SIC Saarland Informatics Campus

UNIVERSITÄT DES SAARLANDES

mpi max planck institut informatik

Question Answering Systems

Templates: From text to curated KGs

Rishiraj Saha Roy
Max Planck Institute for Informatics, Germany

2

Starts at 14:15

Mic → Mute

Video → off

Saarland University, Summer Semester 2020

Logistics: Quick walkthrough

Logistics: Recap

- New registrations are closed
  - 94 members as of now, competitive
- Online lectures for entire semester
- Save Internet traffic
  - Please mute your microphones
  - Turn off videos
- Recommended: Read papers before class
- All questions primarily via chat
  - To be answered in two ~15 min slots: immediately after break, immediately after class

- Use headset if you want to speak
- Connect to LAN if possible instead of WiFi
- Switch off other Internet-hogging stuff during class
- Use laptop/desktop instead of mobile
- Slides and recordings after lecture
- Material for first lecture on Google Groups

Logistics: Course

- Research course, no textbook
- Roughly two (long) research papers discussed in lectures every week
  - Influential work in community
  - Work from our group: qa.mpi-inf.mpg.de
- One written assignment per week (more on next slide)
- One toy coding assignment at the end
- Overlaps with IRDM last lecture ☺
- Teaching assistant: Magdalena Kaiser

Assignment

- 10 sentences on each paper
  - 4-sentence summary
  - 3 positives (one sentence each)
  - 3 negatives (one sentence each)
- 10 sentences x 2 papers = 20 ± 5 sentences per week

plaintext
- In text file: 2020-qa-<your-name>-week-01.txt as attachment

02

03

- Email to me and Magdalena with subject
  - 2020 QA Assignment Week 01
  - 2020 QA Assignment Week 02
  - 2020 QA Assignment Week 03
  - ...
- Deadline: Before start of next class

For this week, extension of 1 week

No further extensions

Assignment (2)

- Contribute to final grade: Excellent, Good, Fair, Poor
- No right or wrong answers, or *appropriate length*
- Goal is to reduce arbitrariness in final exam
- No presentation of assignments necessary
- Clear cases of plagiarism will result in de-registration
- Estimate of hours per week
  - 2 hours lecture
  - 2 x ~2 hours reading two papers and commenting
  - 6 hours in all ~ 6 ECTS ☺

Exam

- Exams are online, individual, via Zoom
- 1 oral main exam
  - Tuesday 21 July 2020
  - 10 – 15 minutes per person
  - Depends on number of students
- 1 oral re-exam
  - Tuesday 04 August 2020
- No re-re-exam ☺

- Lecture plan on course website
- Background: IR, NLP, DM, ML, …

Question of the day

we

How can automatically learn templates for question answering?

You'll find this covered in

- Never-Ending Learning for Open-Domain Question Answering over Knowledge Bases
  - Abujabal et al.
  - The Web Conference 2018
  - https://myahya.org/publications/neqa-abujabal-www2018.pdf
- Learning Surface Text Patterns for a Question Answering System
  - Ravichandran and Hovy
  - ACL 2002
  - https://www.aclweb.org/anthology/P02-1006.pdf

Research paper 1

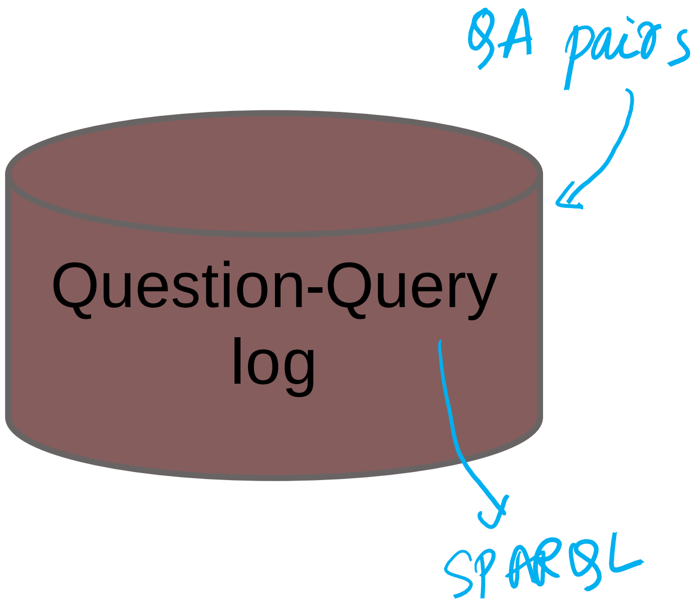
Never-Ending Learning for Open-Domain Question Answering over Knowledge Bases

Tackling diversity: Our 5-point agenda

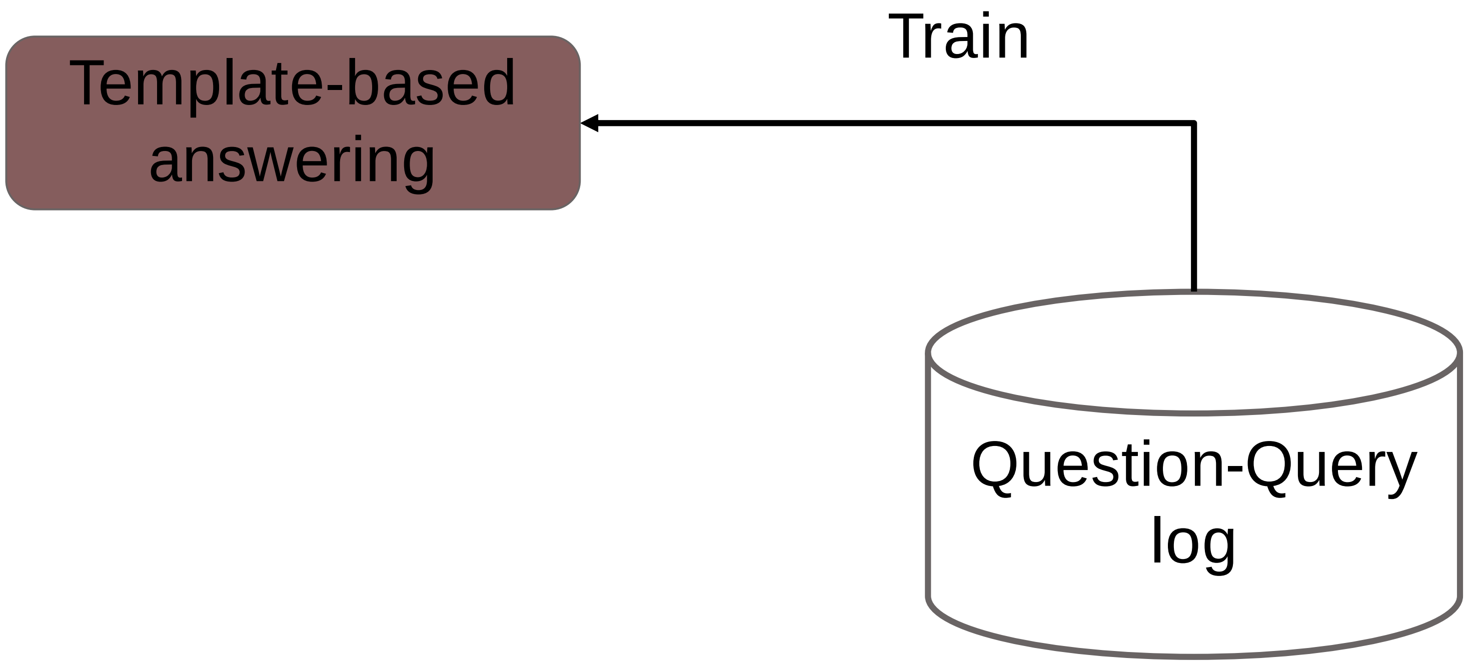
1. Start small with seed training data
2. Learn syntactic templates on-the-fly
3. Extend coverage with semantic similarity
4. User feedback to prevent erroneous learning
5. Assimilate into continuous learning framework

NEQA: Never-ending Learning for Question Answering

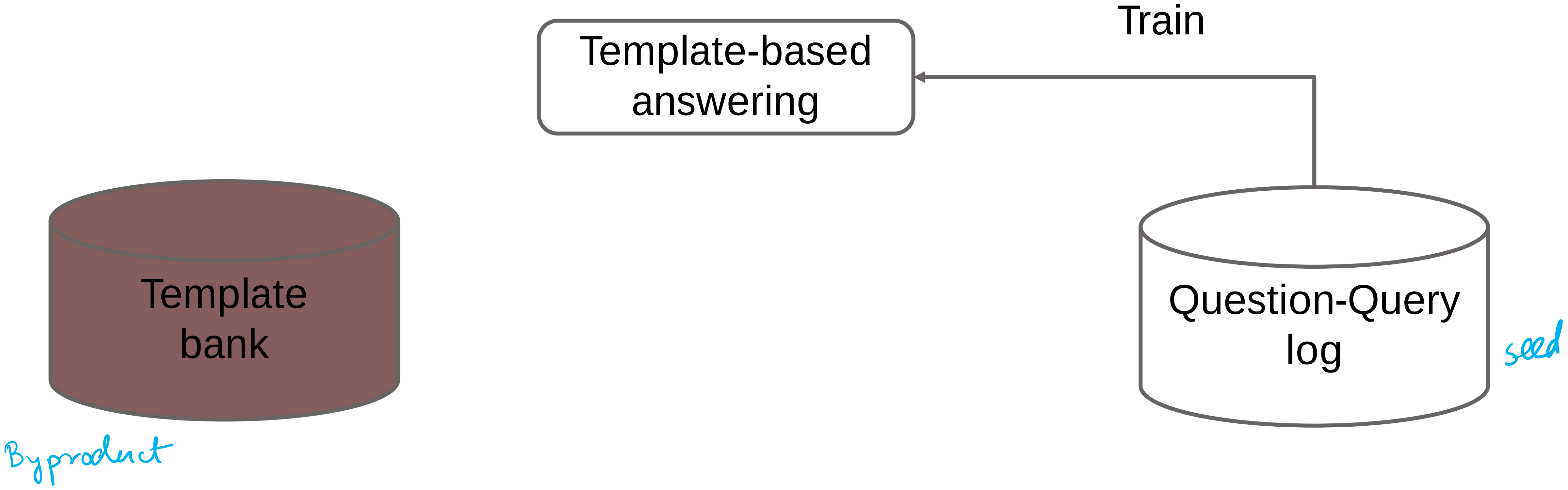
Never-ending learning with NEQA



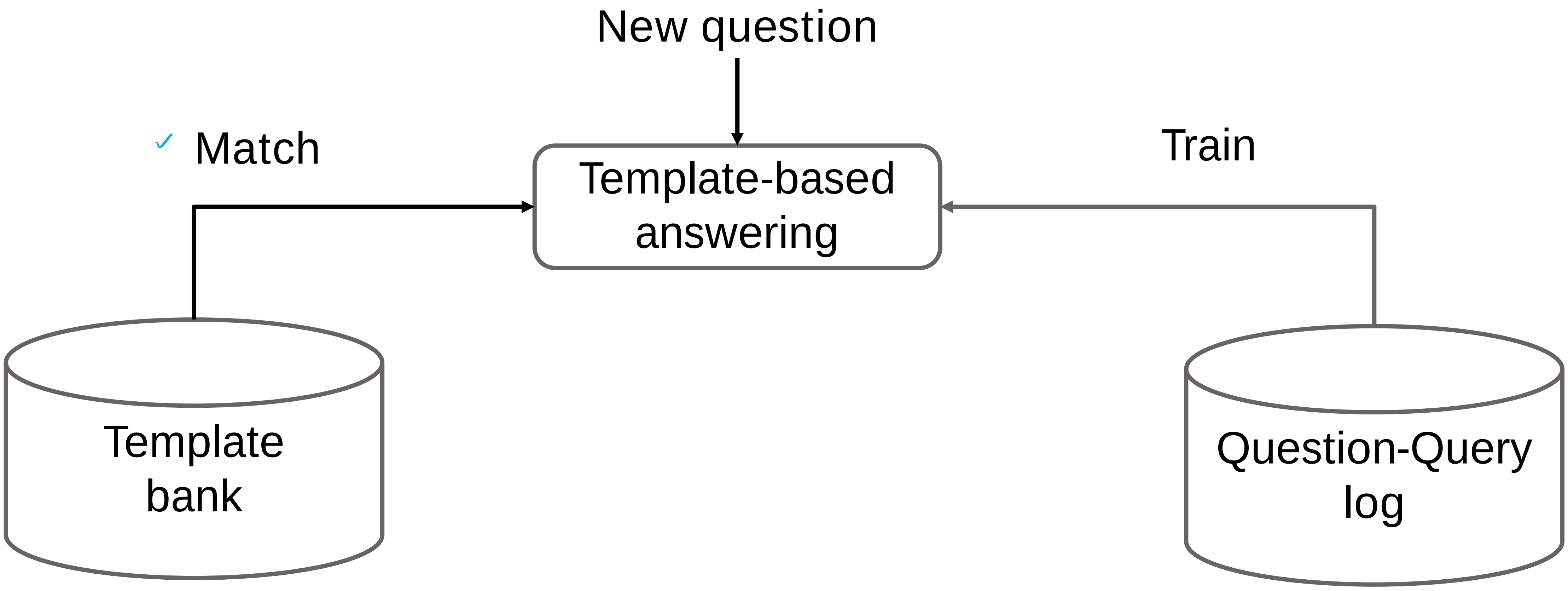
Never-ending learning with NEQA



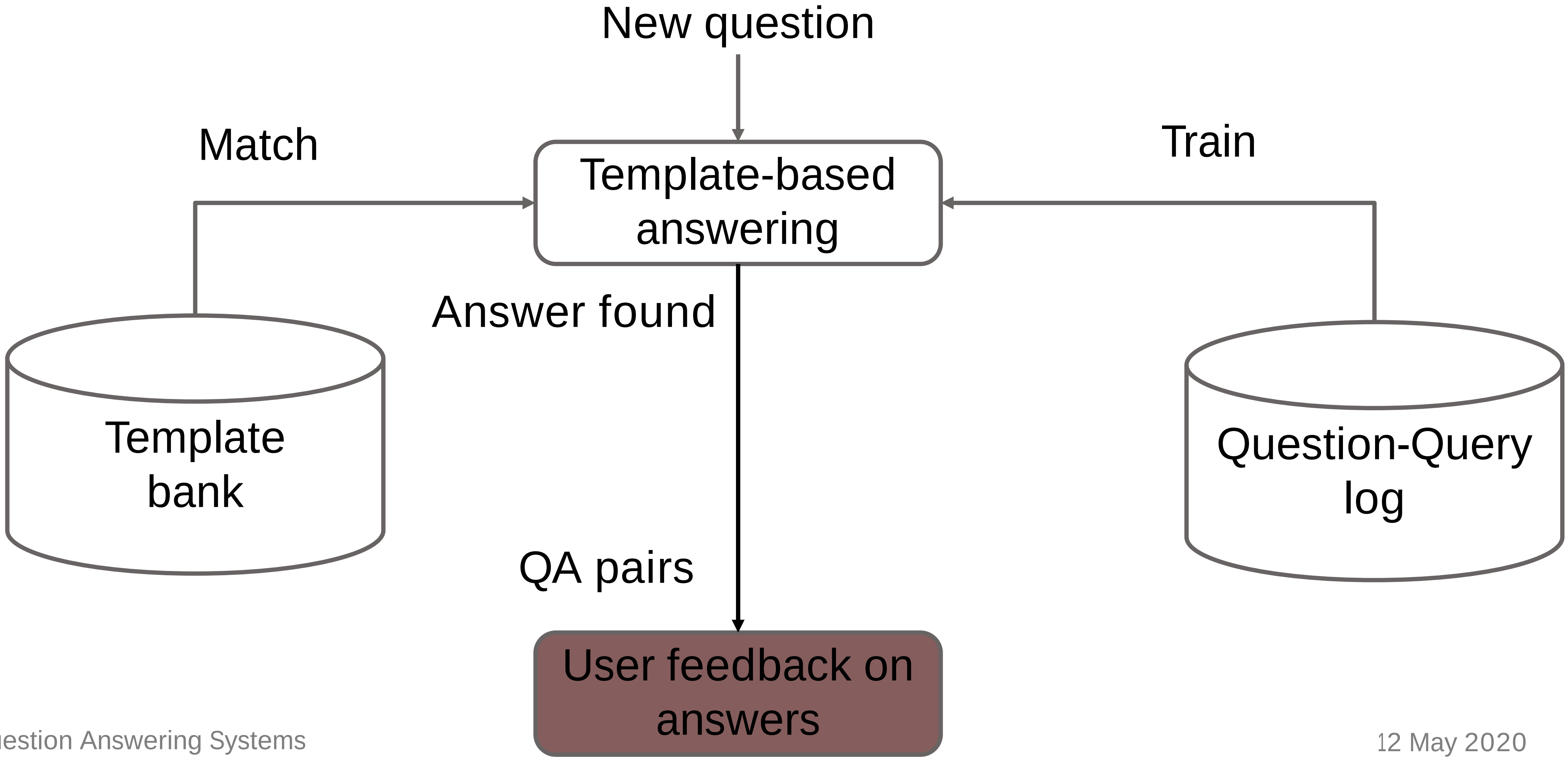
Never-ending learning with NEQA



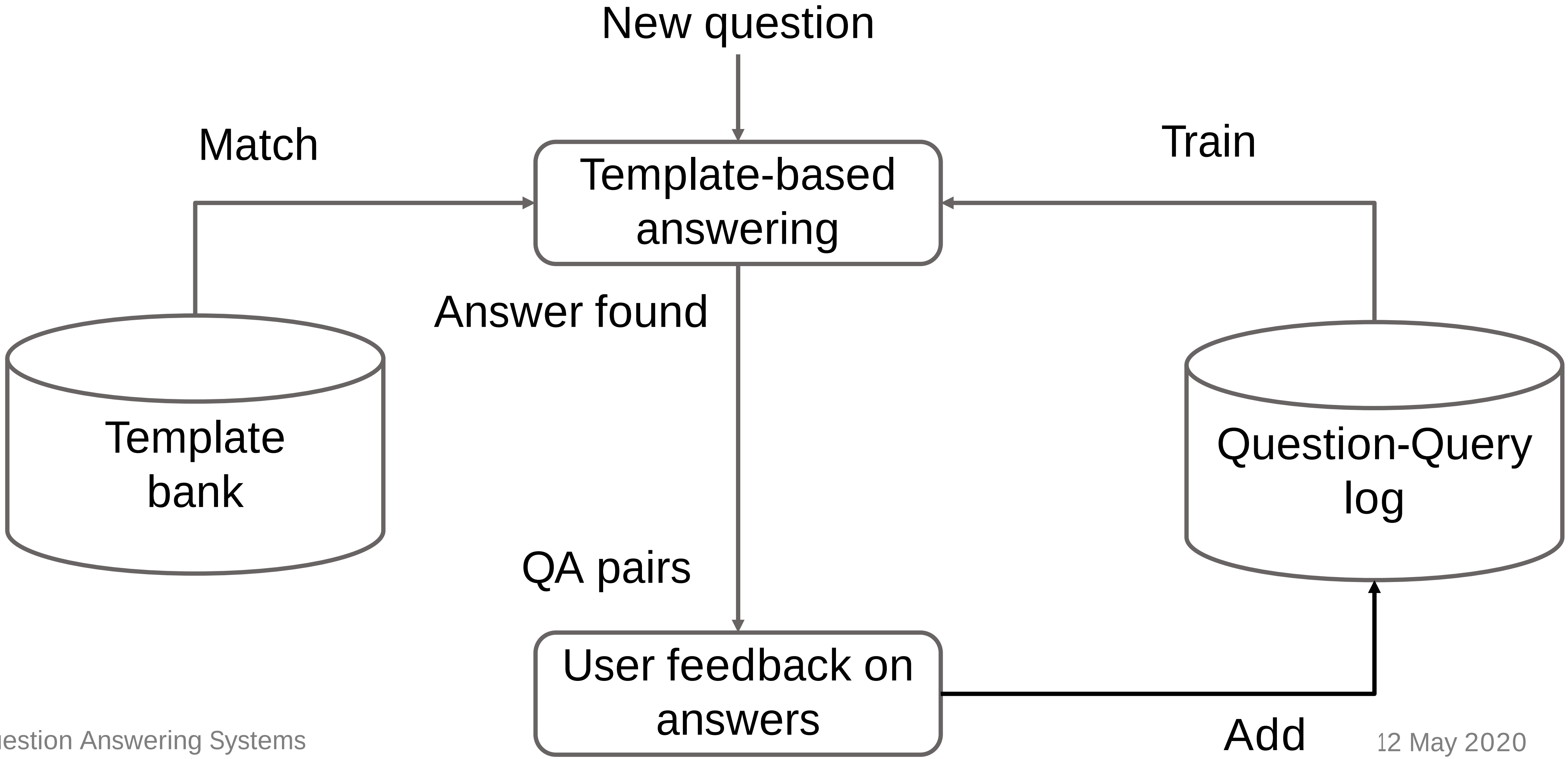
Never-ending learning with NEQA



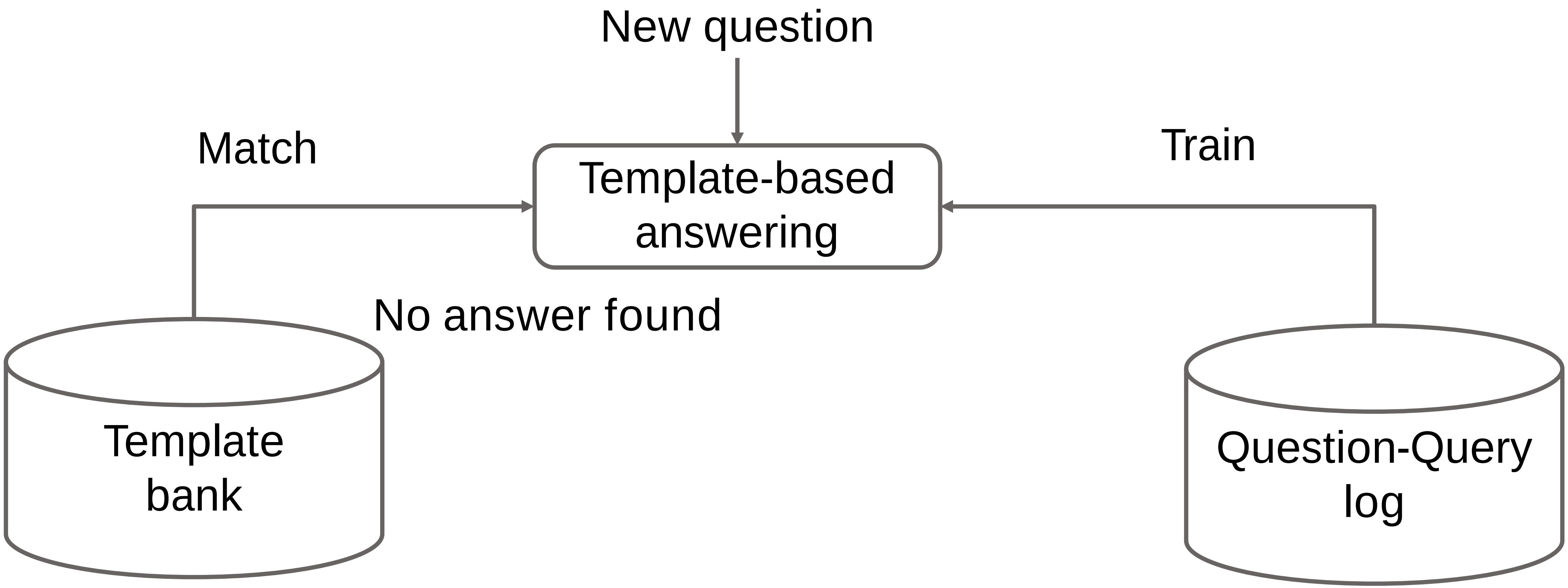
Never-ending learning with NEQA



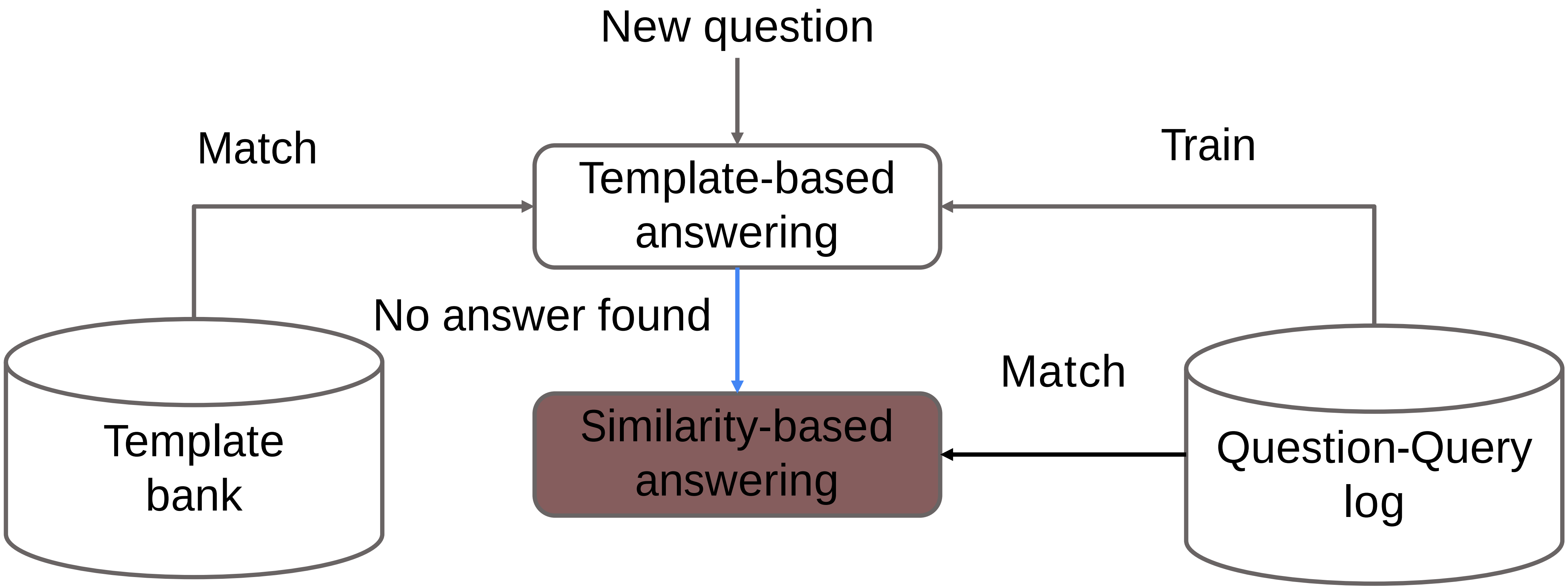
Never-ending learning with NEQA



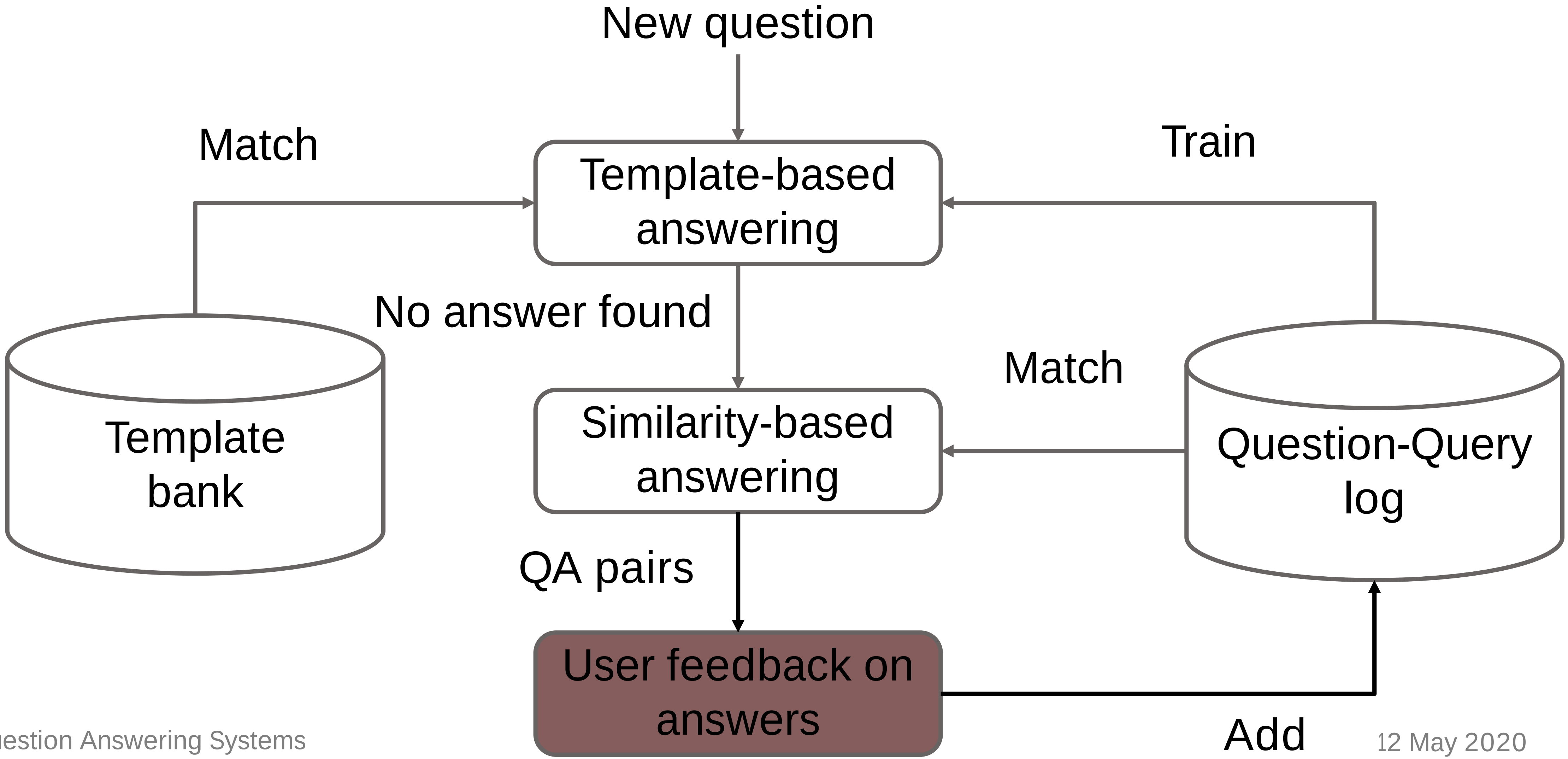
Never-ending learning with NEQA



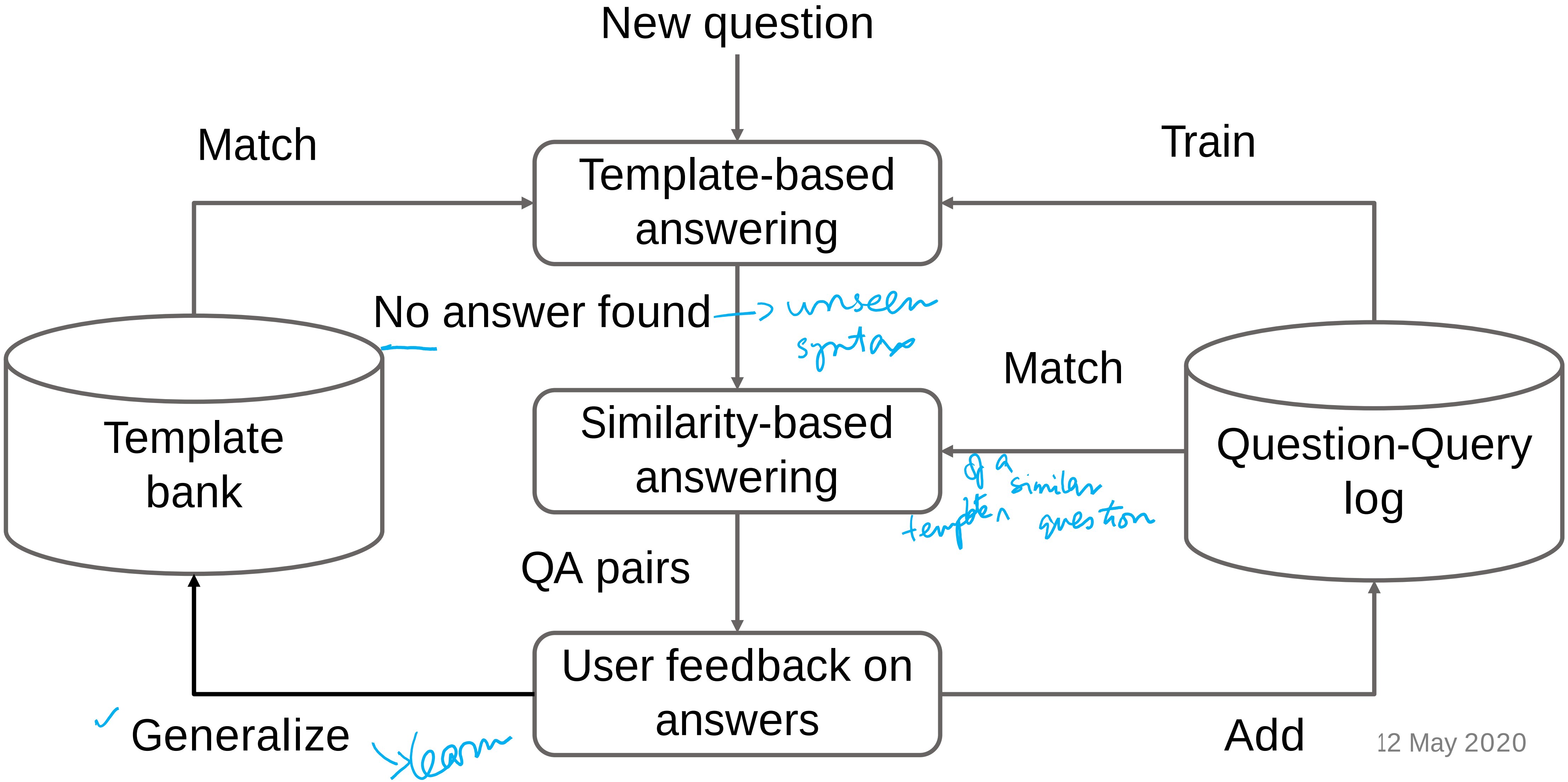
Never-ending learning with NEQA



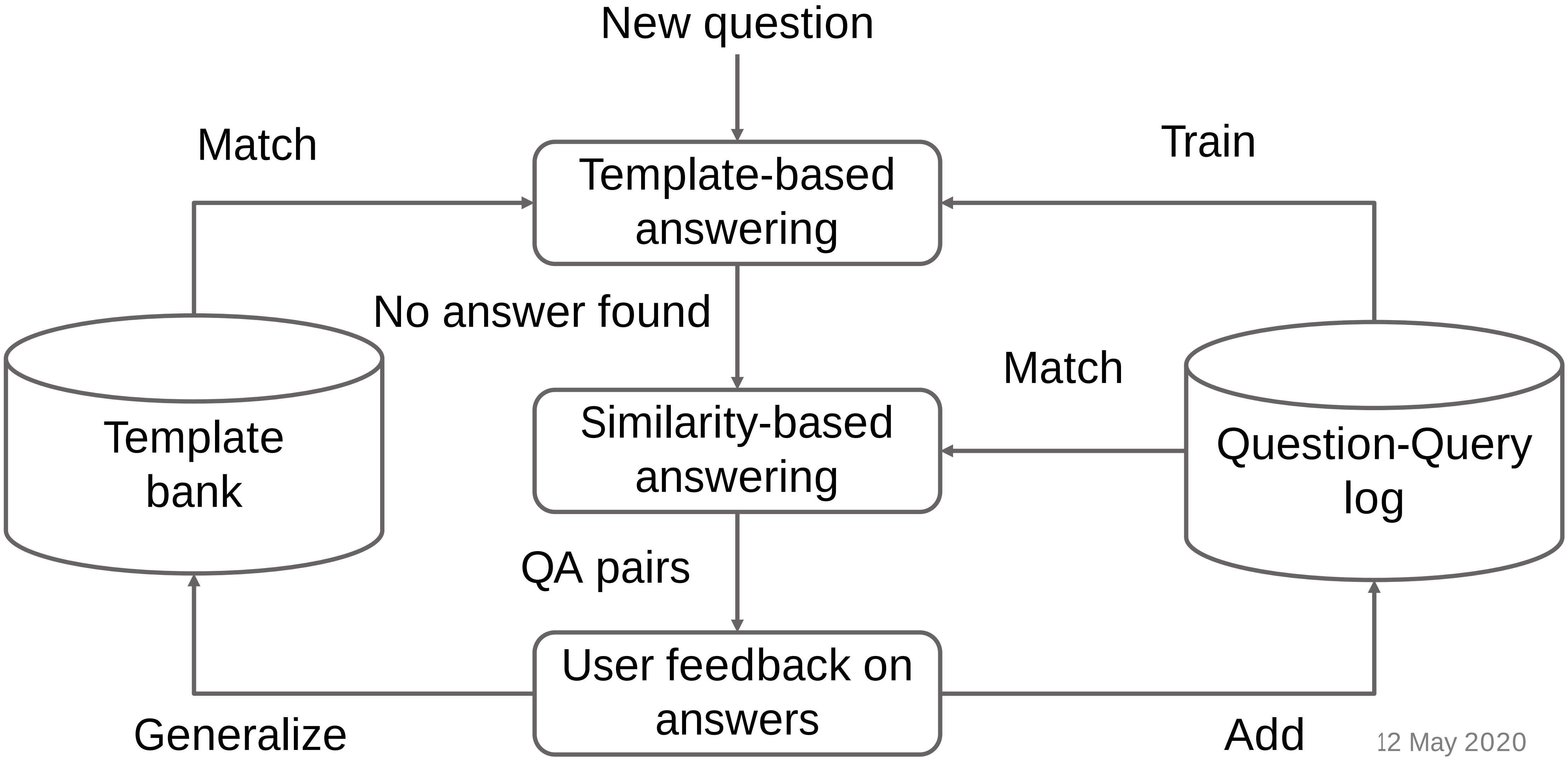
Never-ending learning with NEQA



Never-ending learning with NEQA



Never-ending learning with NEQA



Training NEQA

NL SPARQL

- Collecting question-query pairs difficult
- Start with question-answer pairs instead (can come from Web users)
- Create queries by distant supervision
- learn
  Generalize to create slot-aligned templates

strong supervision

distant superision

Distant supervision from QA pairs

Question: What are the Oscar award nominations of Nolan?

Answer: Best Director (entity)

| QA pair

Distant supervision from QA pairs

Question: What are the Oscar award nominations of Nolan?
Answer: Best Director

ChristopherNolan

Distant supervision from QA pairs

off-the-shelf → TagME, AIDA

Question: What are the Oscar award nominations of Nolan?

Answer: Best Director

(KG entity)

ID 65

BestDirector

NERD system

ChristopherNolan

Distant supervision from QA pairs

Question: What are the Oscar award nominations of Nolan?
Answer: Best Director

- Retain shortest path between question and answer entities
- Retain answer type information

ChristopherNolan → nominatedFor → Award → awardTitle → BestDirector → type → AcademyAward

pred
ent
pred
ans ent
type 1
type 2
type 3
type
CVTs (FB)
qualifiers

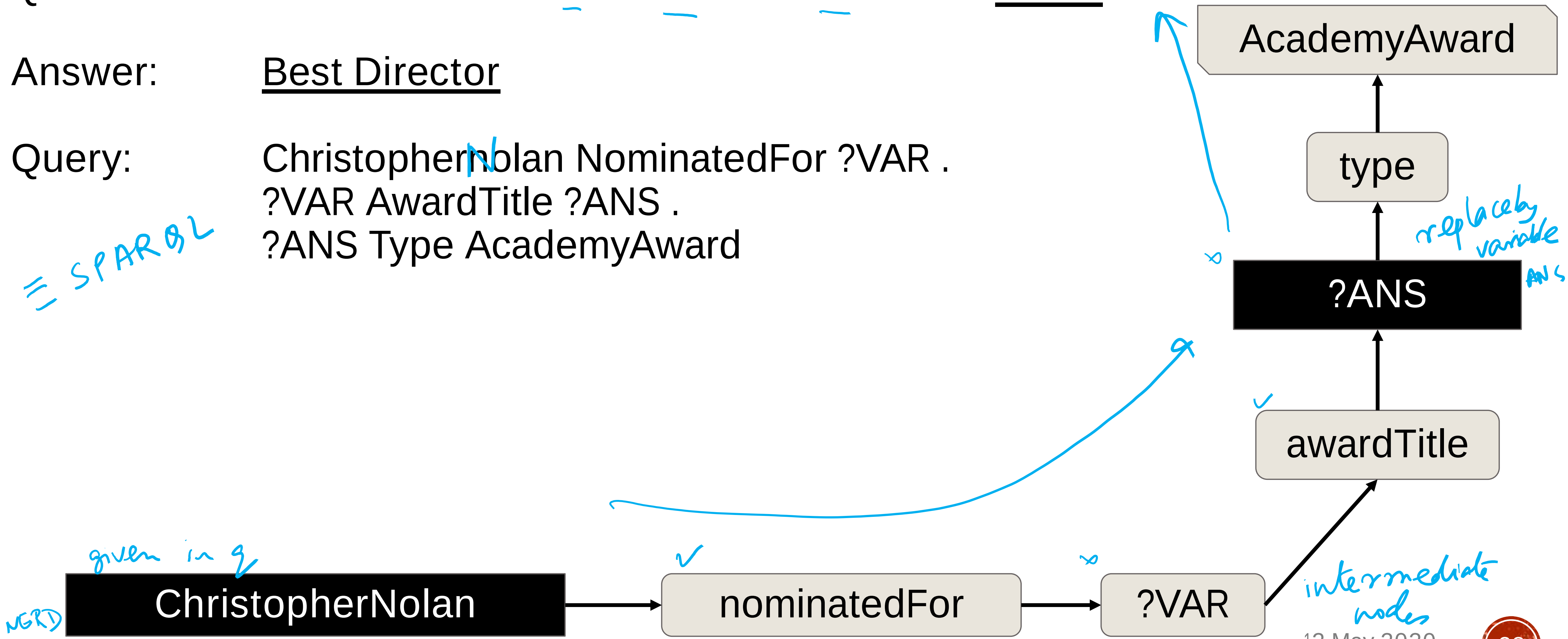
Distant supervision from QA pairs

Question: What are the Oscar award nominations of Nolan?

Answer: Best Director

Query: ChristopherNolan NominatedFor ?VAR .
?VAR AwardTitle ?ANS .
?ANS Type AcademyAward

≡ SPARQL

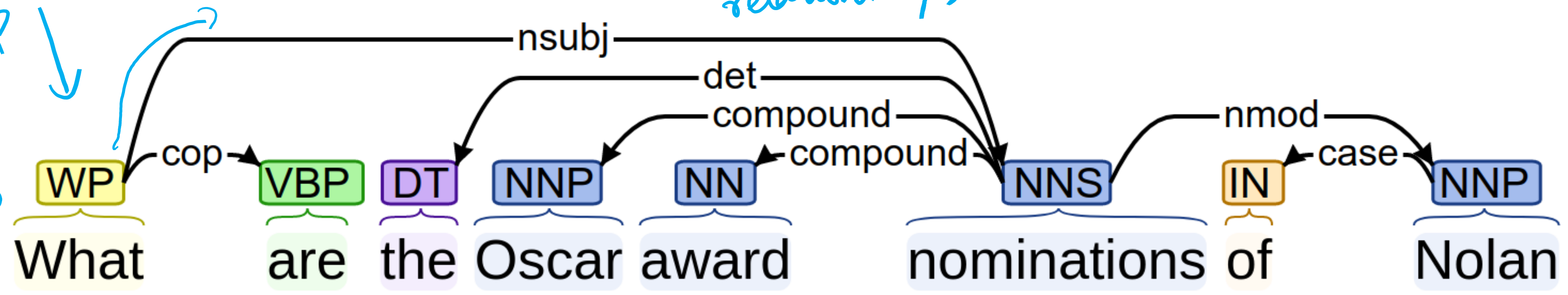


Extract question phrases

relationships

DP

pos



ques

extract phrase

DP n-gram

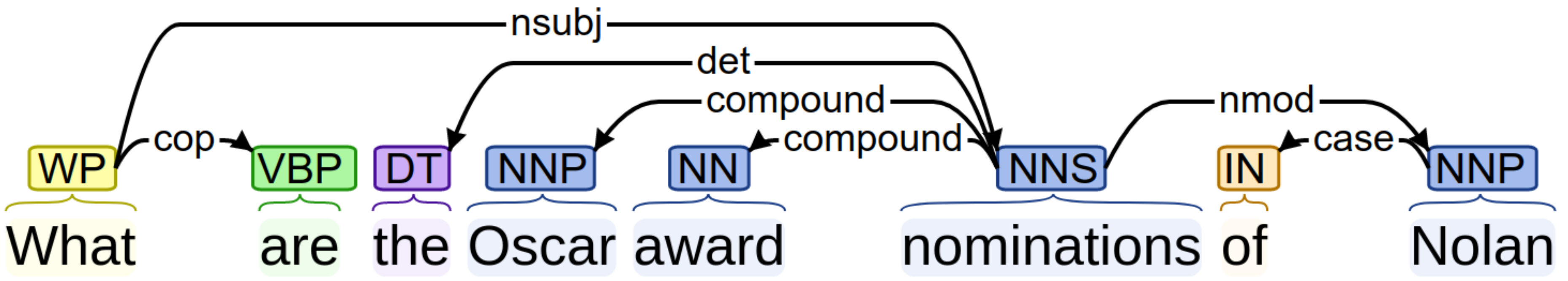
unigrams

what nominations	oscar nominations	oscar award	nominations of	u oscar

what are	oscar award nominations	u nominations	u award	bigrams award nominations

Dependency parsing: https://web.stanford.edu/~jurafsky/slp3/15.pdf
Constituency parsing: https://web.stanford.edu/~jurafsky/slp3/13.pdf

Extract query items



→ entity

Nolan

CN via NERD

what nominations | oscar nominations | oscar award | nominations of | oscar

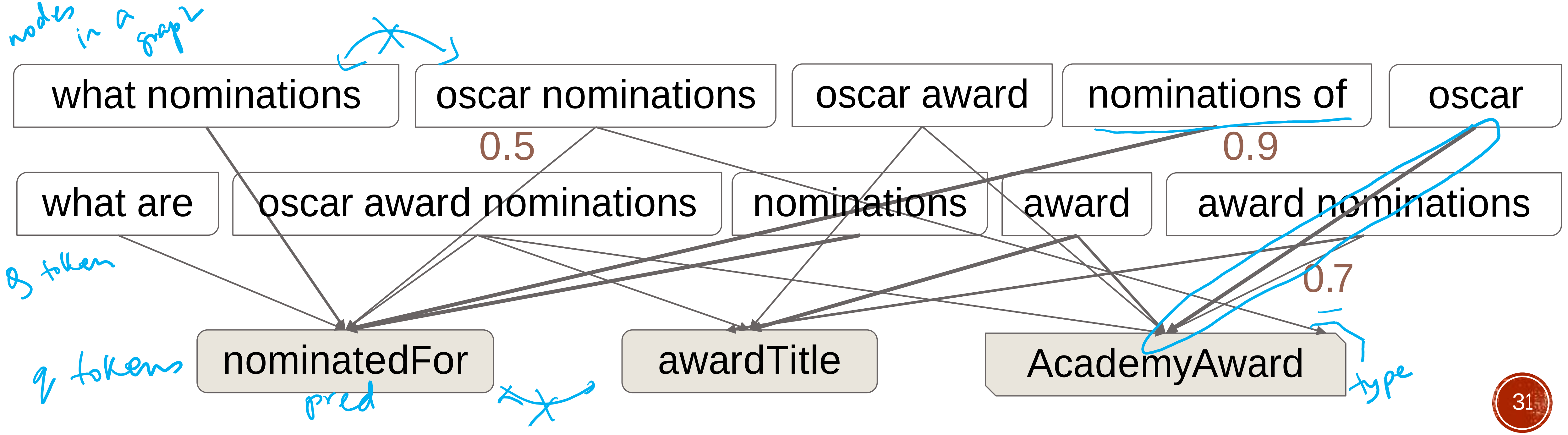
what are | oscar award nominations | nominations | award | award nominations

query items → nominatedFor | awardTitle | AcademyAward

Create candidate alignments

- Bipartite graph with edge weights (Yahya et al. 2012)
- Weights from lexicons L_P and L_T (Abujabal et al. 2017, Berant and Liang 2013)

predicate type



Create candidate alignments

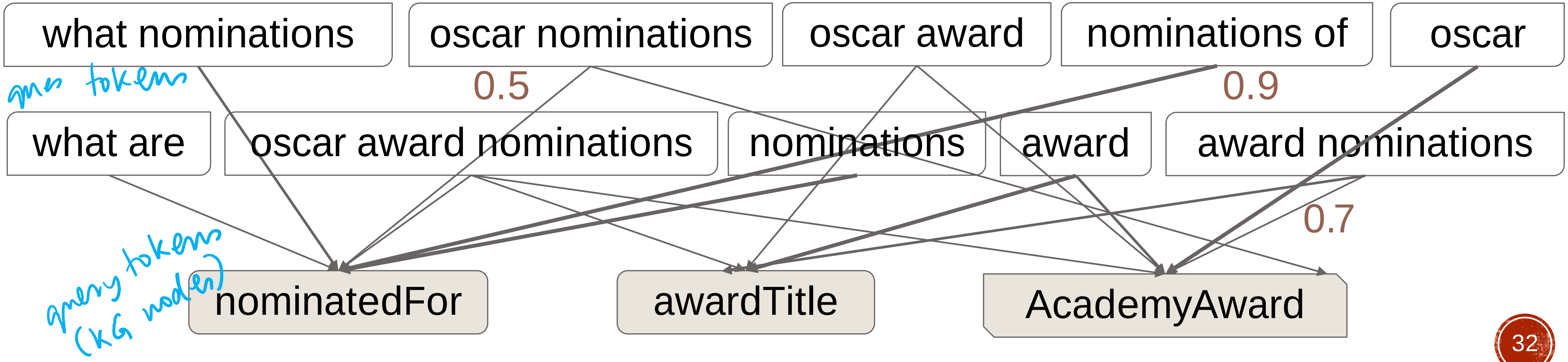
Lp

Phrase	KG Predicate	Weight
nominee for	nominatedFor	0.8
nominations of	nominatedFor	0.9
oscar nominations	nominatedFor	0.5

NL
annotation
E1
E2, KG item

LT

Phrase NL	KG Type	Weight
Academy Award	AcademyAward	0.9
Oscar	AcademyAward	0.7
Oscar Award	AcademyAward	0.8



Optimal mapping via ILP

obj fn.
+ constraints

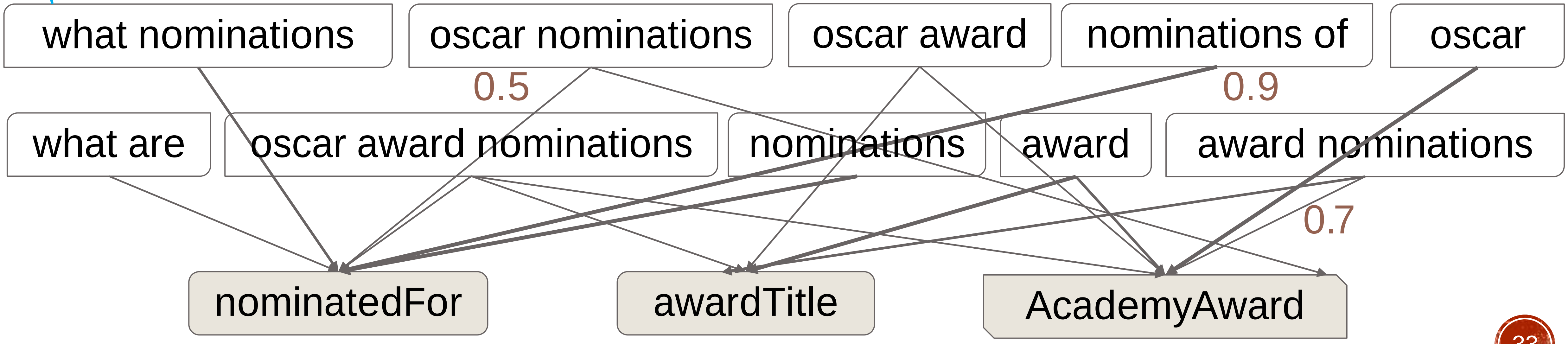
- Best alignment of items with Integer Linear Program (ILP)
- Constraint 1: Each KG item obtained from at most one phrase
- Constraint 2: Token contributing to entity cannot contribute to any other phrase

nolan

- Constraint 3: One phrase can map to at most one type

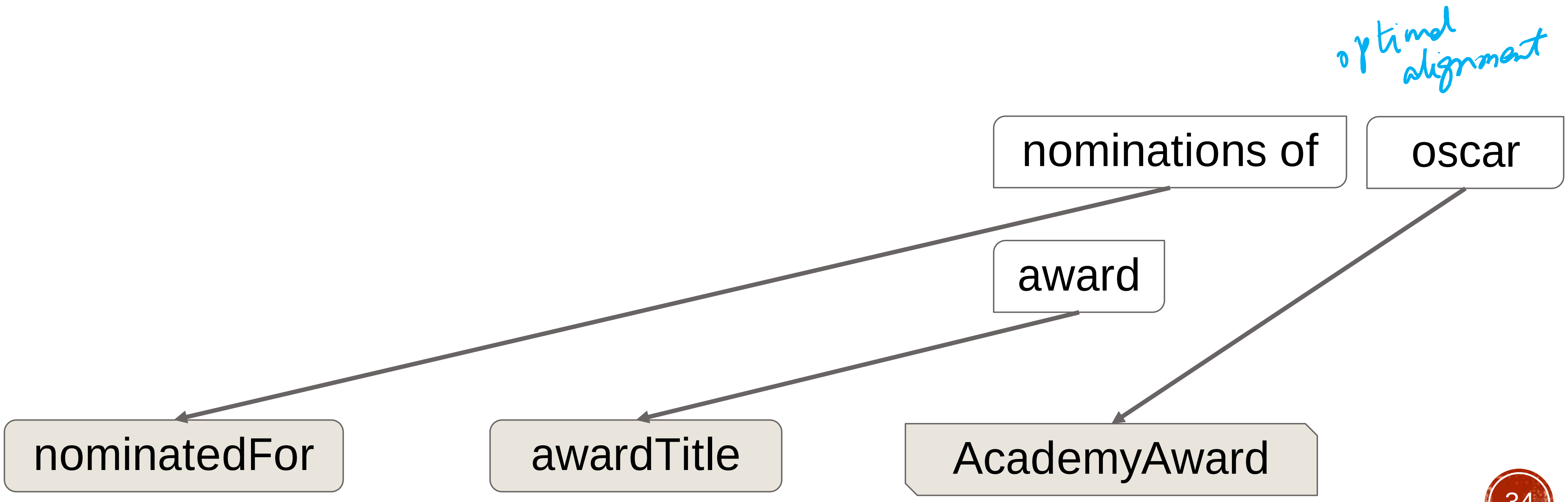
what goes where

phrase ↓ → n-grams / DP n-grams



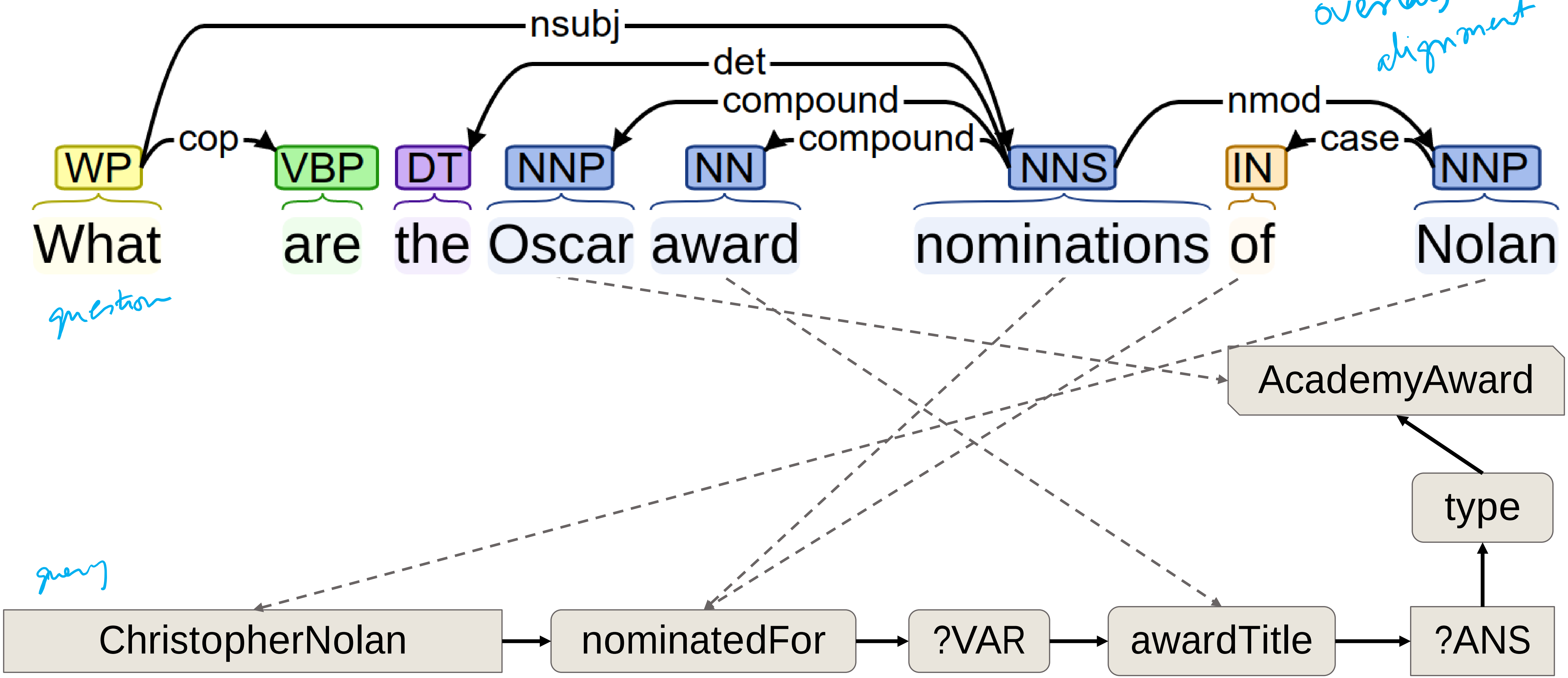
Optimal mapping via ILP

- Best alignment of items with Integer Linear Program (ILP)
- Constraint 1: Each KG item obtained from at most one phrase
- Constraint 2: Token contributing to entity cannot contribute to any other phrase
- Constraint 3: One phrase can map to at most one type

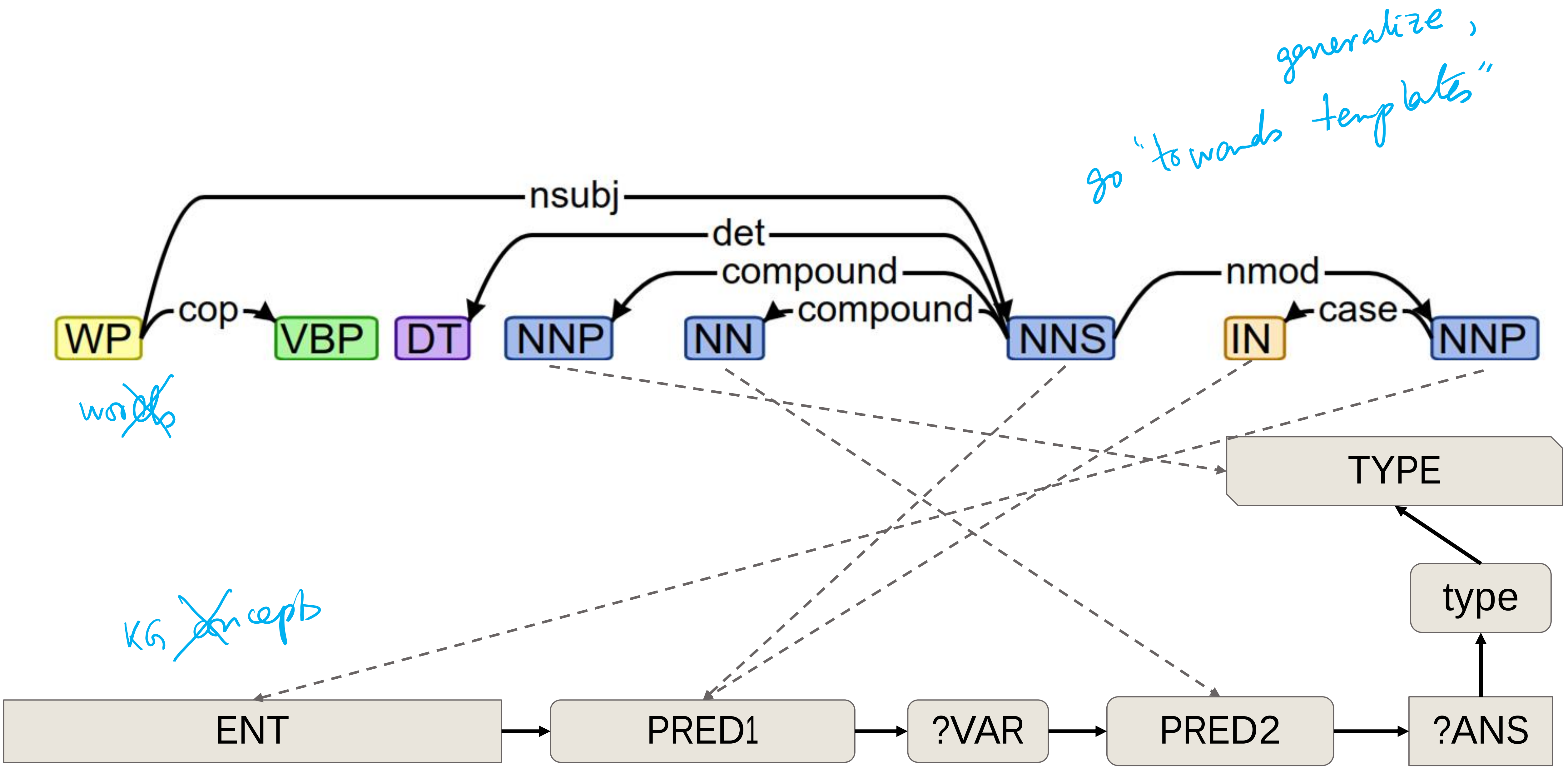


Apply alignment to question-query

superimpose/
overlay
alignment



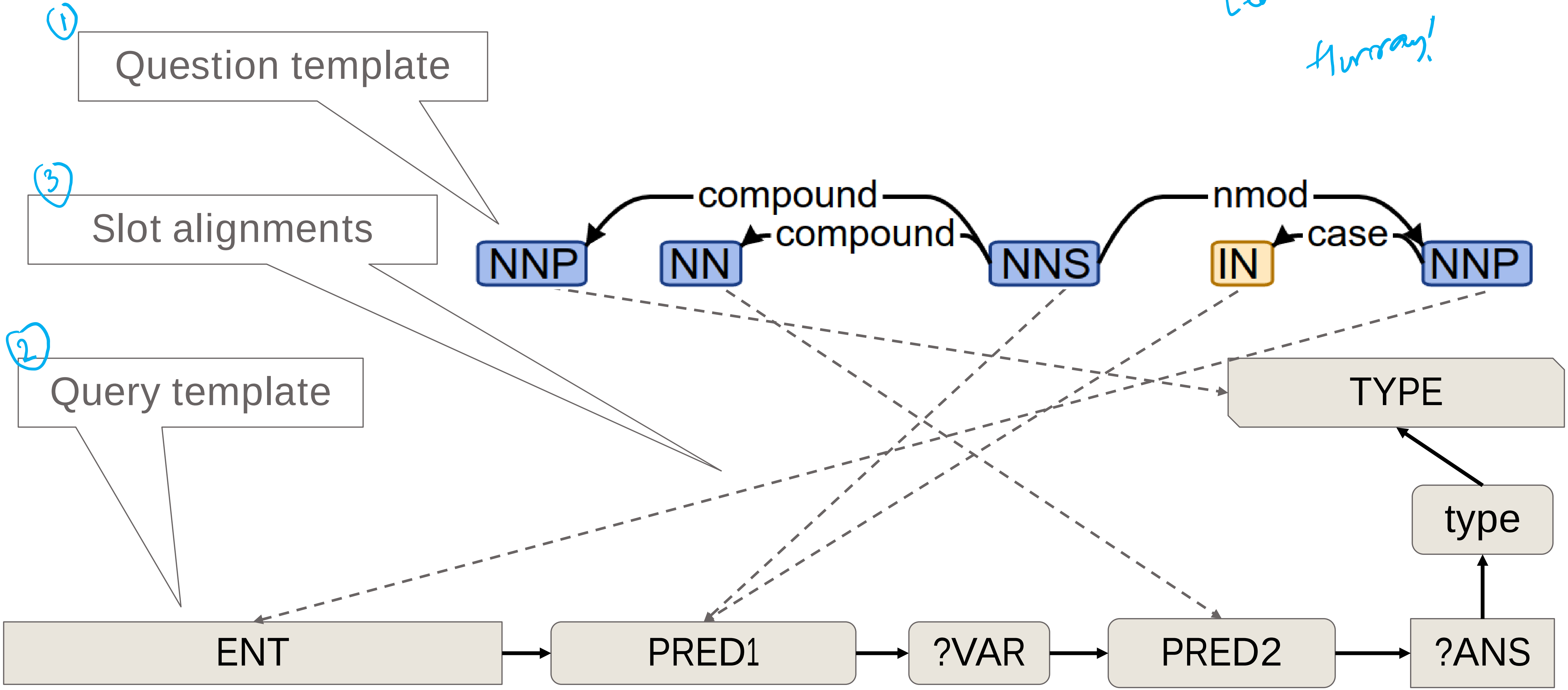
Replace concrete items by roles



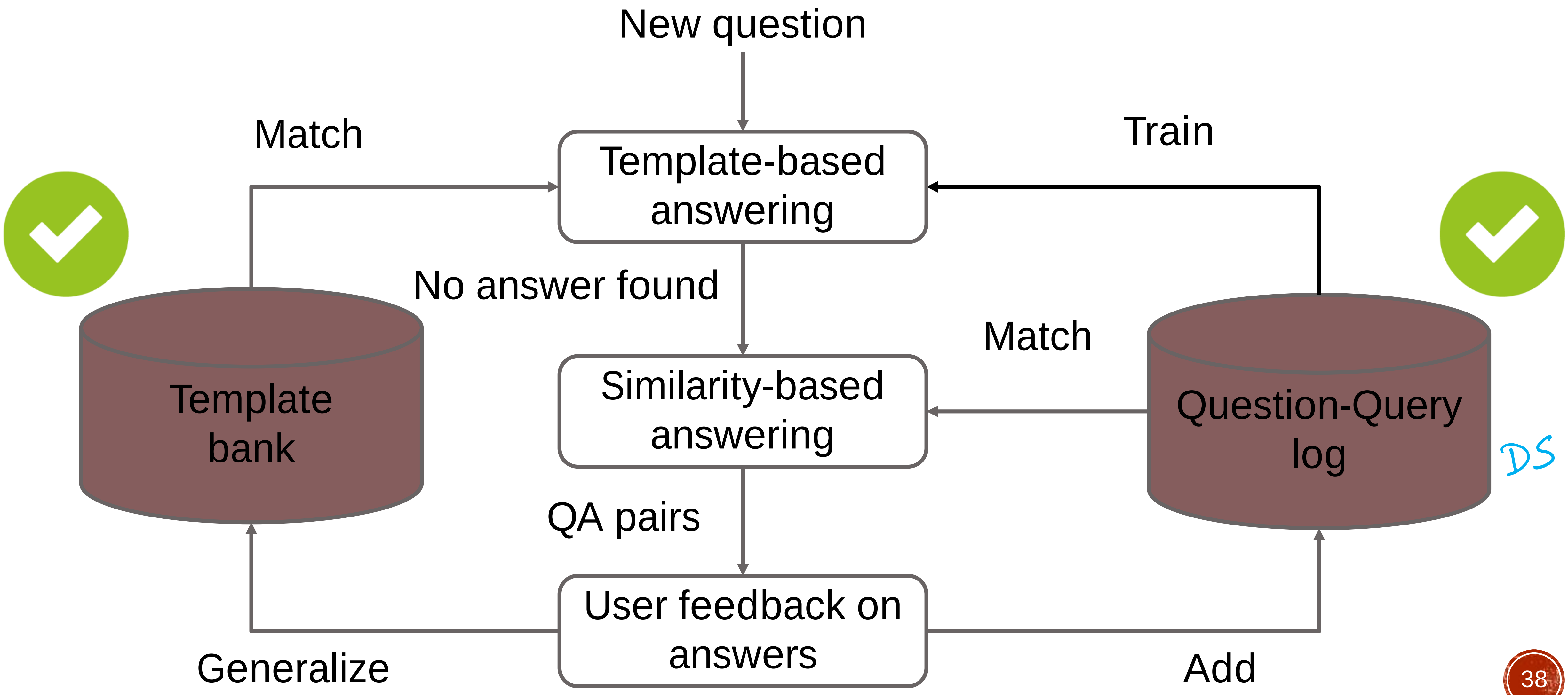
Drop unnecessary question words

Learnt a template!

Hurray!

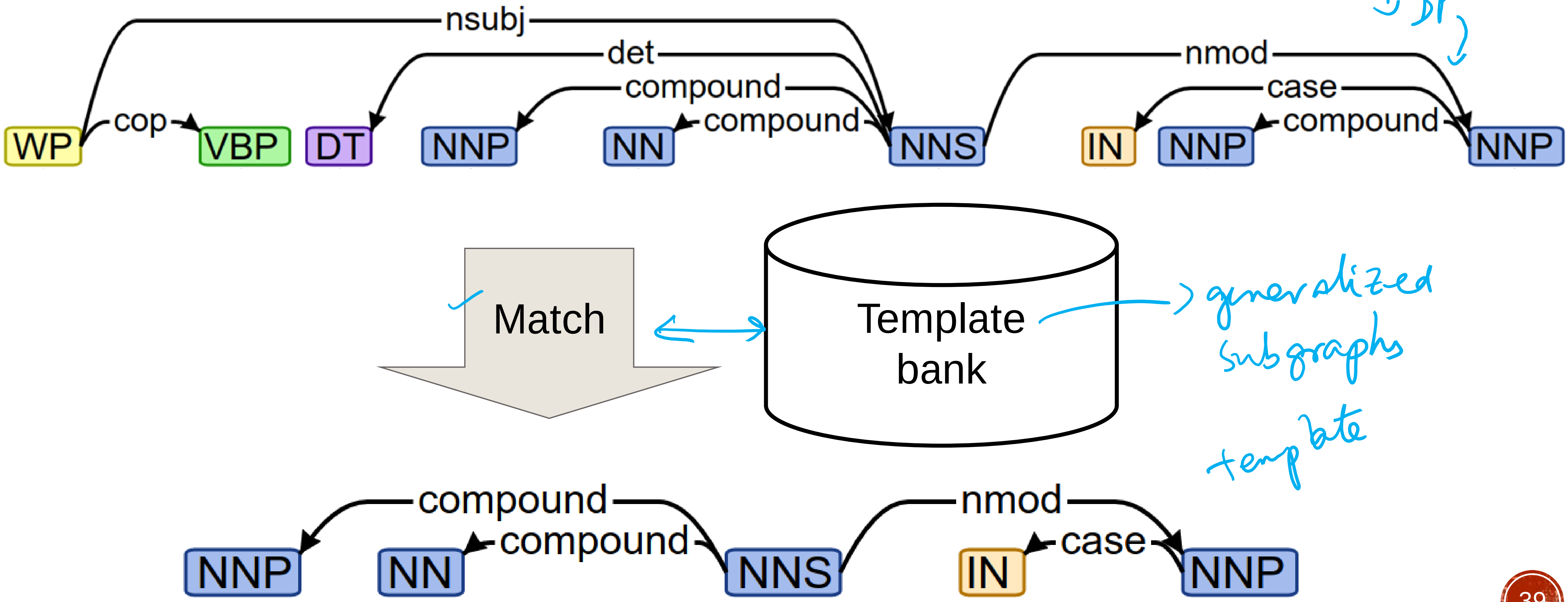


Never-ending learning with NEQA

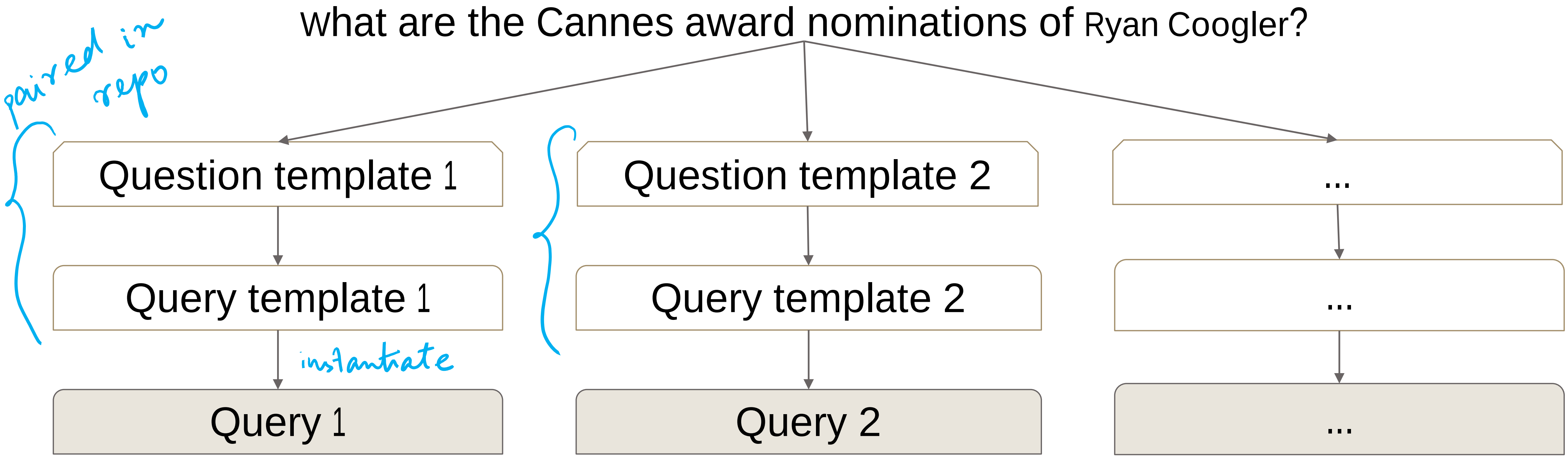


Answering with templates

✓ New question: What are the Cannes award nominations of Ryan Coogler?



Instantiating queries



RyanCoogler nominated ?VAR
.
?VAR awardTitle ?ANS .
?ANS Type CannesAward

$q_1 \neq q_2$

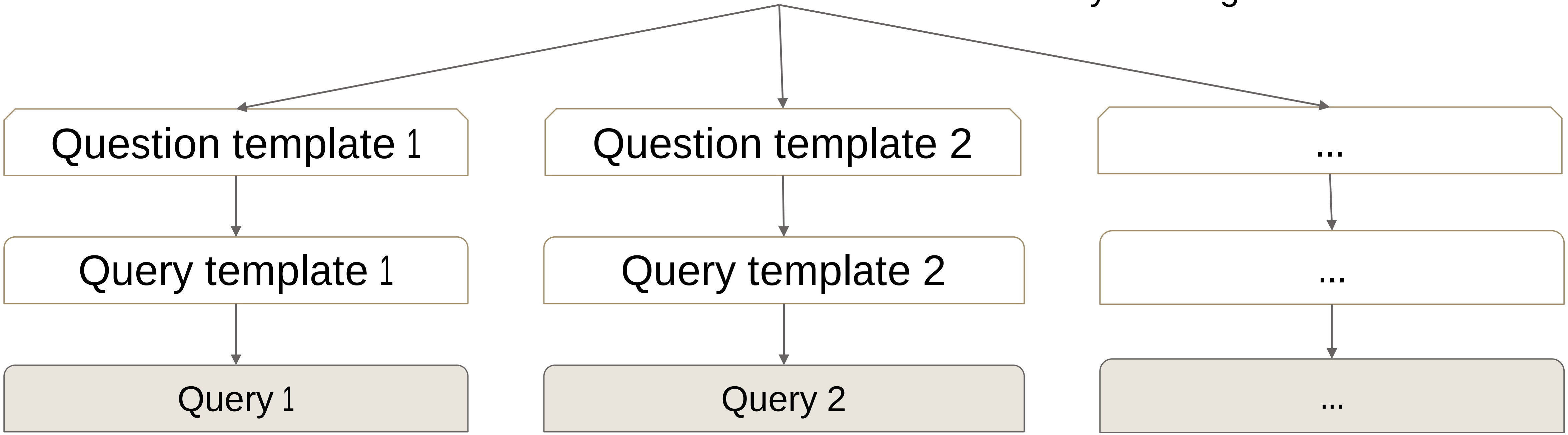
RyanCoogler awarded ?VAR .
?VAR awardTitle ?ANS .
?ANS Type GoldenGlobe

q_3

...

Instantiating queries

What are the Cannes award nominations of Ryan Coogler?



RyanCoogler nominated ?VAR
.
?VAR awardTitle ?ANS .
?ANS Type CannesAward

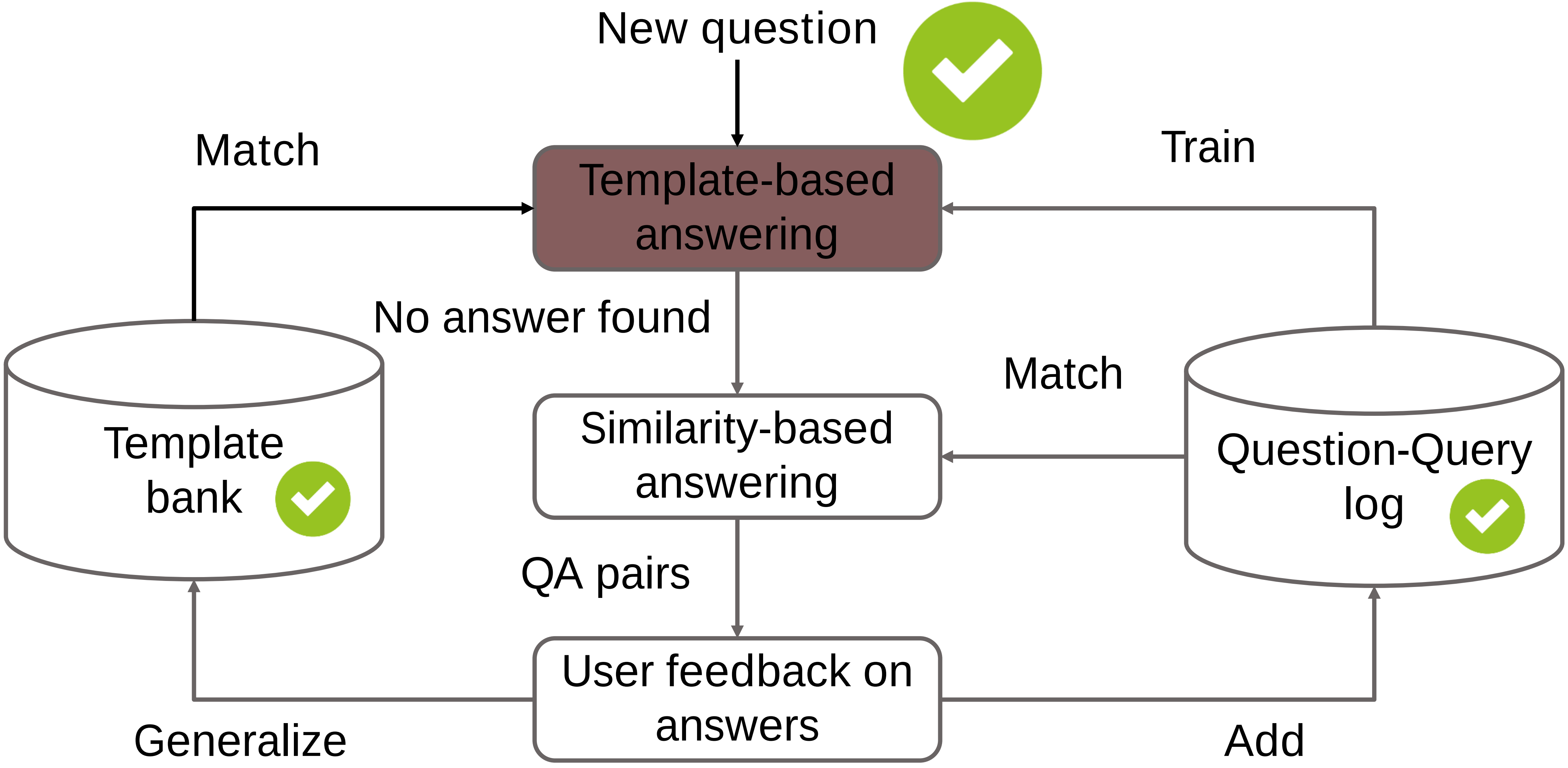
RyanCoogler awarded ?VAR .
?VAR awardTitle ?ANS .
?ANS Type GoldenGlobe

Rank queries with learning to rank

and execute best query

pairwise LTR

Never-ending learning with NEQA



Tackling language diversity

We resume at 15:15

- If we can answer:
  - What are the Oscar award nominations of Nolan?

Same syntax!

- Then we should be able to answer:
  - What are the Cannes award nominations of Ryan Coogler?
  - Which Oscar award nominations did Nolan receive?

Same semantics!

ent-res
pred-prop
type-class

SPO /structure
E → S/O
T → O
P → P

Tackling language diversity: Semantics

- If we can answer:
  - What are the Oscar award nominations of Nolan?
- Then we should be able to answer:
  - Which Oscar award nominations did Nolan receive?

Same semantics!

Semantic similarity: Component 1

Q_{log}: What are the Oscar award nominations of Nolan?

Q_{new}: Which Oscar nominations did Nolan receive?

reward
surface level
match

- Language models (LM): Computed using maximum likelihood probabilities of n-grams from Q_{new} in Q_{log}

Semantic similarity: Component 2

already in log

Q_{log}: What are the Oscar nominations of Nolan?

all q-s in log

Q_{new}: Which Academy Award nominations did Nolan receive?

new question

film
movie

- Word2Vec: Cosine similarity between contextual embeddings of words and phrases in Q_{new} and Q_{log}

cos (word2vec(oscar), w2v-emb(academy))

Semantic similarity: Component 3

in log

Q_{log}: What are the Oscar award nominations of Nolan?

Q'_{log}: Which films by Nolan have Oscar award nominations?

Q_{new}: Which Oscar nominations did Nolan receive?

sim matching
- support ~> LM surface
- typicality ~> type
- specificity

expected

- Answer types: Cosine similarity between expected answer types (awards, films, directors) of Q_{new} and Q_{log} [Own work in Ziegler et al. 2017]

Answering with similarity function

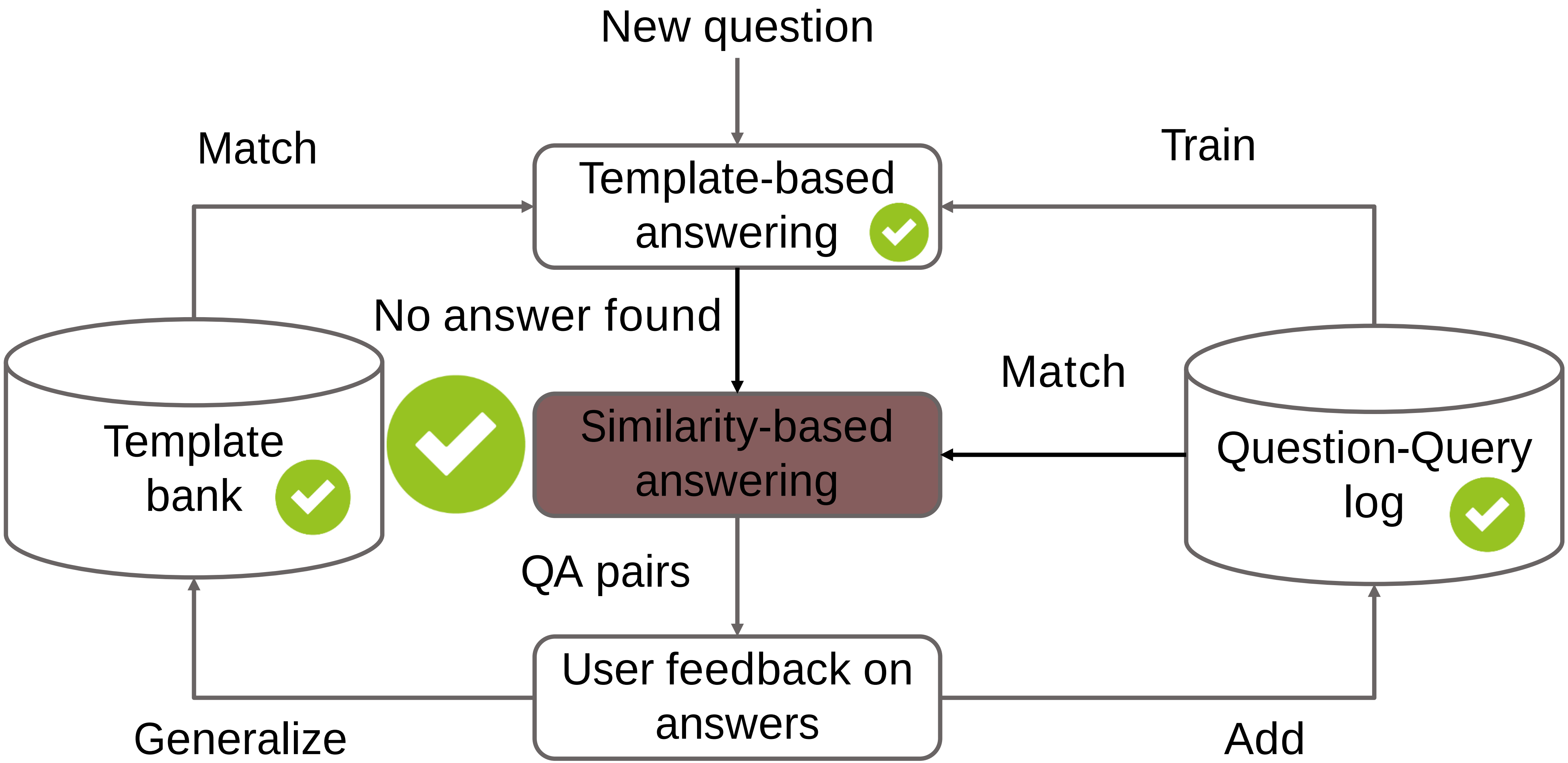
- Similarity score: Linear combination of three factors

  sim score   all
- Compute for new question against questions in log
- Find best question (max score) $q_{best} \rightarrow q^{temp}_{best} \rightsquigarrow q^{temp}_{best} \rightsquigarrow q_{best}$
- Execute corresponding query

  onKG

  ANS

Never-ending learning with NEQA



Closing the loop with user feedback

- So far, assumed all answers were correct: Pseudo-relevance

  Blind feedback

- Pseudo-relevance degrades quality
- Users provide feedback on answers

Model

I/P → O/P → Feedback

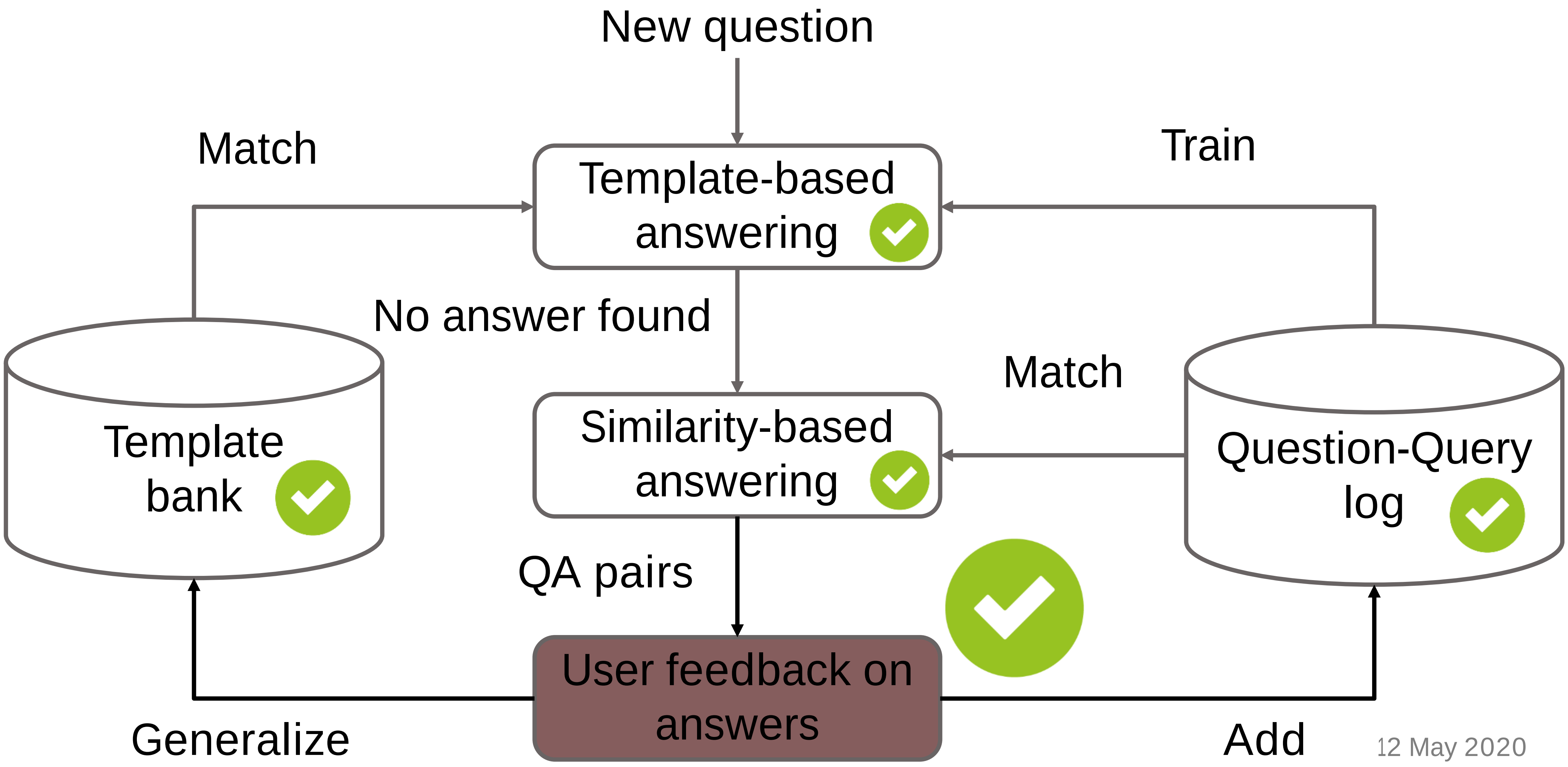
Question:	Which Oscar nominations did Nolan receive?
Answer:	Best Director ✓
User:	

Feedback triggers continuous learning

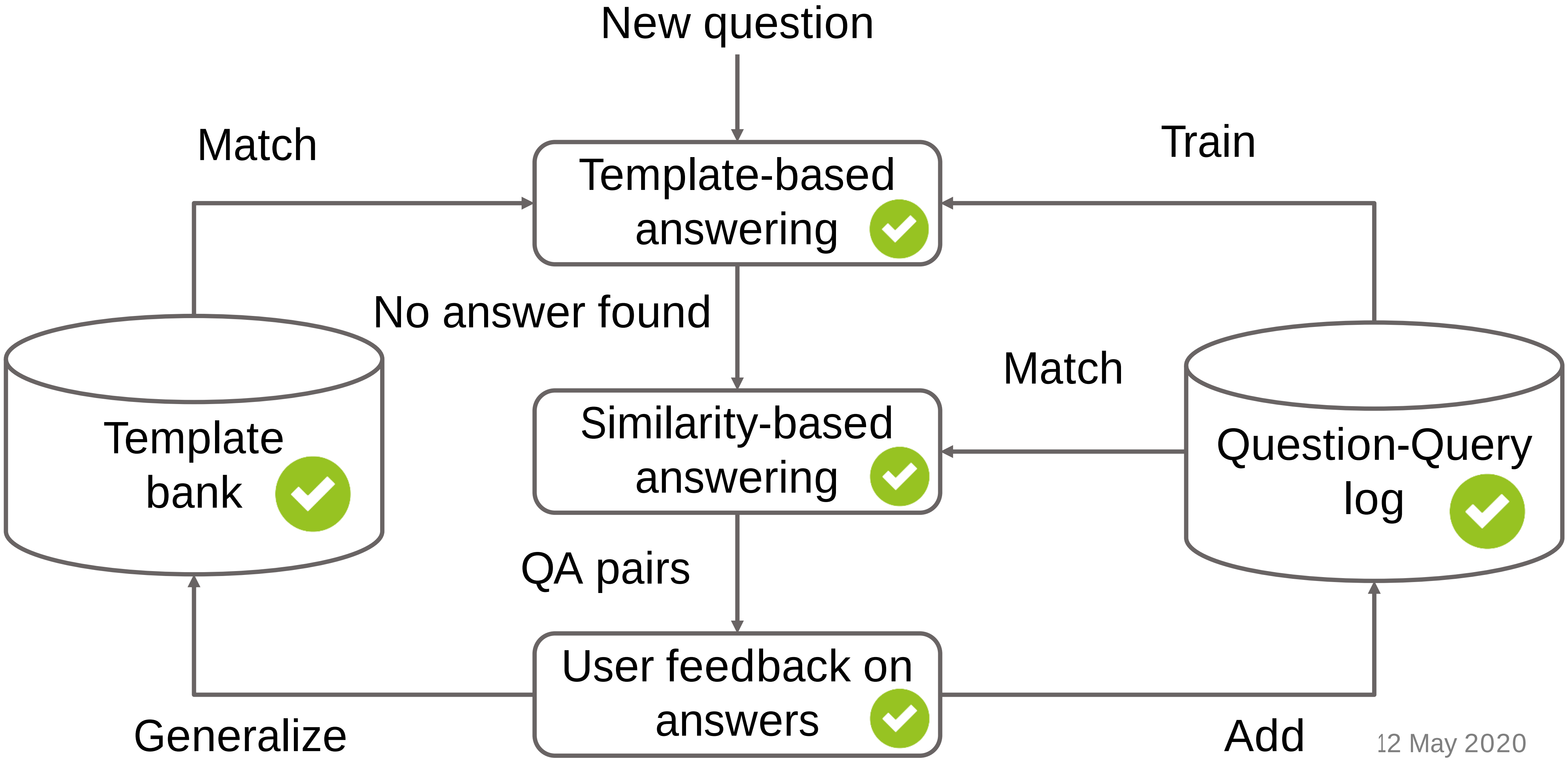
+ve

- Feedback on answer propagated to query
- Positive feedback:
  - Learn new template from question-query
  - Add new question-query to log
  - Update learning-to-rank model (query ranking)

Never-ending learning with NEQA



Never-ending learning with NEQA



Experimental results: Setup

- Benchmark: WebQuestions (6K questions)
- KG: Freebase (1.9B triples)
- Metric: F1-Score (harmonic mean of precision and recall)
- Baselines: KG-QA algorithms

Experimental results: Performance

300Q

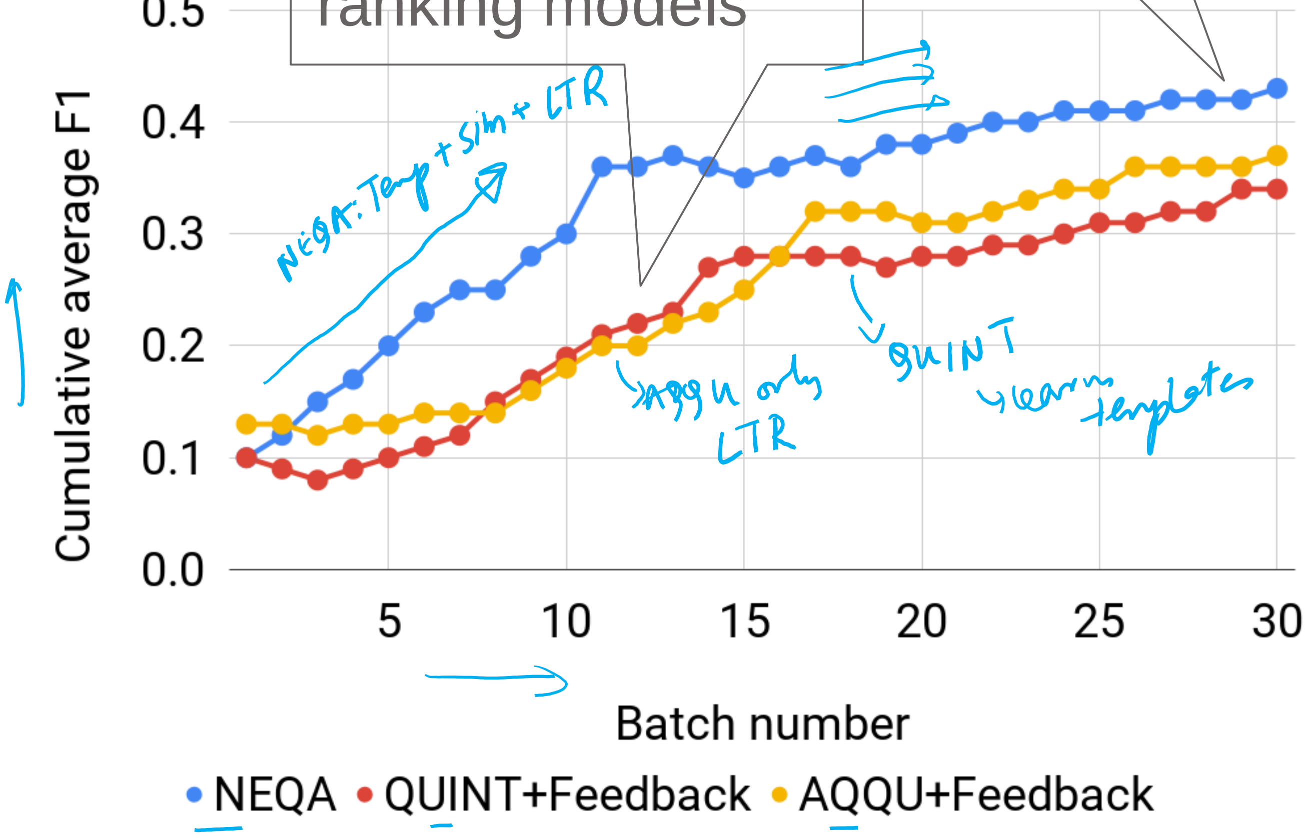
Starts small but improves over static (deep) learning systems ~3K

Method	F1-Score
NEQA	0.510*
Savenkov et al. '16	0.490
Xu et al. '16	0.470
Reddy et al. '16	0.500
Bast and Haussmann '15	0.494
Bordes et al. '14	0.392

Dol

NEQA learns new templates

Baselines update ranking models



Experimental results: Anecdotes

With learnt template: What is the name of the currency used in [Italy]?

diff syntax

With similarity function: What is the currency in [Denmark]?

diff syntax

With similarity function: What kind of money is used in [Israel]?

same semantics

The QUINT+NEQA family

QUINT

- Automated template generation for question answering over knowledge graphs — 79 — 2017
  A Abujabal, M Yahya, M Riedewald, G Weikum
  Proceedings of the 26th international conference on world wide web, 1191-1200

demo

- Quint: Interpretable question answering over knowledge bases — 16 — 2017
  A Abujabal, RS Roy, M Yahya, G Weikum
  Proceedings of the 2017 Conference on Empirical Methods in Natural Language ...

NEQA

- Never-ending learning for open-domain question answering over knowledge bases — 23 — 2018
  A Abujabal, R Saha Roy, M Yahya, G Weikum
  Proceedings of the 2018 World Wide Web Conference, 1053-1062

TIPI

- Efficiency-aware Answering of Compositional Questions using Answer Type Prediction — 3 — 2017
  D Ziegler, A Abujabal, RS Roy, G Weikum
  Proceedings of the Eighth International Joint Conference on Natural Language ...

TEQUILA

- TEQUILA: Temporal question answering over knowledge bases — 7 — 2018
  Z Jia, A Abujabal, R Saha Roy, J Strötgen, G Weikum
  Proceedings of the 27th ACM International Conference on Information and ...

- TempQuestions: A Benchmark for Temporal Question Answering — 7 — 2018
  Z Jia, A Abujabal, R Saha Roy, J Strötgen, G Weikum
  Companion Proceedings of the The Web Conference 2018, 1057-1062

Research paper 2

Learning Surface Text Patterns for a Question Answering System

Learning surface text patterns for a question answering system
D Ravichandran, E Hovy - Proceedings of the 40th annual meeting on …, 2002 - dl.acm.org
In this paper we explore the power of surface text patterns for open-domain question answering systems. In order to obtain an optimal set of patterns, we have developed a method for learning such patterns automatically. A tagged corpus is built from the Internet in a bootstrapping process by providing a few hand-crafted examples of each question type to Altavista. Patterns are then automatically extracted from the returned documents and standardized. We calculate the precision of each pattern, and the average precision for each …
Cited by 1050 Related articles All 28 versions

Surface answer patterns

- Apply surface patterns → templates to pinpoint answer
  “<person> was born in <year> ...”
  “<person> (<year>--<year>)...”

Reproduced with permission from (Ravichandran & Hovy 02)

- Learn:

  for each → question type Qtarget:

  – submit anchor terms to search engines

  – extract sentences

  – apply suffix tree

  – measure precision

Qtarget	# of questions	MRR on TREC-10
BirthYear	8	0.478
Inventor	6	0.167
Discoverer	4	0.125
Definition	102	0.34
WhyFamous	3	0.667
Location	16	0.750

Outline

words/phrases/entities instead of docs/passages

- Direct answers from text
- Factoid, open-domain QA

general domain

method

- Learning patterns
- Answering with patterns
- Uses bootstrapping

Mozart

When was X born?

text

"Mozart was born in 1756."
"Gandhi (1869–1948)..."

"<NAME> was born in <BIRTHDATE>"
"<NAME> (<BIRTHDATE>–"

Learning patterns (I)

→ example → seed

- Select an example BIRTHYR When was Moz born? → {Mozart, 1756}
- Submit keywords to search engine → Mozart 1756 → Google / Altavista
- Get top-1000 docs → relevance
- Preprocess docs for regex matching with tools like egrep → pattern matching
  - Split into sentences Stanford sentence splitter
  - Keep only sentences with keywords Mozart + 1756, prune others
  - Normalize whitespace space, ENTER, TAB
  - Remove tags HTML

Learning patterns (II)

trie

cat, catch, cater

- Pass each retained sentences through a suffix tree constructor

storage

- Retain only those phrases in the suffix tree that contain question and answer term

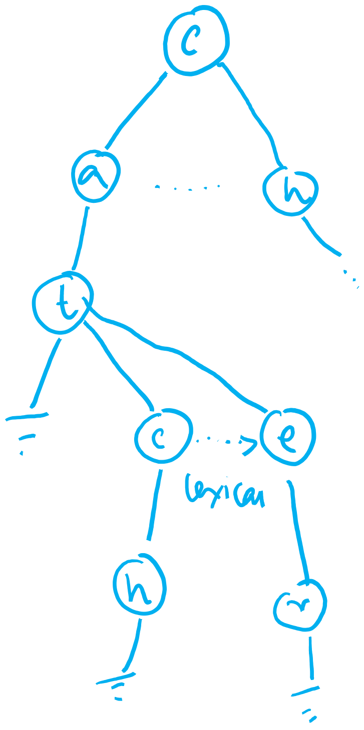
filter

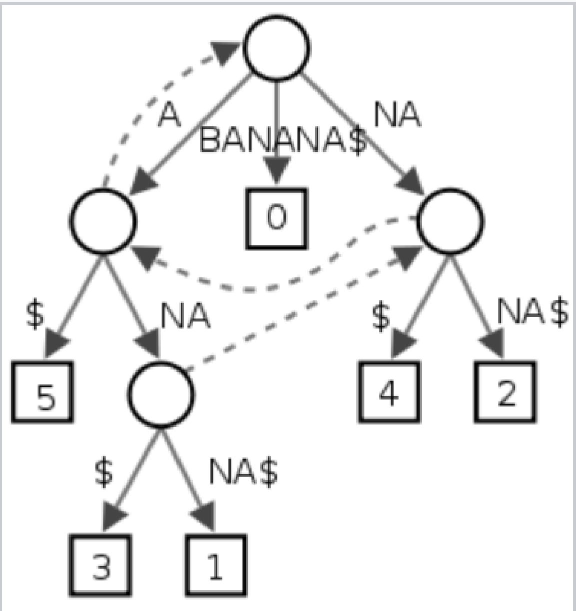
M yr

- Generalize to pattern

Mozart → (NAME)

1756 → (ANSWER)





Suffix tree for the text BANANA. Each substring is terminated with special character $. The six paths from the root to the leaves (shown as boxes) correspond to the six suffixes A$, NA$, ANA$, NANA$, ANANA$ and BANANA$. The numbers in the leaves give the start position of the corresponding suffix. Suffix links, drawn dashed, are used during construction.

Learning patterns (III)

- Repeat for different examples of same question type

Mozart
Gandhi
Trump

For BIRTHDATE

a. born in <ANSWER> , <NAME>
b. <NAME> was born on <ANSWER> ,
c. <NAME> (<ANSWER> -
d. <NAME> (<ANSWER -)
...

living ppl / date of death not known

Quality of patterns

- Need to evaluate pattern quality before being used for answering
- Repeat earlier pattern extraction process but without answer term
- Search for pattern in extracted sentences
- Calculate the precision of pattern
- Retain only patterns with enough support
- Use of precision guided by MLE

$P = C_a / C_o$

ans others

1756 + wrong answers

Mozart was born in <ANY_WORD>

Mozart was born in 1756

Answering with templates

- Determine type of new question
- Identify question term

existing systems

- Create query
- Preprocess documents as before
- Replace question term with <NAME>
- Match patterns from table
- Retrieve words that match <ANSWER>
- Rank with pattern precision

ans ranking

Table
P_1
P_2 → match in text
P_3

match
→ <NAME> + <ANS>

Evaluation with TREC data

- TREC-10 (2001) data
- More info: https://trec.nist.gov/data/qa.html

TREC QA

- Questions:

  https://trec.nist.gov/data/qa/2001_qadata/main_task_QAdata/qa_main.894-1393.txt

MRR: 1/2

$\frac{1}{\text{rank}}$

TREC Corpus

Question type	Number of questions	MRR on TREC docs
BIRTHYEAR	8	0.48
INVENTOR	6	0.17
DISCOVERER	4	0.13
DEFINITION	102	0.34
WHY-FAMOUS	3	0.33
LOCATION	16	0.75

TREC corpus

Web

Question type	Number of questions	MRR on the Web
BIRTHYEAR	8	0.69
INVENTOR	6	0.58
DISCOVERER	4	0.88
DEFINITION	102	0.39
WHY-FAMOUS	3	0.00
LOCATION	16	0.86

Web

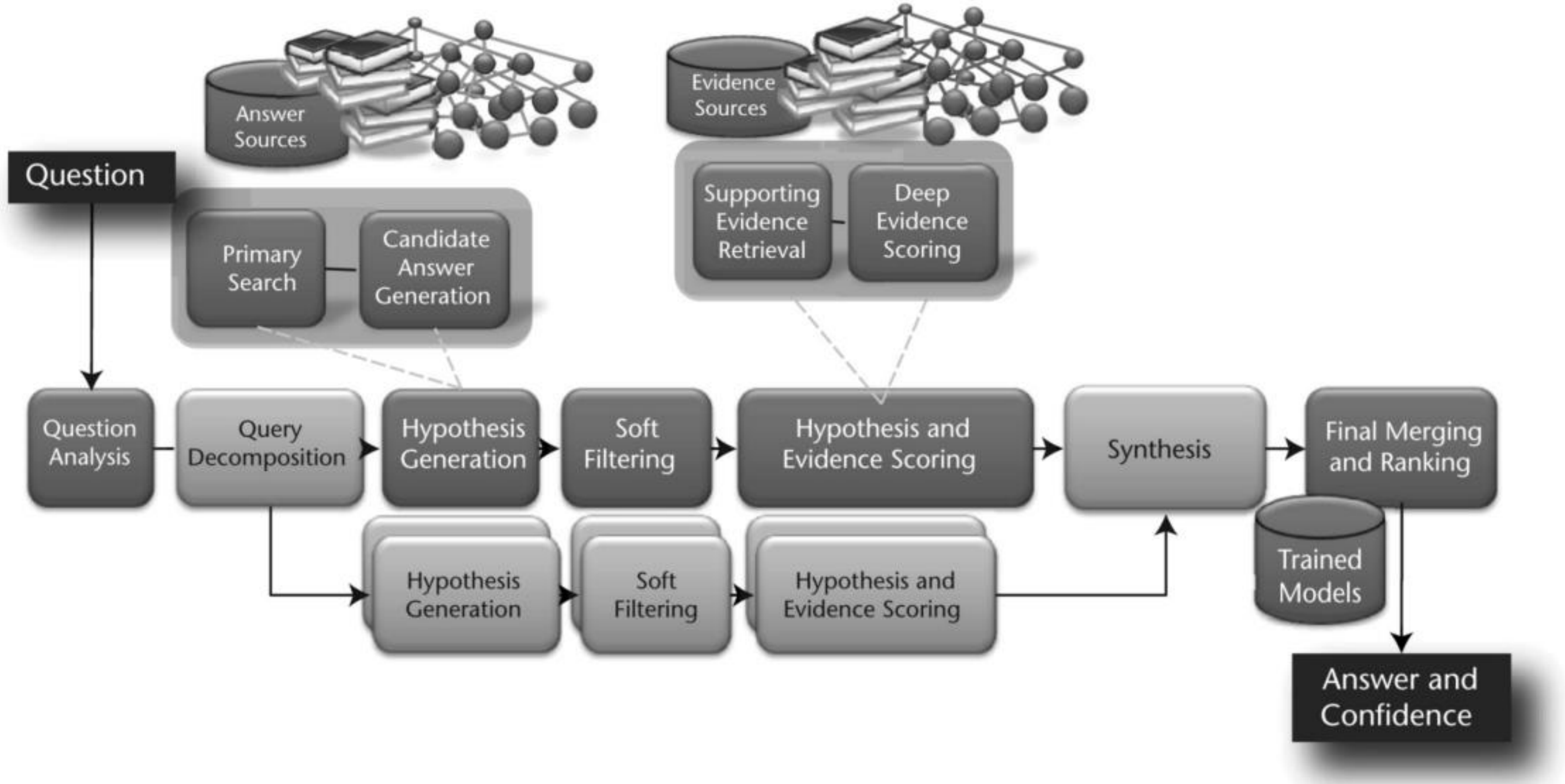
higher

Sys 1

Sys 2

The IBM Watson System

The IBM Watson System



[Building Watson: An overview of the DeepQA project](https://www.aaai.org/ojs/index.php/aimagazine/article/view/2303)
https://www.aaai.org/ojs/index.php/aimagazine/article/view/2303

Conclusions

- Templates are a powerful method for question-answering
- Applicability may extend beyond QA
- Limited by coverage
- Learning via similarity and feedback enables handling diversity