# When a Knowledge Base Is Not Enough- Question Answering over Knowledge Bases with External Text Data

Date: 2016/11/08
Author: Denis Savenkov, Eugene Agichtein
Source: ACM SIGIR' 16
Advisor: Jia-ling Koh
Speaker : Yi-hui Lee

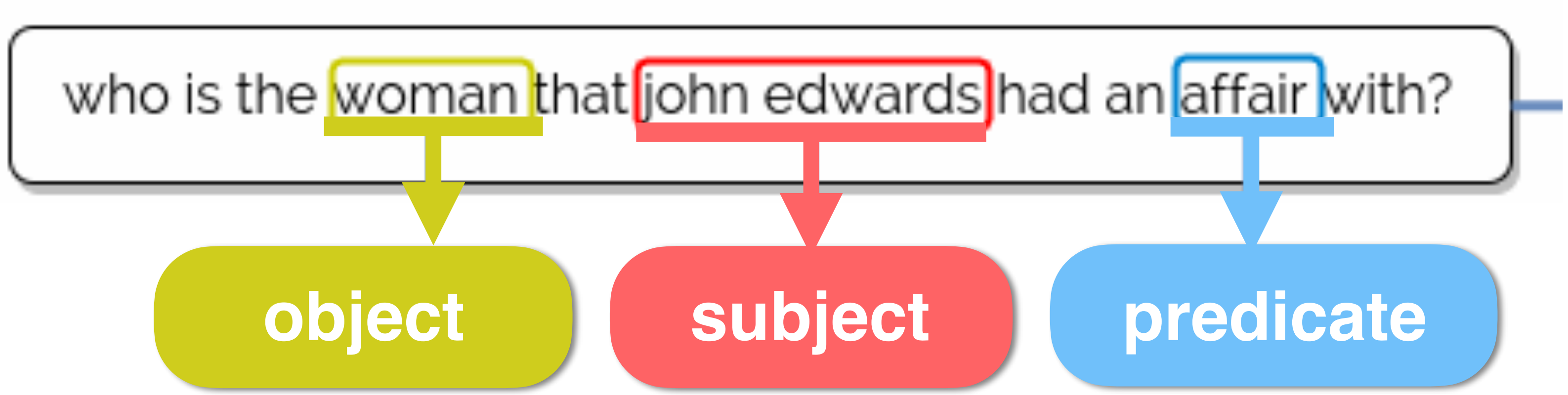
# Outline

- **Introduction**
- Approach
- Experiment
- Conclusion

# Introduction

- Question Answering:
  - Text-centric, or Text-QA: use text document collections to retrieve passages relevant to a question and extract candidate answers
  - Knowledge base-centric, or KBQA

    -RDF triples [subject, predicate, object]



# Introduction(cont.)

**Aqqu KBQA system**

Query Template

```
SELECT DISTINCT ?a_entity {
   <q_entity> <predicate> ?a_entity .
}
```

```
SELECT DISTINCT ?a_entity {
   <q_entity> <predicate_1> ?cvt_node .
   ?cvt_node <predicate_2> ?a_entity .
}
```

```
SELECT DISTINCT ?a_entity {
   <q_entity_1> <predicate_1> ?cvt_node .
   ?cvt_node <predicate_2> <q_entity_2> .
   ?cvt_node <predicate_3> ?a_entity .
}
```

question entity

mediator node

answer entity

Extension

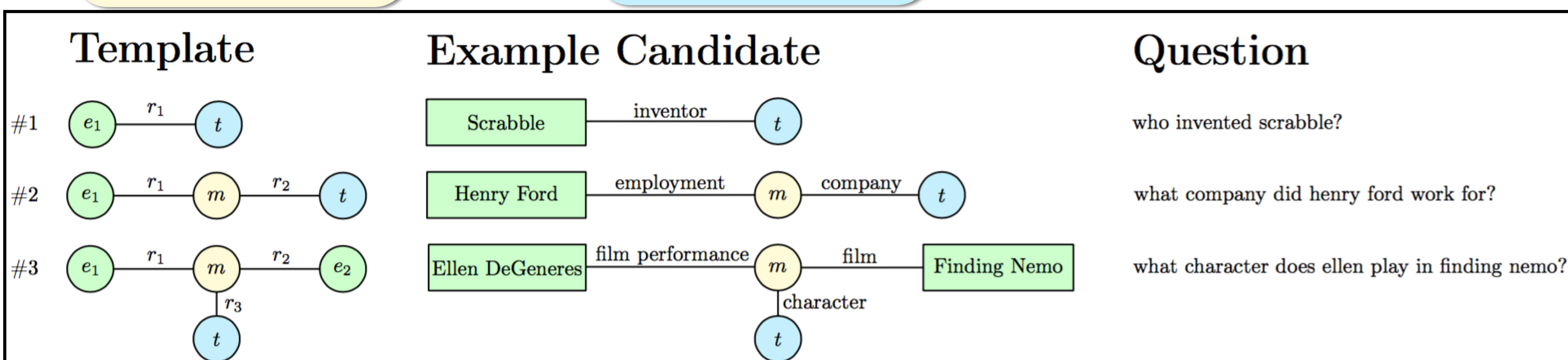
**Basic system extensions**

Question

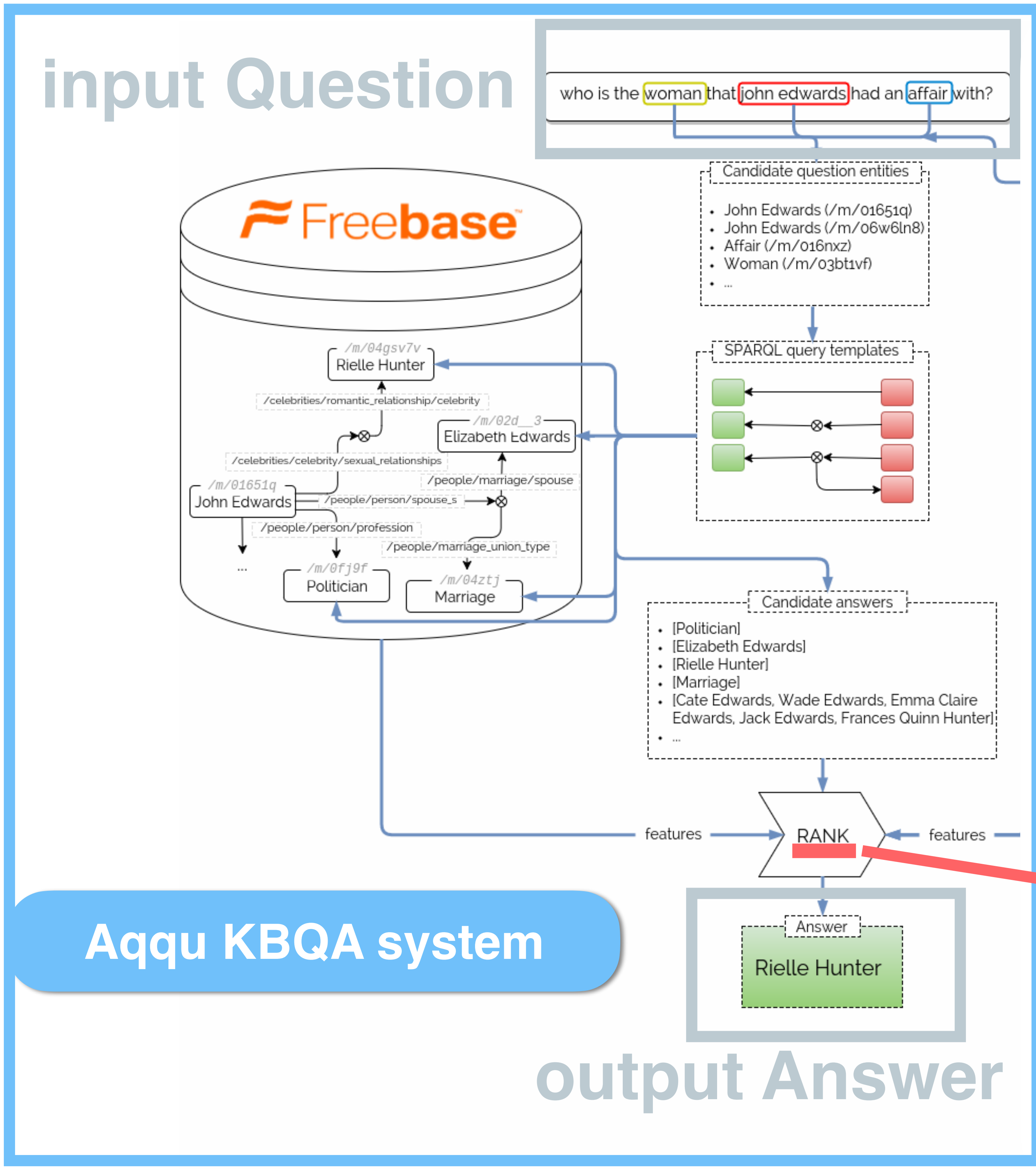
“what team did david beckham play for in 2011?”

Query Template

```
SELECT DISTINCT ?a_entity {
   <q_entity_1> <predicate_1> ?cvt_node .
   ?cvt_node <from_predicate> ?date_from .
   ?cvt_node <to_predicate> ?date_to .
   ?cvt_node <predicate_2> ?a_entity .
   FILTER ( <question_date> >= ?date_from AND
            <question_date> <= ?date_to )
```



# Introduction(cont.)




## Query Template

```
SELECT DISTINCT ?a_entity {
    <q_entity> <predicate> ?a_entity .
}
```

```
SELECT DISTINCT ?a_entity {
    <q_entity> <predicate_1> ?cvt_node .
    ?cvt_node <predicate_2> ?a_entity .
}
```

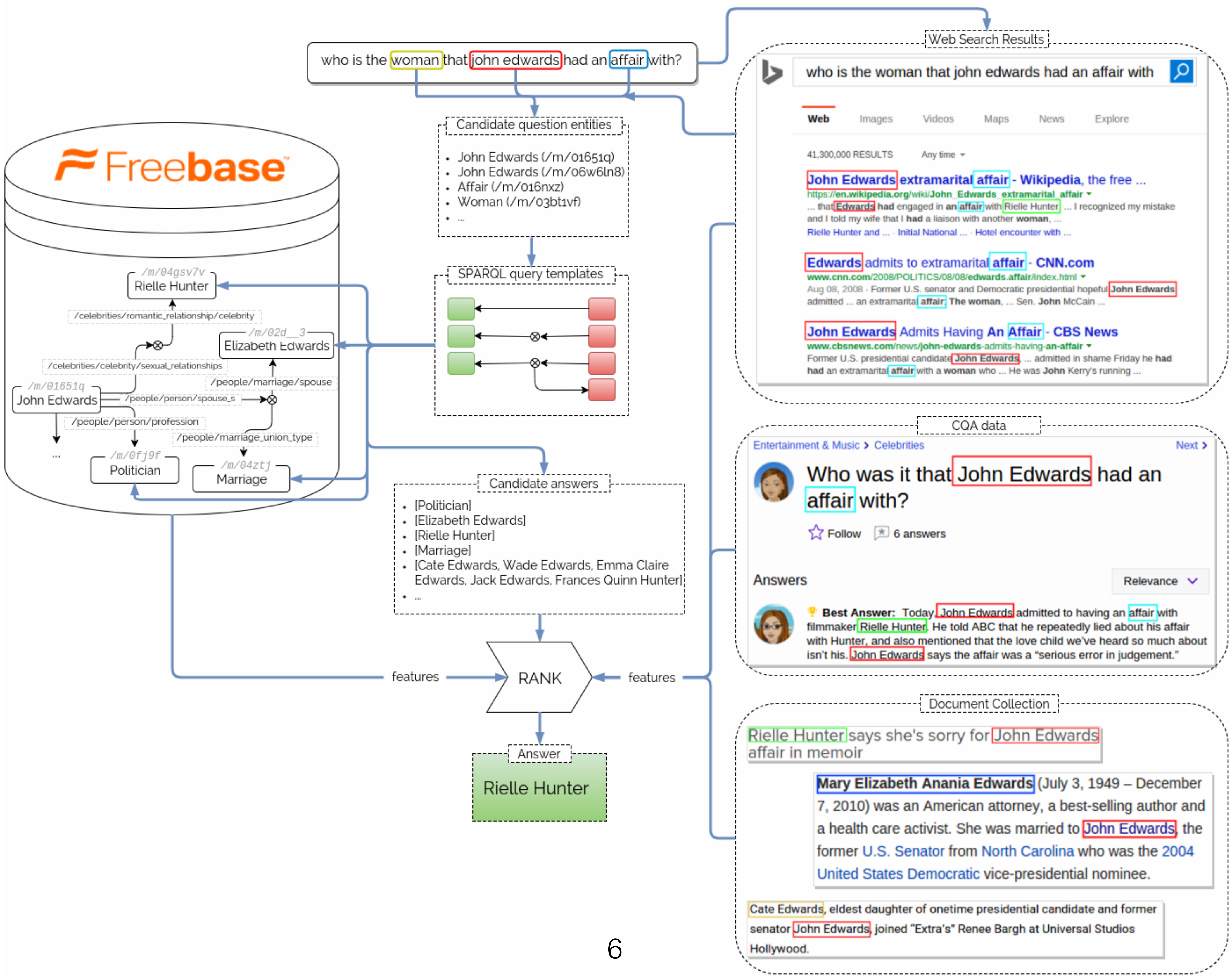
```
SELECT DISTINCT ?a_entity {
    <q_entity_1> <predicate_1> ?cvt_node .
    ?cvt_node <predicate_2> <q_entity_2> .
    ?cvt_node <predicate_3> ?a_entity .
}
```

```
SELECT DISTINCT ?a_entity {
    <q_entity_1> <predicate_1> ?cvt_node .
    ?cvt_node <from_predicate> ?date_from .
    ?cvt_node <to_predicate> ?date_to .
    ?cvt_node <predicate_2> ?a_entity .
    FILTER ( <question_date> >= ?date_from AND
             <question_date> <= ?date_to )
}
```

**pairwise leaning to rank using Random Forest Model**

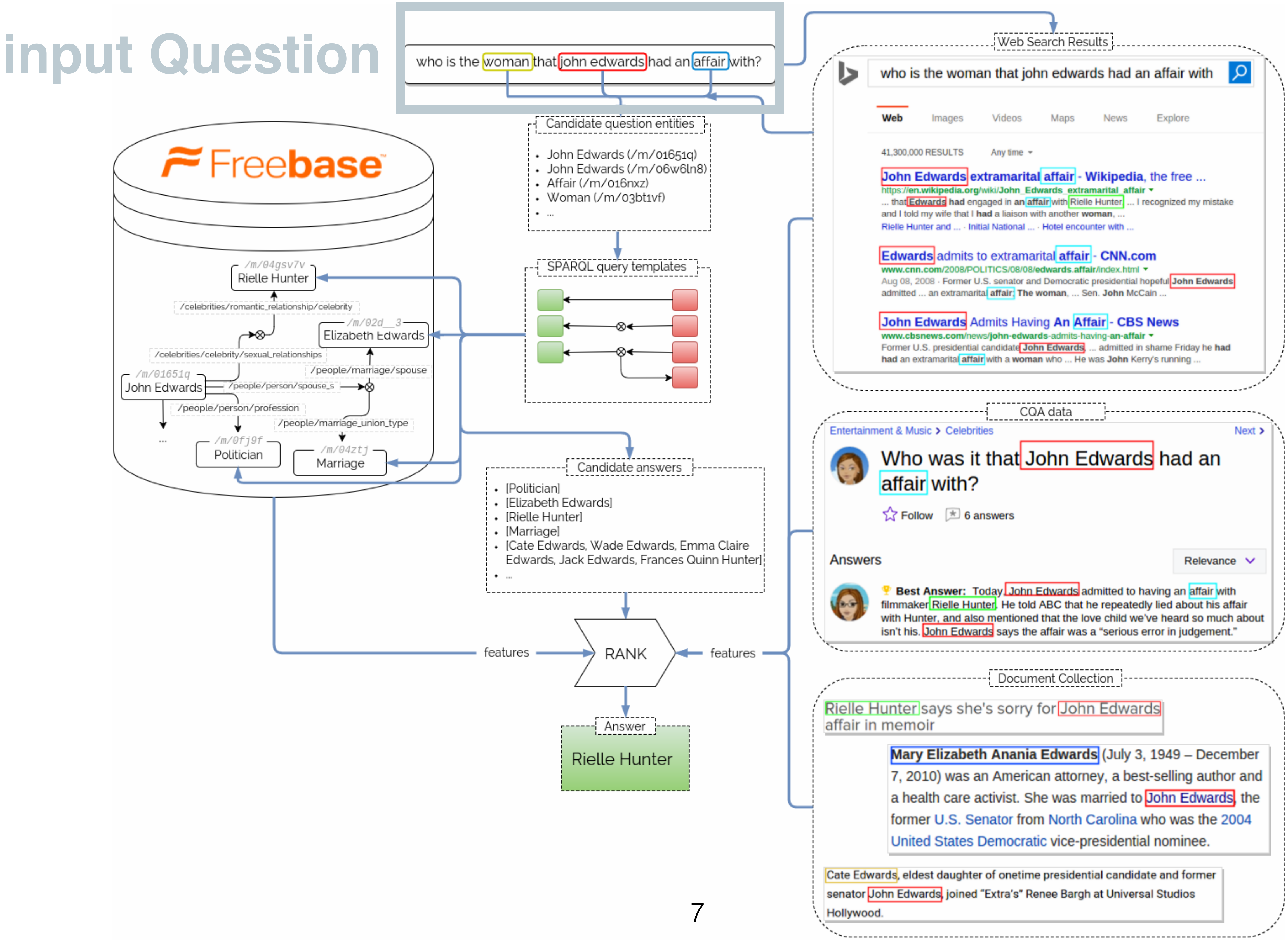
# Introduction(cont.)

- Text2KB Question Answering system's architecture



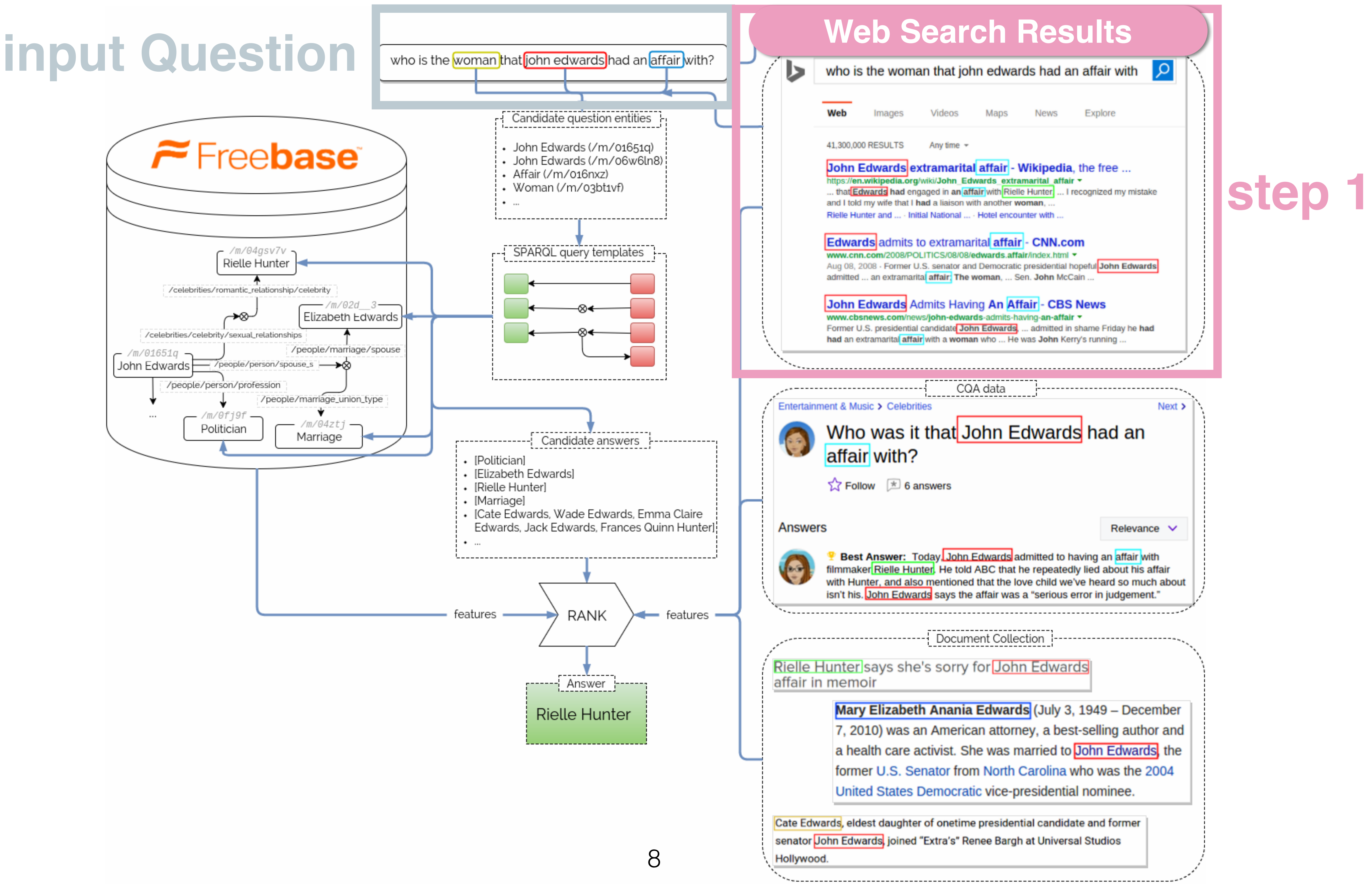
# Introduction(cont.)

- Text2KB Question Answering system's architecture



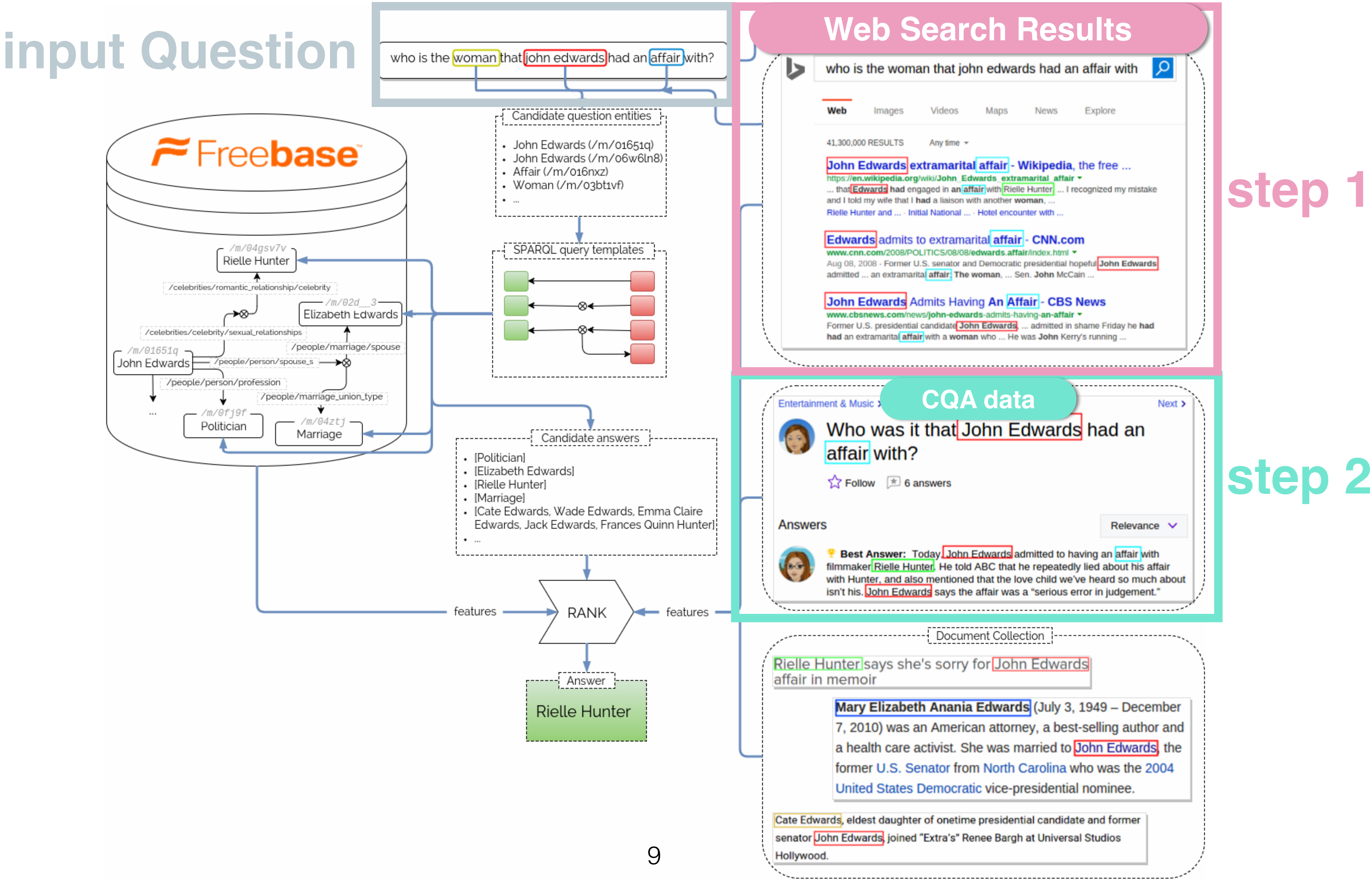
# Introduction(cont.)

- Text2KB Question Answering system's architecture



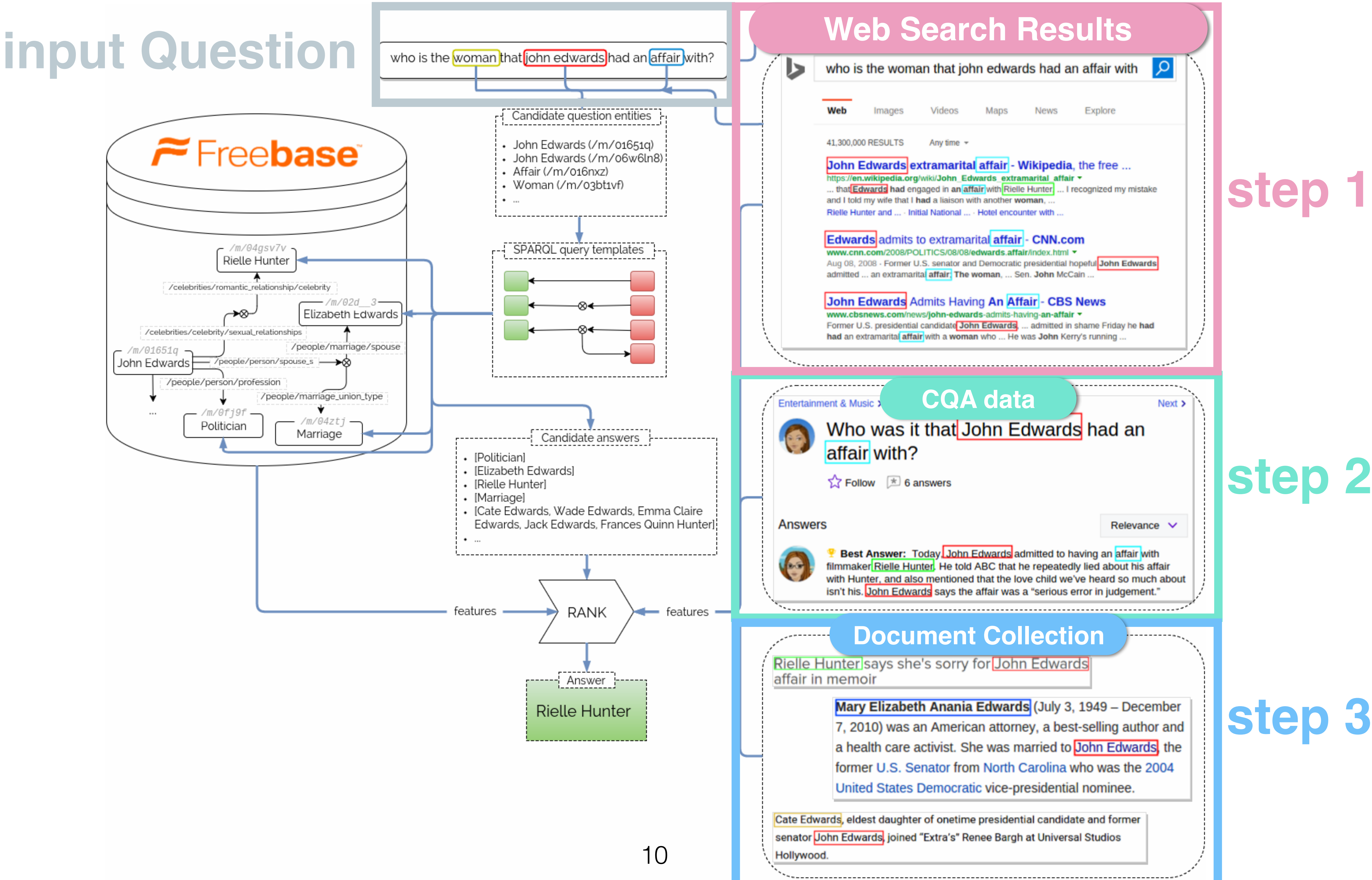
# Introduction(cont.)

- Text2KB Question Answering system's architecture



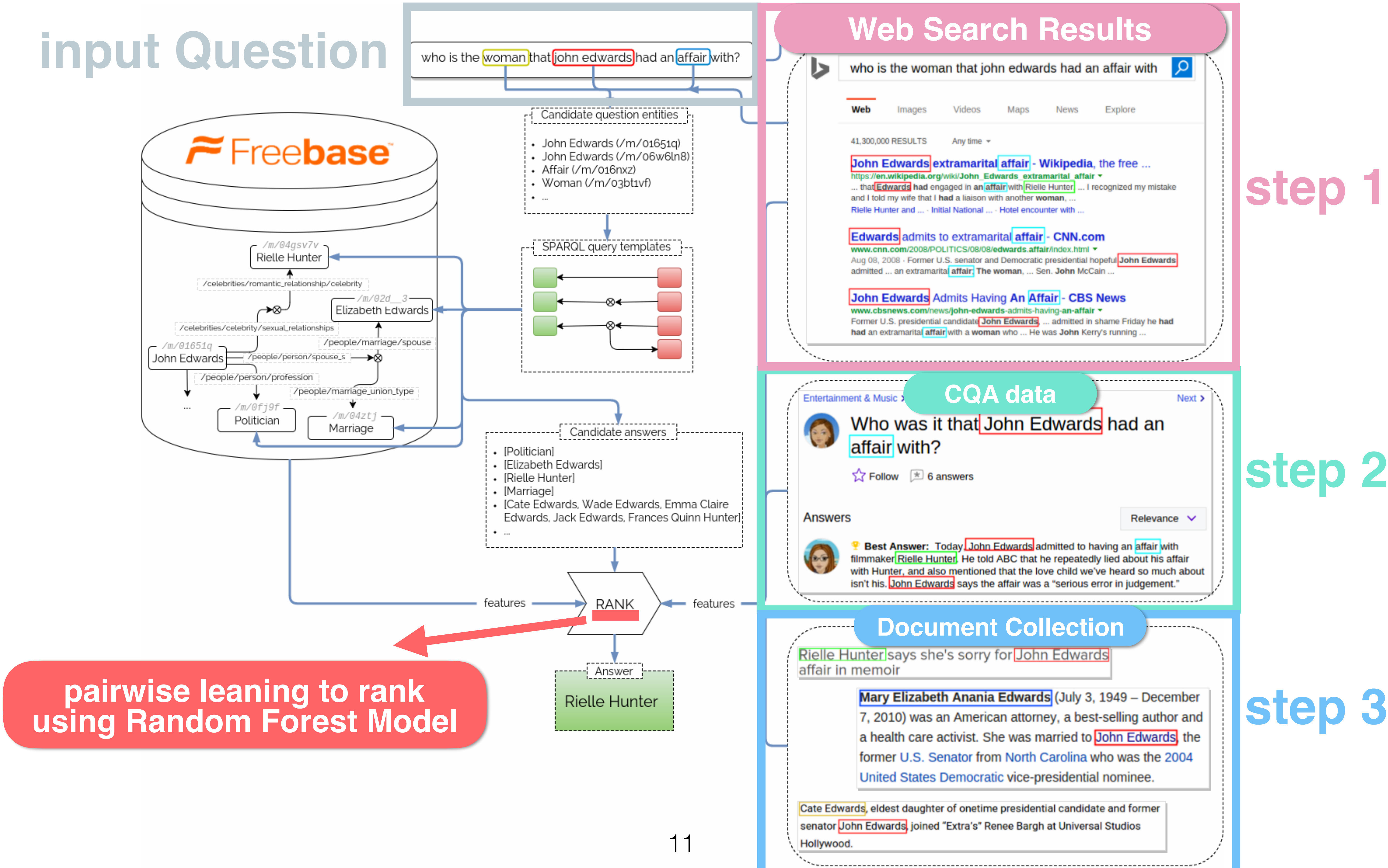
# Introduction(cont.)

- Text2KB Question Answering system's architecture



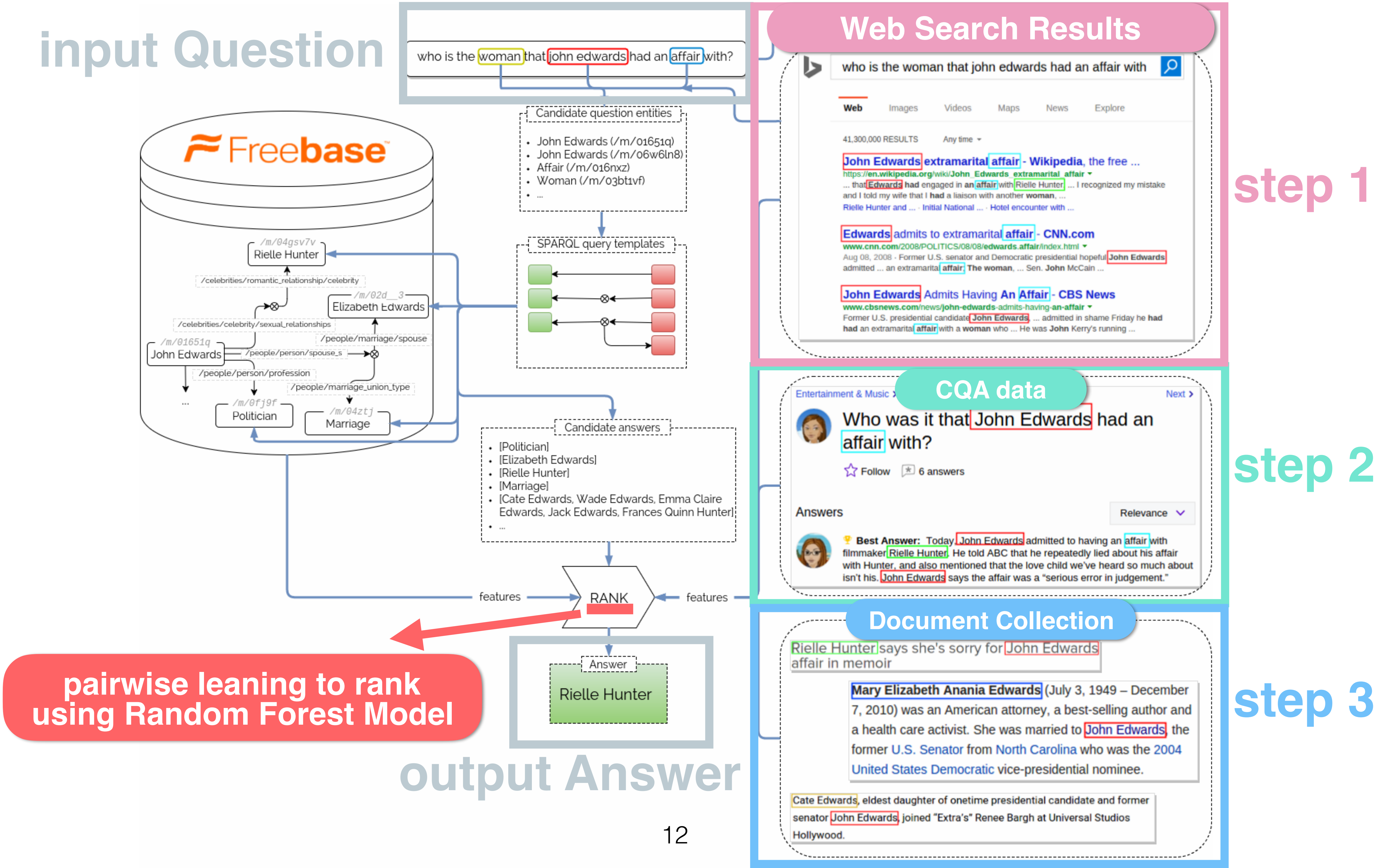
# Introduction(cont.)

- Text2KB Question Answering system's architecture



# Introduction(cont.)

- Text2KB Question Answering system's architecture

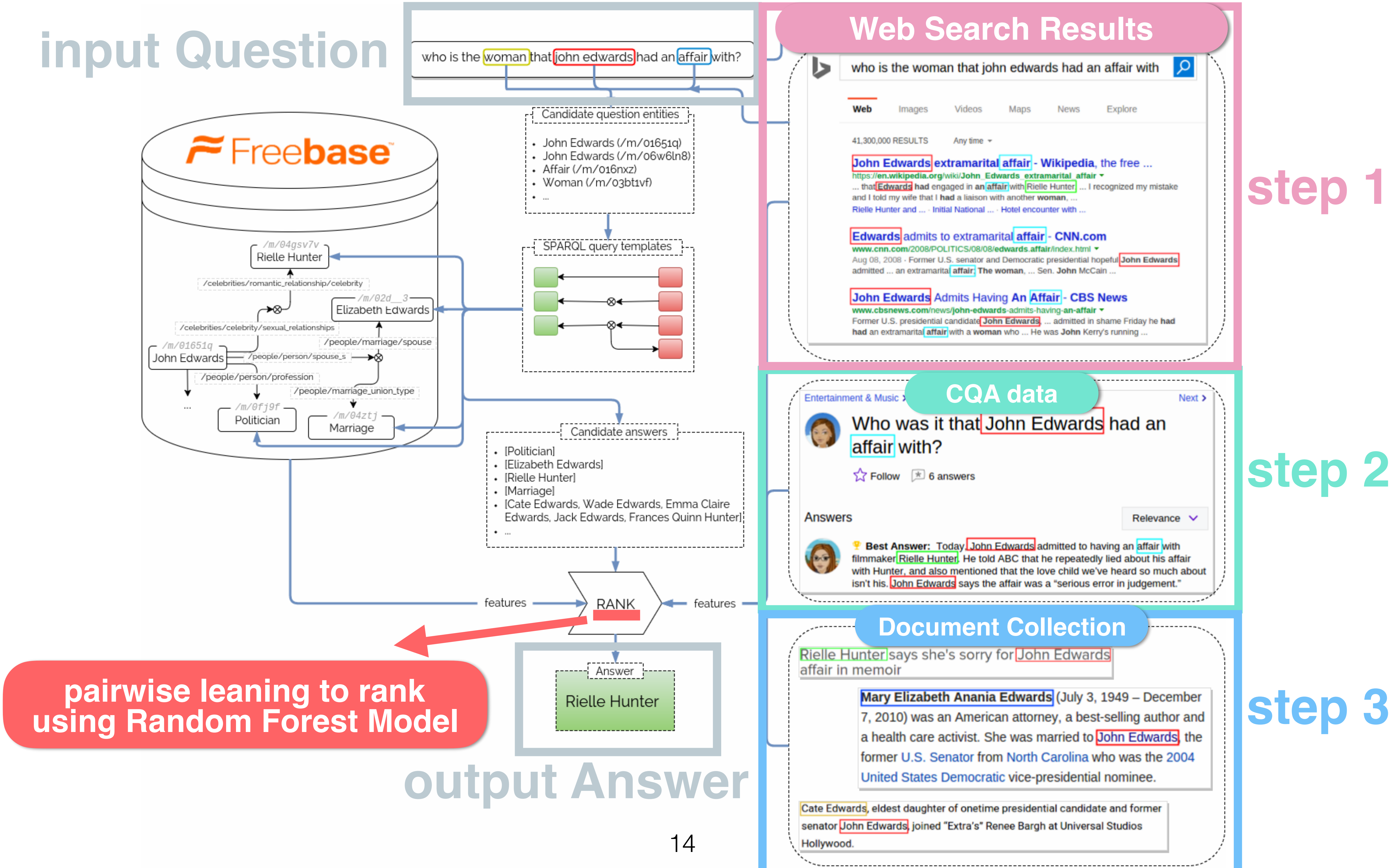


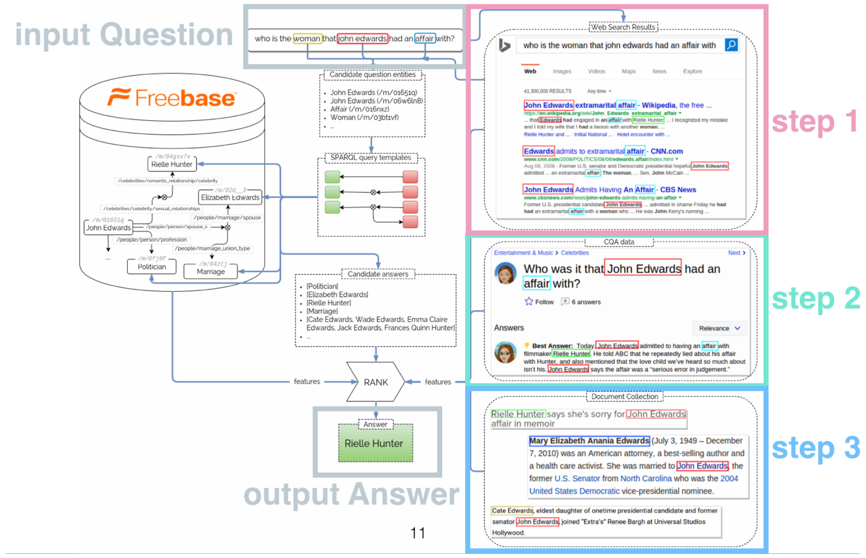
# Outline

- Introduction
- **Approach**
- Experiment
- Conclusion

# Approach

- Text2KB Question Answering system's architecture

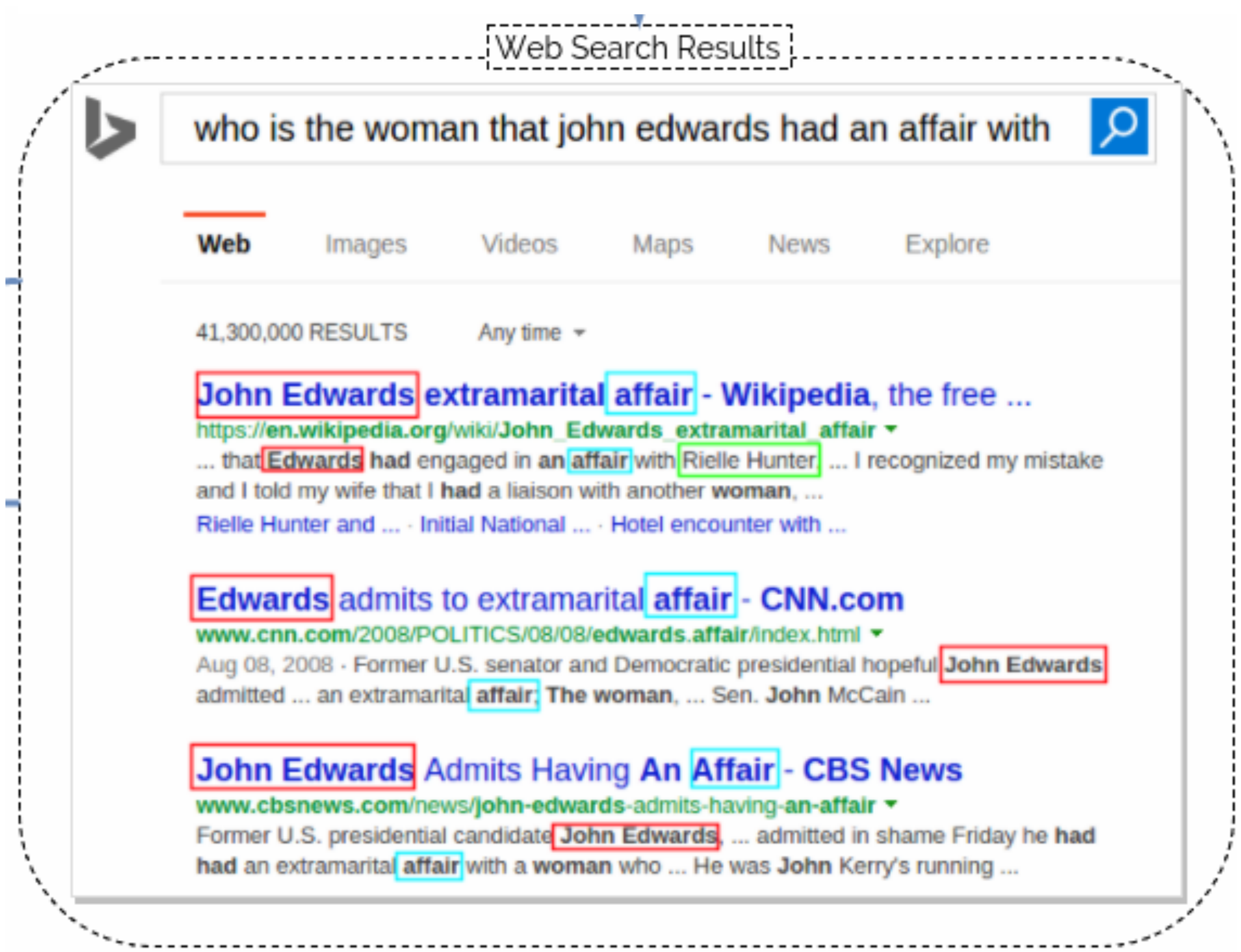





# Approach(cont.)

- Web search results for KBQA:
  - Question entity identification:

$$\max_{e_t \in M \backslash Stop, q_t \in Q \backslash Stop} 1 - \boxed{dist}(e_t, q_t) \geq 0.8$$




**Jaro-Winkler string distance**

The Jaro distance $d_j$ of two given strings $s_1$ and $s_2$ is

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}$$

Where:

- $m$ is the number of *matching characters* (see below);
- $t$ is half the number of *transpositions* (see below).

$$d_w = d_j + (\ell p(1 - d_j))$$

Given the strings $s_1$ *MARTHA* and $s_2$ *MARHTA* we find:

- $m = 6$
- $|s_1| = 6$
- $|s_2| = 6$
- There are mismatched characters T/H and H/T leading to $t = \frac{2}{2} = 1$

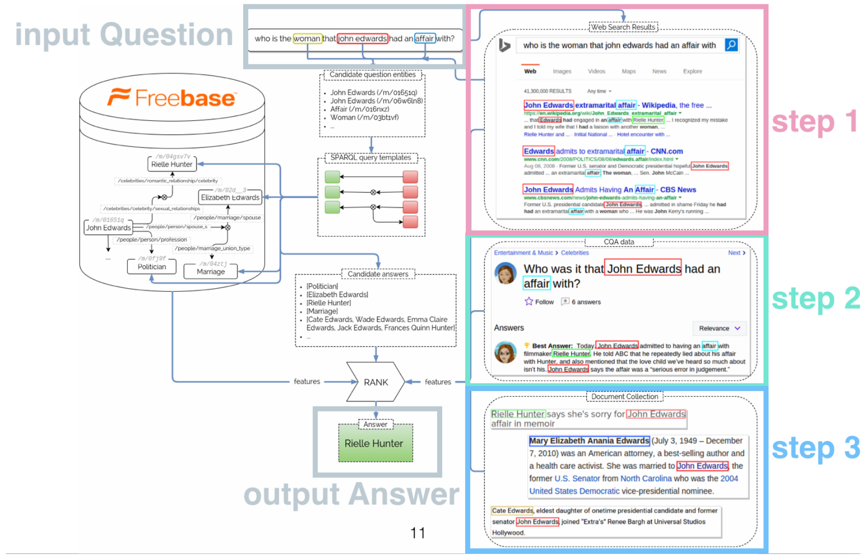
We find a Jaro score of:

$$d_j = \frac{1}{3}\left(\frac{6}{6} + \frac{6}{6} + \frac{6-1}{6}\right) = 0.944$$

$\ell = 3$

Thus:

$$d_w = 0.944 + (3 * 0.1(1 - 0.944)) = 0.961$$




# Approach(cont.)


- Web search results for KBQA:
  - Answer candidate features:

    Step 1. Precompute term and entity IDF scores

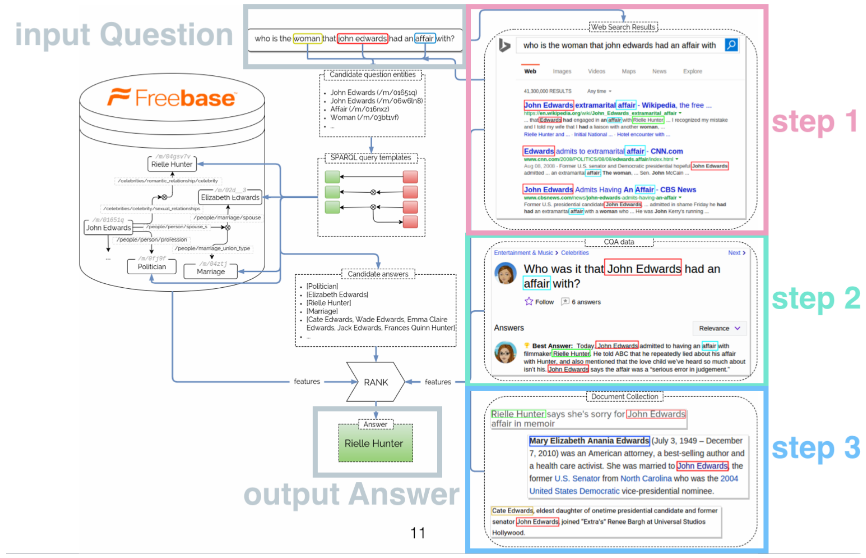
    Step 2. Snippet and Document represent by TF-IDF vectors

    Step 3. Combined token and entity vectors

    Step 4. Answer candidate represent by TF-IDF vectors as well

    Step 5. Cosine similarities between answer and each of 10 snippet and document. Using average score and maximum score as features

    Step 6. Compute Answer similarities as well




# Approach(cont.)


- Web search results for KBQA:
  - Answer candidate features:

    Step 1. John Edwards: 0.7, affair: 0.3, Rielle Hunter: 0.7, …

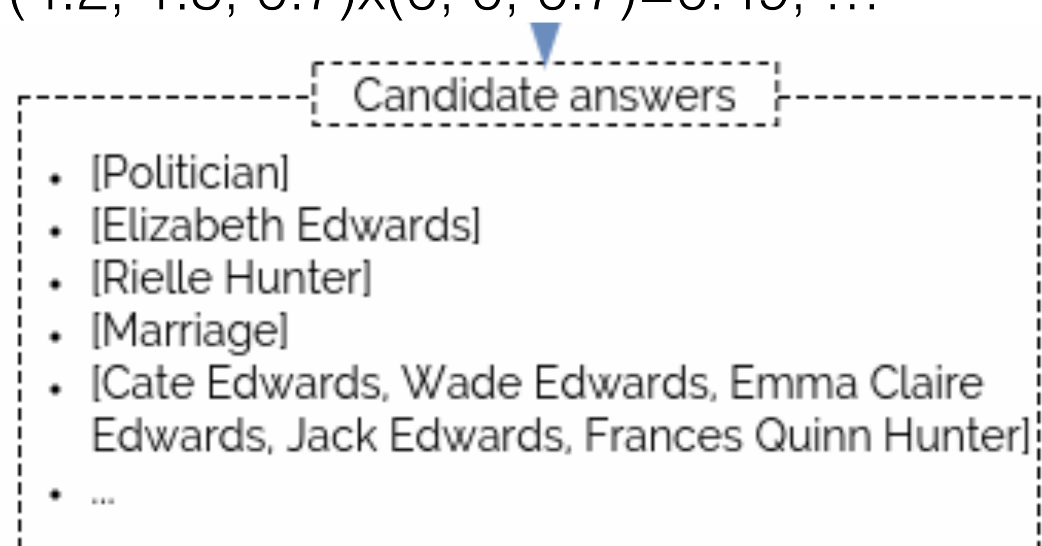
    Step 2. (6*0.7, 5*0.3, 1*0.7)=(4.2, 1.5, 0.7), …

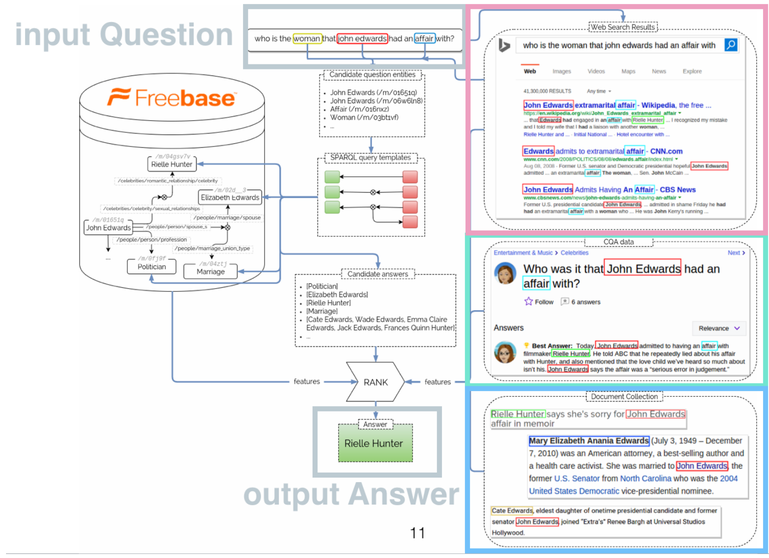
    Step 3. Combined token and entity vectors

    Step 4. [Politician](0, 0, 0), [Elizabeth Edwards](0, 0, 0), [Rielle Hunter](0, 0, 0.7), …

    Step 5. Cosine similarities between answer and snippet, (4.2, 1.5, 0.7)x(0, 0, 0.7)=0.49, …

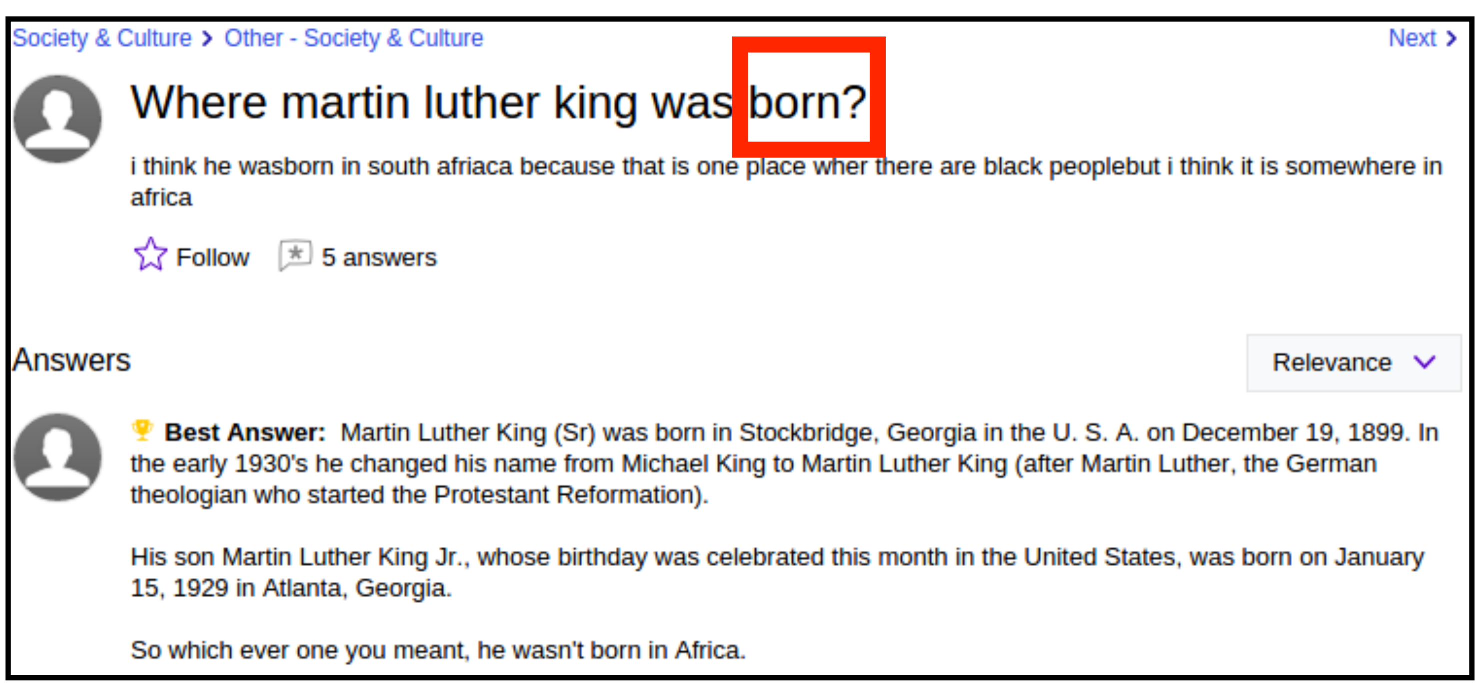
    Step 6. Features: 0.49, …



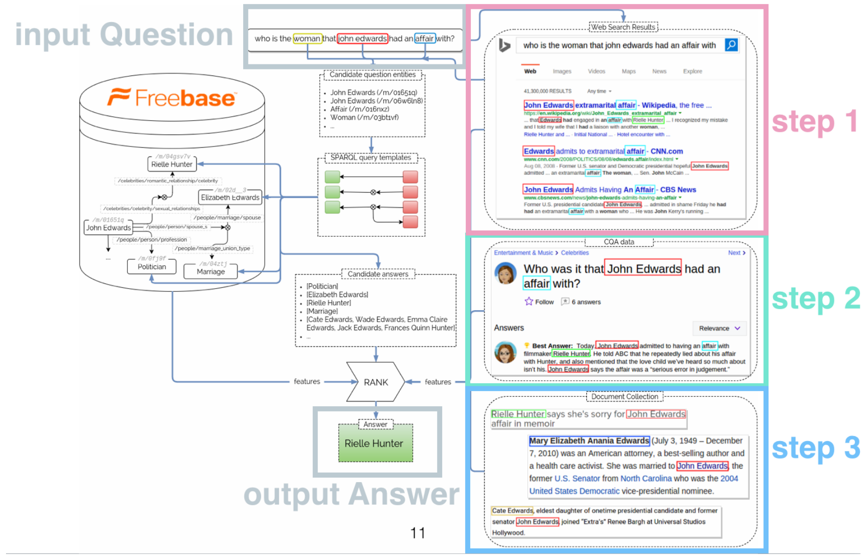



# Approach(cont.)

- CQA data for Matching Questions to Predicates:
  - Dataset: Yahoo! WebScope L6 dataset, question and answer texts were run through an entity linker.
  - Label question-answer pair with predicates between entities mentioned in the question and in the answer.
  - PMI $$\mathrm{pmi}(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}$$



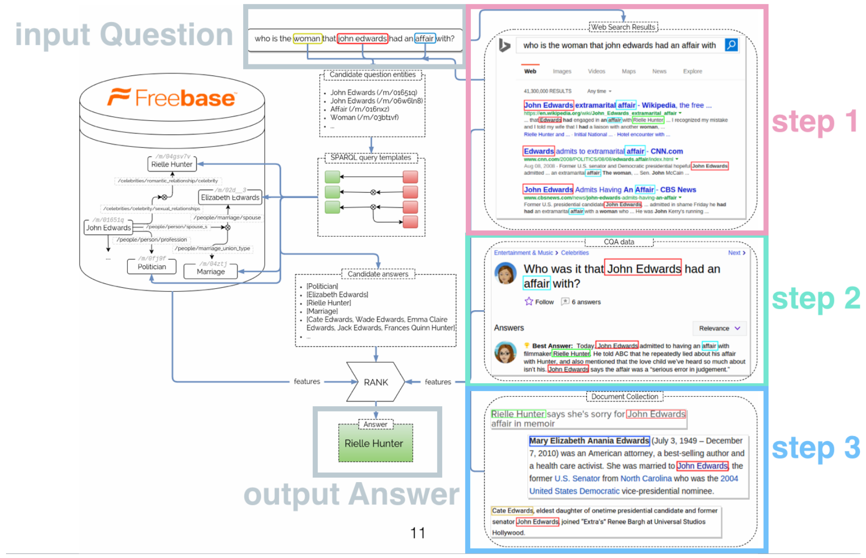

| Predicate | Term |
|---|---|
| people.person.date_of_birth<br>people.person.date_of_death<br>location.location.people_born_here | born |




# Approach(cont.)

- CQA data for Matching Questions to Predicates:

| Term | Predicate | PMI score |
|---|---|---|
| born | people.person.date_of_birth<br>people.person.date_of_death<br>location.location.people_born_here | 3.67<br>2.73<br>1.60 |
| kill | people.deceased_person.cause_of_death<br>book.book.characters | 1.70<br>1.55 |
| currency | location.country.currency_formerly_used<br>location.country.currency_used | 5.55<br>3.54 |
| school | education.school.school_district<br>people.education.institution<br>sports.school_sports_team.school | 4.14<br>1.70<br>1.69 |
| win | sports.sports_team.championships<br>sports.sports_league.championship | 4.11<br>3.79 |




# Approach(cont.)

- Estimating Entity Associations: question and answer entities are likely to be mentioned together
  - Ranking candidate answers through textual data(<u>ClueWeb12 corpus</u>)
  - Language Model score: $p(Q|e_1, e_2) = \prod_{t \in Q} p(t|e_1, e_2)$

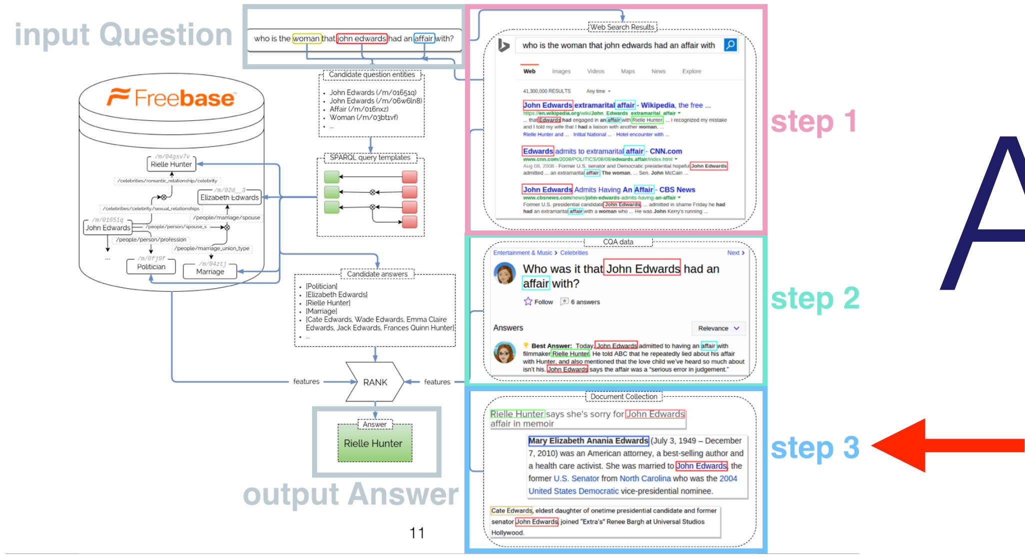
term

question entity

answer entities

| Entity 1 | Entity 2 | Term counts |
|---|---|---|
| John Edwards | Rielle Hunter | campaign, affair, mistress, child, former ... |
| John Edwards | Cate Edwards | daughter, former, senator, courthouse, greensboro, eldest ... |
| John Edwards | Elizabeth Edwards | wife, hunter, campaign, affair, cancer, rielle, husband ... |
| John Edwards | Frances Quinn | daughter, john, rielle, father, child, former, paternity... |

use the minimum, average, maximum score over all answer entities as features




# Approach(cont.)

- Pairwise learning to rank model:

Candidate answers

- [Politician]
- [Elizabeth Edwards]
- [Rielle Hunter]
- [Marriage]
- [Cate Edwards, Wade Edwards, Emma Claire Edwards, Jack Edwards, Frances Quinn Hunter]
- ...

(Politician, Elizabeth Edwards)-> +1(Politician is better than Elizabeth Edwards)
(Politician, Rielle Hunter)-> -1(Rielle Hunter is better than Politician)
(Politician, Marriage)-> +1(Politician is better than Marriage)
……

- Classifier: Random Forest Model

# Outline

- Introduction
- Approach
- **Experiment**
- Conclusion

# Experiment

- Methods Compare:
  - Aqqu
  - Text2KB(Web search): Bing search
  - Text2KB(Wikipedia search)
  - STAGG: The current highest performing KBQA system as measured on the WebQuestion dataset.

# Experiment(cont.)

- Datasets: standard evaluation procedure for the WebQuestions dataset
  - The original 70-30% train- test split (3,778 training and 2,032 test instances).
  - Within the training split, 10% was set aside for validation.

# Experiment(cont.)

- Evaluation Metrics
  - WebQuestions dataset have primarily used the average F1-score as the main evaluation metric

$$\boxed{avg\ F1 = \frac{1}{|Q|}\sum_{q \in Q} f1(a_q^*, a_q)}$$

$$\boxed{f1(a_q^*, a_q) = 2\frac{precision(a_q^*, a_q) recall(a_q^*, a_q)}{precision(a_q^*, a_q) + recall(a_q^*, a_q)}}$$

$$\boxed{precision(a_q^*, a_q) = \frac{|a_q^* \cap a_q|}{|a_q|} \text{ and } recall(a_q^*, a_q) = \frac{|a_q^* \cap a_q|}{|a_q^*|}}$$

  - $a_q^*$: correct answers
  - $a_q$: given answers

# Experiment(cont.)

- Methods Compare:

| System | avg Recall | avg Precision | F1 of avg P and R | avg F1 |
|---|---|---|---|---|
| OpenQA [16] | - | - | - | 0.35 |
| YodaQA [4] | - | - | - | 0.343 |
| Jacana [30] | 0.458 | 0.517 | 0.486 | 0.330 |
| SemPre [6] | 0.413 | 0.480 | 0.444 | 0.357 |
| Subgraph Embeddings [10] | - | - | 0.432 | 0.392 |
| ParaSemPre [7] | 0.466 | 0.405 | 0.433 | 0.399 |
| Kitt AI [28] | 0.545 | 0.526 | 0.535 | 0.443 |
| AgendaIL [8] | 0.557 | 0.505 | 0.530 | 0.497 |
| STAGG [31] | 0.607 | **0.528** | **0.565** | **0.525** |
| Aqqu (baseline) [3] | 0.604 | 0.498 | 0.546 | 0.494 |
| Text2KB (Wikipedia search) | **0.632*** (+4.6%) | 0.498 | 0.557* (+2.0%) | 0.514* (+4.0%) |
| Text2KB (Web search) | **0.635*** (+5.1%) | 0.506* (+1.6%) | **0.563*** (+3.1%) | **0.522*** (+5.7%) |

# Experiment(cont.)

- Datasource and Features Contribution:
  - T: notable type score model as a ranking feature
  - DF: date range filter-based query template
  - WebEnt: using web search result snippets for question entity identification
  - WikiEnt: using wikipedia search result snippets for question entity identification
  - Web: using web search results for feature generation
  - Wiki: using wikipedia search results for feature generation
  - CQA: using CQA-based [question term, KB predicate] PMI scores for feature generation
  - CW: features, computed from entity pairs language model, estimated on ClueWeb

# Experiment(cont.)

| System | R | P | F1 |
|---|---|---|---|
| Aqqu | 0.604 | 0.498 | 0.494 |
| Text2KB (base) = Aqqu+DF+T | 0.617 | 0.481 | 0.499 |
| +Wiki+CQA+CL | 0.623 | 0.487 | 0.506 |
| +WikiEnt +Wiki+CQA+CL | 0.632 | 0.498 | 0.514 |
| +WebEnt | 0.627 | 0.492 | 0.508 |
| +Web+CQA+CL | 0.634 | 0.497 | 0.514 |
| +WebEnt +Web+CQA+CL | 0.635 | 0.506 | 0.522 |

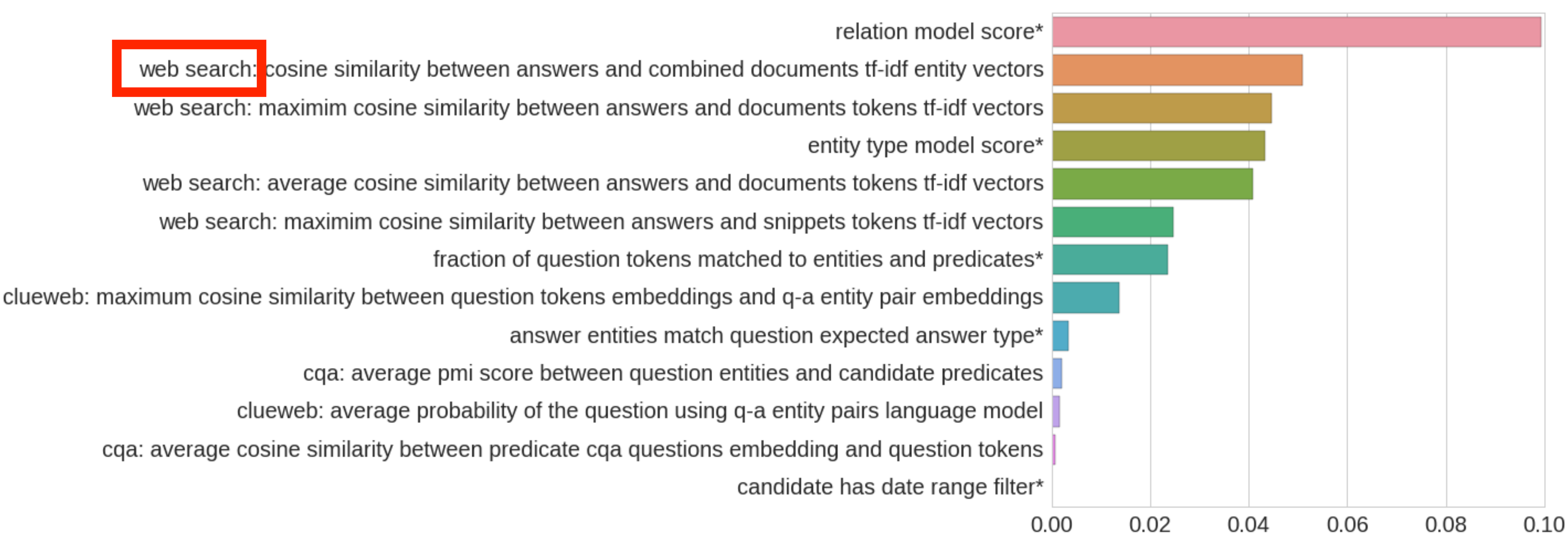
Average Recall (R), Precision (P), and F1 of Aqqu and Text2KB system with and without different components.

| System | R | P | F1 |
|---|---|---|---|
| Text2KB (Web search) | 0.635 | 0.506 | 0.522 |
| Text2KB -Web | 0.633 | 0.496 | 0.513 |
| Text2KB -CQA | 0.642 | 0.499 | 0.519 |
| Text2KB -CL | 0.644 | 0.505 | 0.523 |
| Text2KB -CQA-CL | 0.642 | 0.503 | 0.522 |
| Text2KB -Web-CQA | 0.631 | 0.498 | 0.514 |
| Text2KB -Web-CL | 0.622 | 0.493 | 0.508 |

Average Recall (R), Precision (P), and F1 of Text2KB with and without features based on web search results, CQA data and ClueWeb collection.

# Experiment(cont.)

- Feature Importance for Ranking:




A plot of Gini importances of different features of our answer ranking random forest model (features marked * are not text-based and are provided for comparison)

# Outline

- Introduction
- Approach
- Experiment
- **Conclusion**

# Conclusion

- Unstructured text resources can be effectively utilized for knowledge base question answering.
- Three particular techniques as follows:
  - Web search results for query understanding and candidate ranking.
  - Community question answering data for candidate generation
  - Text fragments around entity pair mentions for ranking

# Conclusion(cont.)

- Future work:
  - Extend our work to the more open setup, similar to the benchmark QALD(Question Answering over Linked Data) hybrid task
  - Questions no longer have to be answered exclusively from the KB. This would require extending the described techniques, and creating new QA benchmarks.