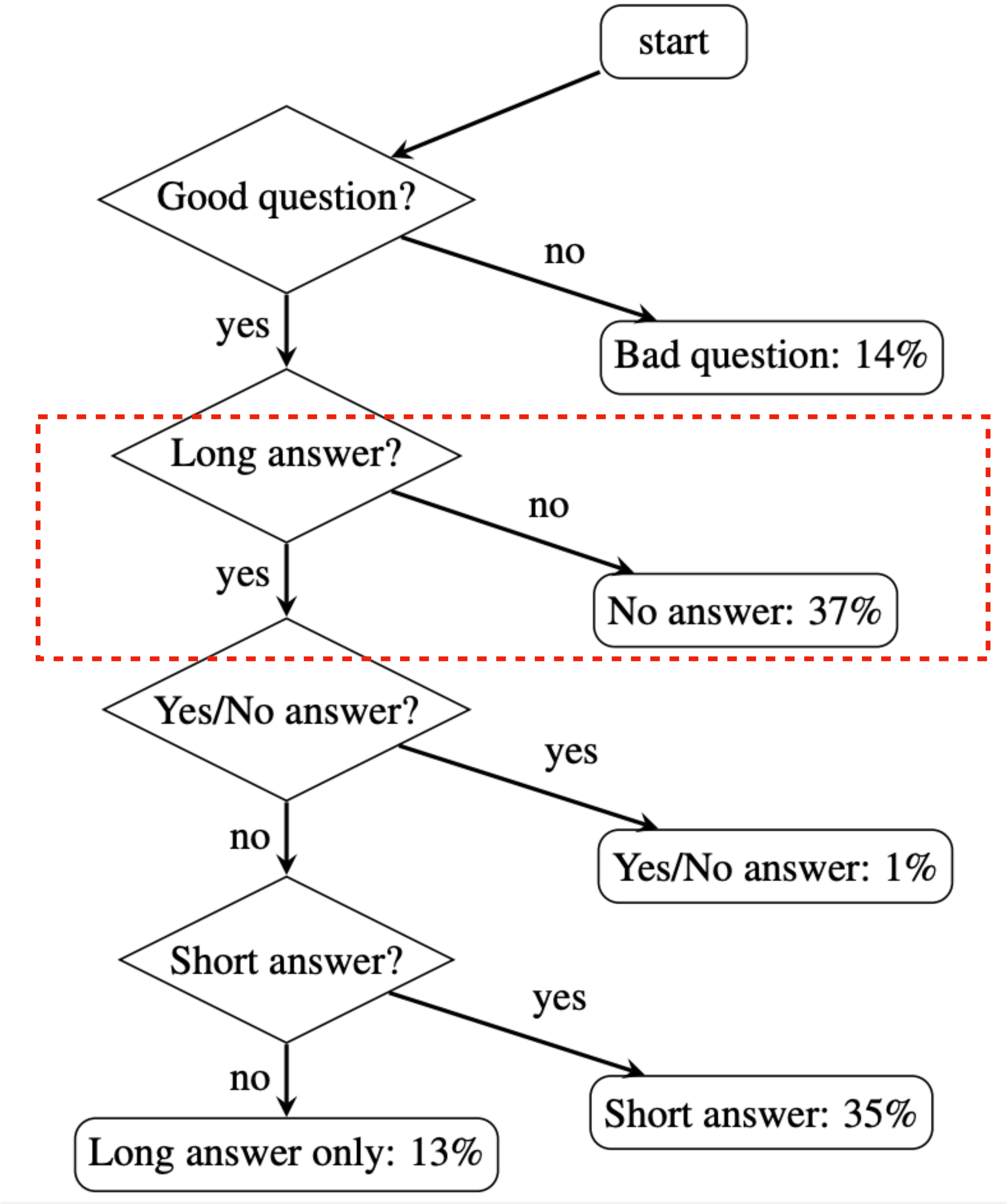
- **Ambitious**
- **Incomprehensible**
- **Depend on false presumption**
- **Opinion seeking**

Issue: Bad question is identify but not removed

Q: Can we identify bad questions?

- **possibly they are mixed with those without answers.**

Long answer identification



What is long answer?

- **The shortest html box wherein answer can be found.**

Criteria of no answer:

- **No answer in each html box**
- **Answers appear in multiple boxes**

Occurrence in the dataset:

- **= 1%+35%+13% = 49%**

Issue 1: Examples without long and short answers

- Multiple possibility:
  - 1. Can be a bad question
  - 2. Good question, but long answer not found
  - 3. Good question and long answers found, but they occur in multiple boxes.

What to do for Issue 1?

- What to do given observed dataset?
  - Use question to provide information of good or bad.
  - Count the number of long answer candidates (box) with answer (Short answer!=Null or Answer can be inferred but too long for short answer).
    - If > *1*, then long answer = *Null*.
    - If = 0, then long answer = *Null*.

Issue 2: Various long answer types

- Issue 2.1: Different text structures for different types.
- Issue 2.2: Long answer type imbalance:
  - Paragraphs: 73%
  - Tables: 19%
  - List items 3%
  - Table rows: 1%
  - …

What to do for Issue 2?

- Issue 2.1: Different text structures for different types
  - 1. Build different sequence encoder(s) for each type, and combine them with a type classifier
    - First, distinguish paragraphs and tables
    - Then, distinguish paragraphs, tables, and table rows.
    - etc…
  - 2. Build a token-based unify encoder, with html token guiding the encoding process
- Issue 2.2: Long answer type imbalance
  - Allow higher weighting for more difficult type for the training of type classifier or the token-based unify encoder (need to do error analysis before that)

Issue 3: long-answer box nested

- A long-answer candidate can be a child of another parent candidate.

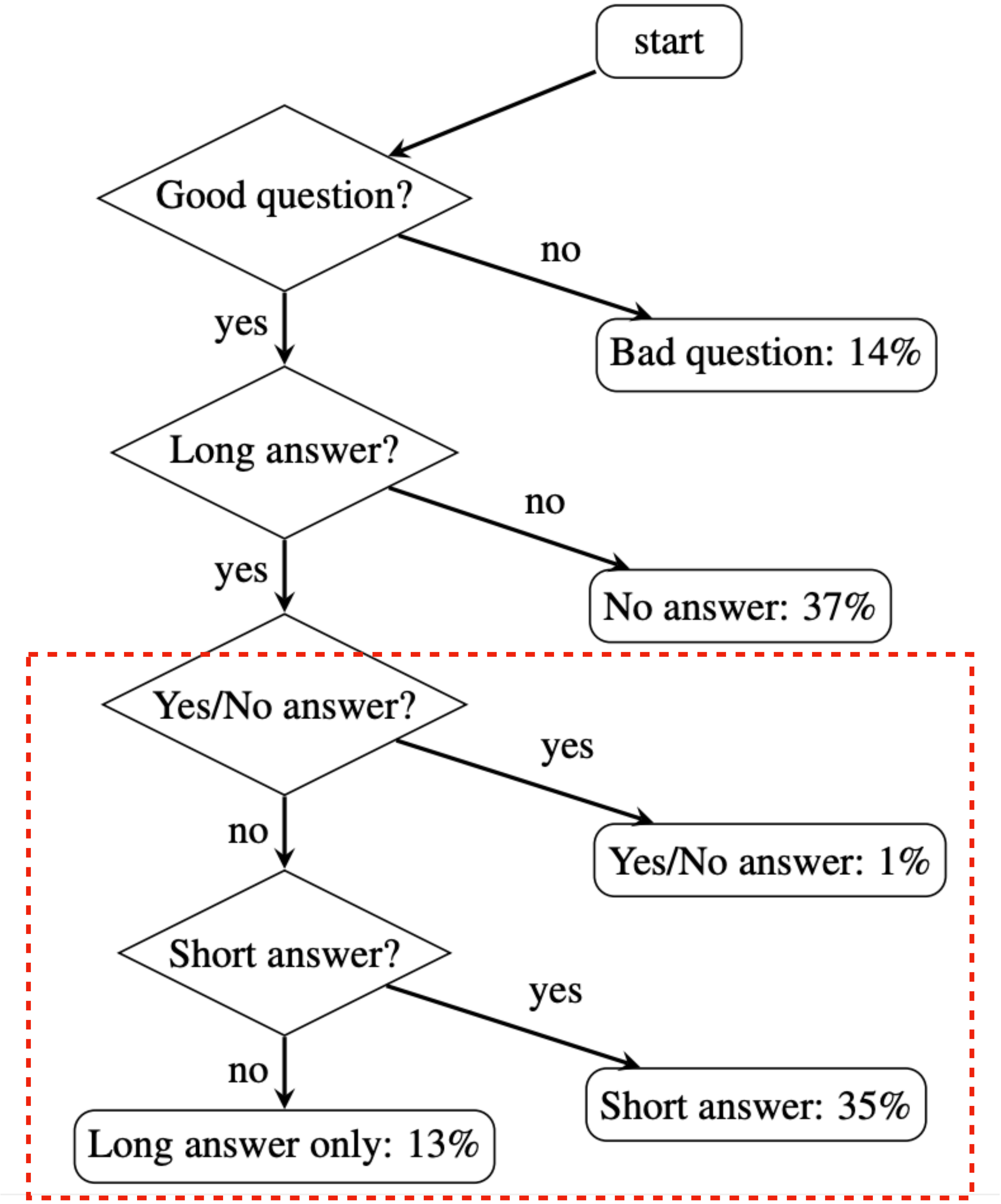
- According to the paper, select the shortest (lowest-level) box.

- Box level is provided in the training data.

What to do for Issue 3?

- 1. Using box-dependency feature:
    - 1. Use top_level as feature.
    - 2. If (1.1) not comprehensive enough, customize the box-dependency feature.
- 2. Need both locally and global identification of long answer
    - *Local*: Given a candidate, predict if a candidate (box) contains the answer.
        - Using only examples of non-Null long answer in training*:*
            - *Positive*: candidate marked as long answer.
            - *Negative:* candidates that is not long answer.
    - *Global*: Given all the candidates, return one candidate as long-answer or return *Null*
        - Features:
            - Box-dependency features (1) of the candidates.
            - Local identification results of the candidates.
            - Question, of course.

Short answer identification



What is short answer?

- **Yes or no**
- **Answer is in the entities in the long answer text**

Criteria of no answer:

- **Bad question or No long answer (i.e., long answer = *Null*)**
- **Answer can be inferred but too long for short answer**

Issue 4: For *Null* short answers

- Issue 4.1: When long answer = *Null*, short answer always = *Null*; however, the reason may be there are multiple long answer boxes (i.e., short answer exists for multiple candidates).

- Issue 4.2: When long answer != *Null*, short answer can = *Null*. =>

    - Answer exists in one long answer candidate X=> short answer exists.

What to do for Issue 4?

- Issue 4.1: When training the short-answer identification, remove the cases where {long answer = Null & short answer = Null}

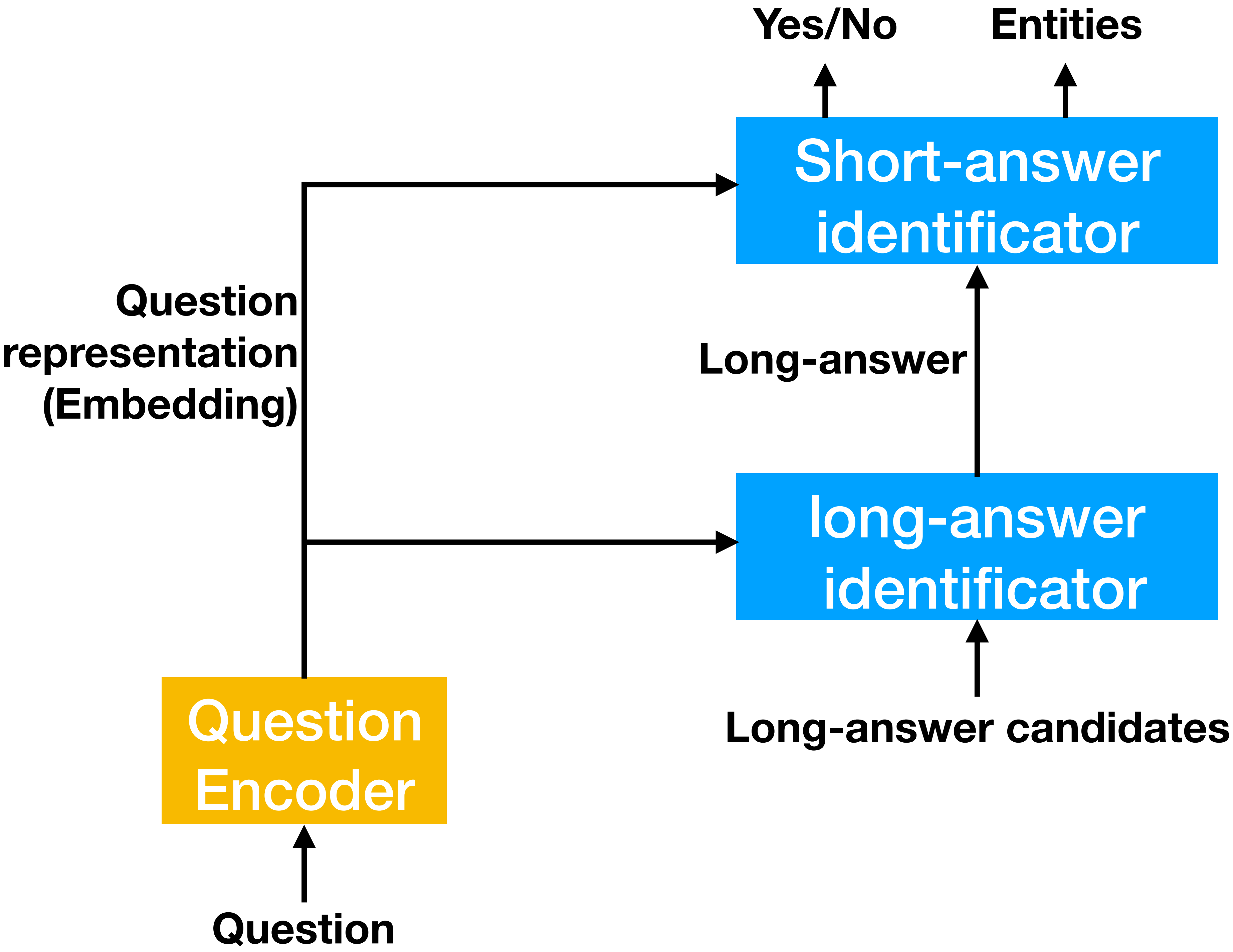
- Issue 4.2: Separate the modeling of short answer identification and long answer identification. (Later discuss model fusion.)

Issue Outline

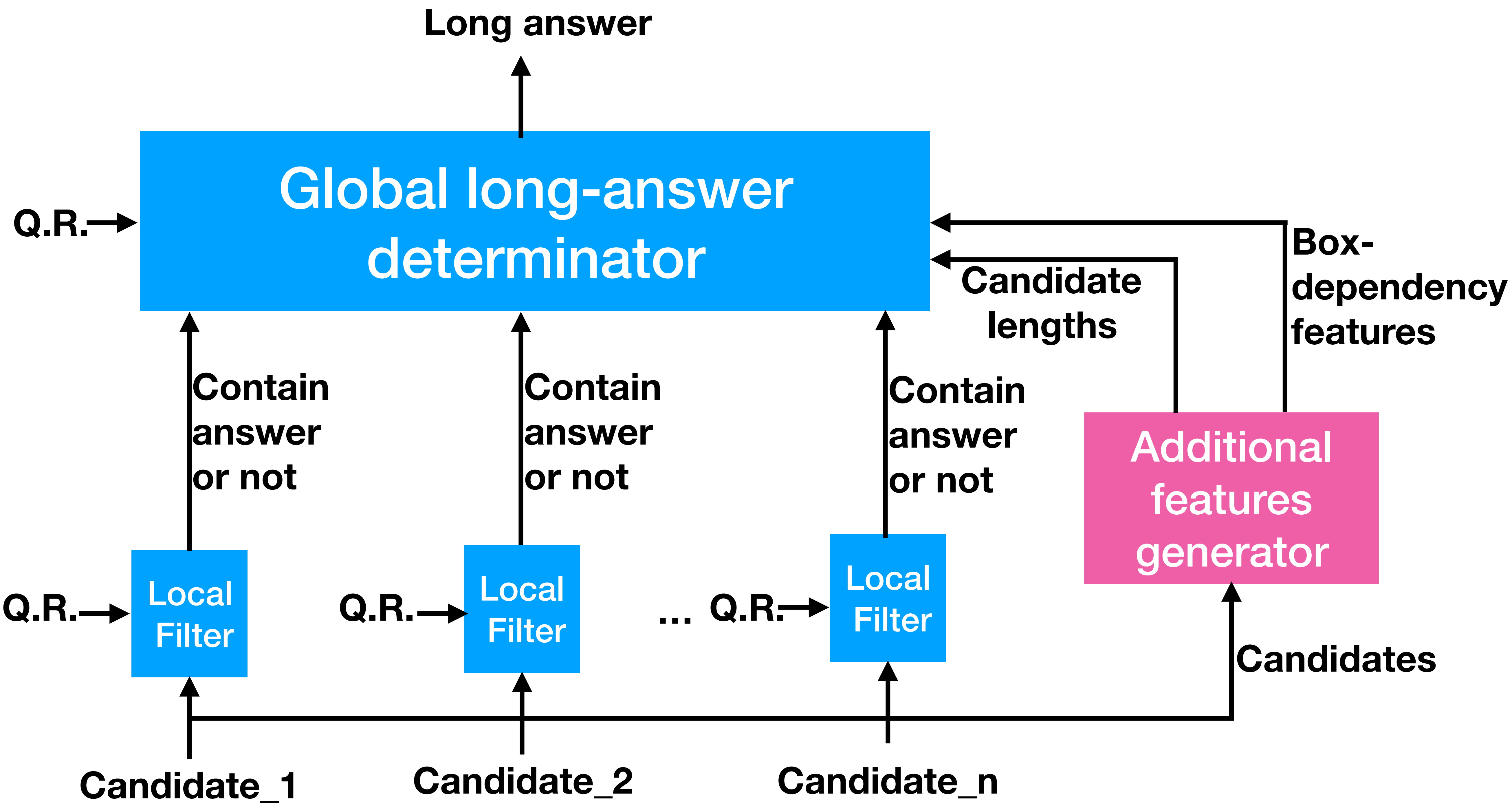
- 1. Examples without long and short answers have multiple possibility
- 2.1/2.2 Various long answer types
- 3. Nested long-answer boxes
- 4.1/4.2 *Null* short answers
- …

Take away: When there is categorization, there is imbalance.

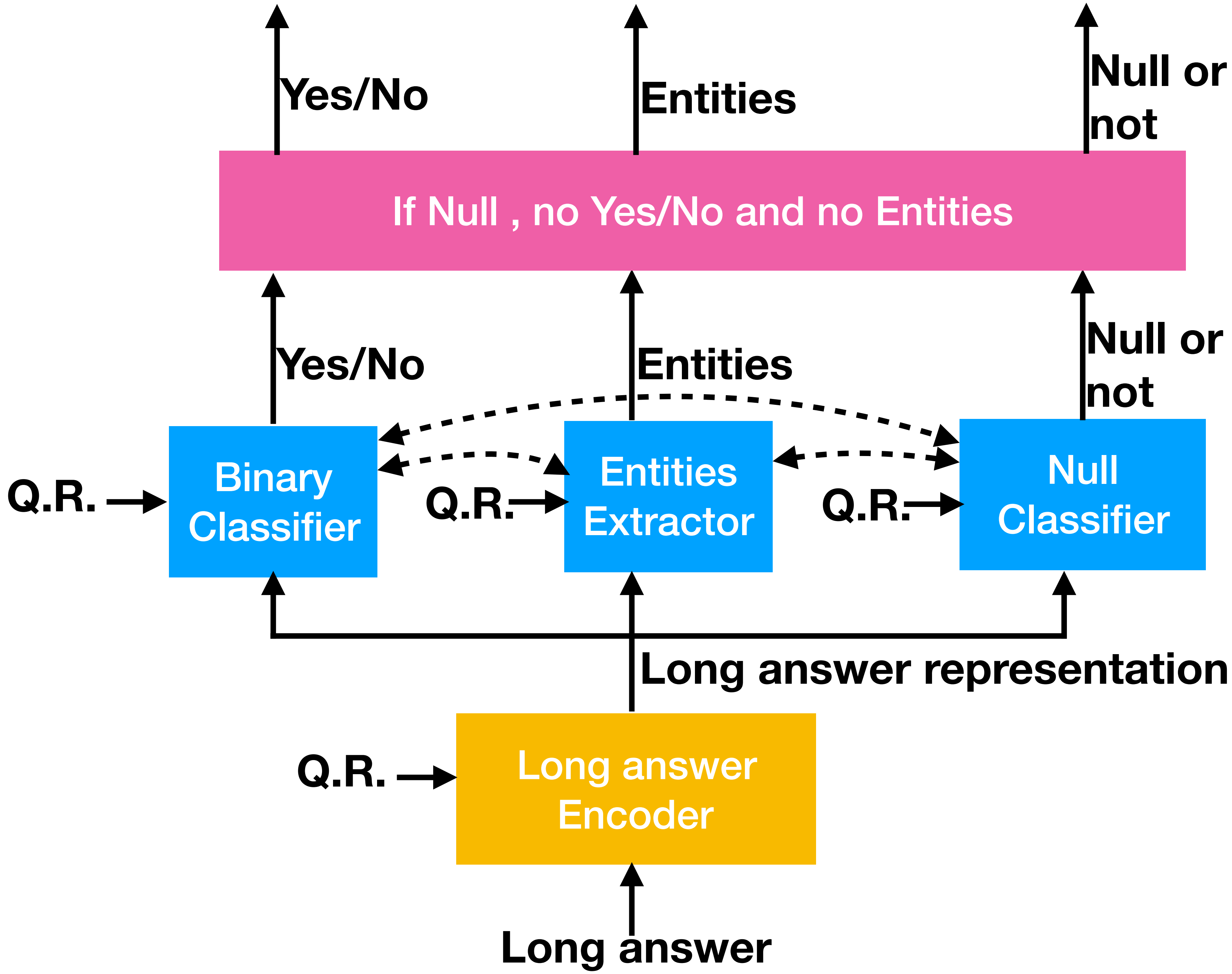
Architecture Prototyping


Yes/No
Entities
Short-answer identificator
Question representation (Embedding)
Long-answer
long-answer identificator
Question Encoder
Long-answer candidates
Question

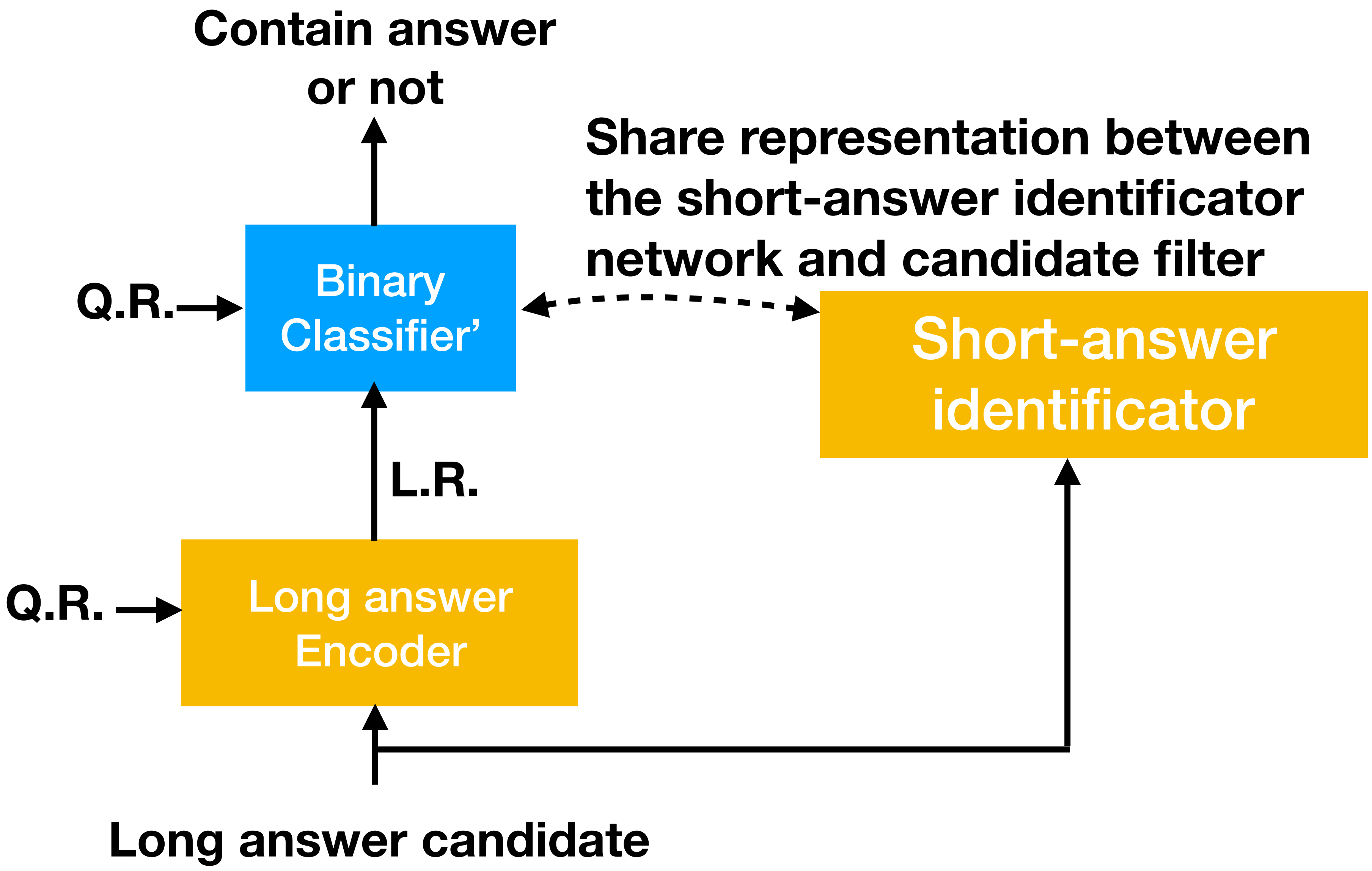
Long-answer Identificator



Short-answer Identificator



Candidate Filter (local filter)


Contain answer or not
Share representation between the short-answer identificator network and candidate filter
Q.R.
Binary Classifier’
Short-answer identificator
L.R.
Q.R.
Long answer Encoder
Long answer candidate

Analysis of Issues (cont.)

- ***Issue 1. Examples without long and short answers have multiple possibility***
  - ***=> May be solved by taking Q.R. as input and taking multiple candidates in the global long-answer determinator***
- Issues 2.1/2.2 Various long answer types
- ***Issue 3. Nested long-answer boxes***
  - ***=> May be solved by taking box-dependent features as input***
- Issues 4.1/4.2 *Null* short answers
  - ***=> Issue 4.2 may be solved by separate the modeling of short/long answer identification***

Solutions to the remaining issues

- Issues 2.1: Make the long-answer encoder token-based, with html token guiding the encoding process

- Issues 2.2: Allow higher weighting for more difficult type for the training of type classifier or the token-based unify encoder

- Issues 4.1: When training the short-answer identification, remove the cases where {long answer = Null & short answer = Null}

More specific design issues

- How to design the entity extractor in the short-answer identification process?
- How to design the Question/Long-answer encoder?
  - Check Bert, Attention mechanism, …
- How to share representation information between two networks?
  - Using memory augmentation
- How to design the classifiers?
- How to design the global long-answer identificator
- …

Other technical issues

- Evaluation metrics implementation
    - F1-score / Precision / Recall for short and long answer
- Should we try the validation and test dataset released by Google?
    - Support more robust validation and testing scores.
- How to share code / analysis notebooks?
- …

Conclusion

- We have proposed an architecture prototype for our competition tasks and discussed the issues related to the architecture.
- It raises some distinct design issues.
- Other technical issues still remain.

Discussion of Milestone and Timeline

- Jeffrey
- Cure
- Dexter

Thanks for your attention!
And may the reward be with us!