11-791 Design and Engineering of Intelligent Information System Fall 2014
Project

Building a Pipeline for Biomedical Question Answering

Important dates

- **Hand out: November 13.**[a]

- **Milestone 2 (M2) : Snippets retrieval. Turn in: November 20.**

  - For this milestone, you only need to send us the URL of your project repository page (i.e.,`https://github.com/COURSENUM-ID/project-teamID` where 'COURSENUM' is the course designation number 11791/11693, and ID is your assigned team number). We will look into your Issues page and Wiki page, and expect you have created milestones and issues, reported the results of your components, and completed proposal.

[a]This version was built on November 13, 2014

Useful information

1. Please visit Piazza regularly to check if a newer version is published. We may have new versions or revised instruction at the begining of each milestone.

   We expect that most of the general communication between the instructor team and students will take place on Piazza `https://piazza.com/class/hyvsubeilei6dd`. For private questions, e.g., regarding grades, you may contact instructors by e-mail. Your friendly TAs are: Avner Maiberg (`amaiberg@andrew.cmu.edu`), Parag Argawal (`paraga@andrew.cmu.edu`), Leonid (Leo) Boytsov (`srchvrs@cmu.edu`), and Xuezi (Manfred) Zhang (`xueziz@andrew.cmu.edu`).

2. Again, both source files and pdf file of this assignment are publicly available on my GitHub

   `http://github.com/amaiberg/software-engineering-preliminary`

   Please feel free to fork the project and send a pull request back to me as some of you did for Homework 0 for any error. Or you can just report an issue at

   `http://github.com/amaiberg/software-engineering-preliminary/isues`

Milestone 2

Creating your first BioASQ pipeline

In this task you will implement components to retrieve document snippets for a biomedical question answering pipeline (based on BioASQ's Task1b Phase A). You will create your own pipeline based on a provided archetype framework from M1. Note: the **only requirement** is that you use the typesystem provided in the archetype. You are otherwise permitted and encouraged to use any additional types and source code you may wish.

M 2.1 Types and evaluation metrics

The following are the output types described in the official BioASQ evaluation documentation [1]:

Snippets

$s_{i,1}, s_{i,2}, s_{i,3}, \ldots$ from the returned articles. Again, the list should be ordered by decreasing confidence. A single snippet list will be returned per question and participant, and the list may contain any number (or no) snippets from any of the returned articles $d_{i,1}, d_{i,2}, d_{i,3}, \ldots, d_{i,k}$. Each snippet will be represented by the unique identifier of the article it comes from and the offsets (character positions in the article) of the snippets beginning and end (offsets of the first and last characters).

Example: Here the JSON array `snippets` corresponds to document snippets.

```
"body": "What is the role of PrnP in mad cow disease?",
      "type": "factoid",
      "id": "160",
      "snippets" : [
                      {
                          "document": "http://www.ncbi.nlm.nih.gov/pubmed/23217568",
                          "beginSection": "sections.0",
                          "endSection": "sections.0",
                          "offsetInBeginSection": 591,
                          "offsetInEndSection": 678,
                          "text": "Our results show a high prevalence of RA ..."
                      },
```

```
...
{
        "beginSection": "sections.0",
        "document": "http://www.ncbi.nlm.nih.gov/pubmed/23217568",
        "endSection": "sections.0",
        "offsetInBeginSection": 1140,
        "offsetInEndSection": 1394,
        "text": "RA in LAC women is not only more common but presents..."
}]
```

Note that for each phase you will return a list of k items in order of decreasing confidence (where $k \leq 100$). See the evaluation section below for how these will be evaluated.

Evaluation

Please refer to table 2.1 to see the evaluation used for each type.

Retrieved items	Unordered retrieval measures	Ordered retrieval measures
snippets	mean percision, recall, F-measure	MAP,**GMAP**

Figure 2.1: Evaluation metrics

Also, please review the definitions of the evaluation metrics. Note that the definitions for Precision and Recall for document snippets are slightly different than the metrics you used in M1:

- Precision and Recall:

$$P_{snip} = \frac{|S \cap G|}{|S|} \tag{2.1}$$

$$R_{snip} = \frac{|S \cap G|}{|G|} \tag{2.2}$$

Where S is the set of all article-offset paris of all the characters in the snippet, and G is the set of all the article-offset pairs of all the characters in the golden snippet. See 2.2 for an illustration of an article-offset pair in an article.

- F-measure:

$$F = 2 \cdot \frac{P \cdot R}{P + R} \qquad \text{(Harmonic mean of Precision and Recall)}$$

- Average Precision (AP):

$$AP = \frac{\sum_{r=1}^{|L|} P(r) \cdot rel(r)}{L_R} \qquad \text{(see below)}$$

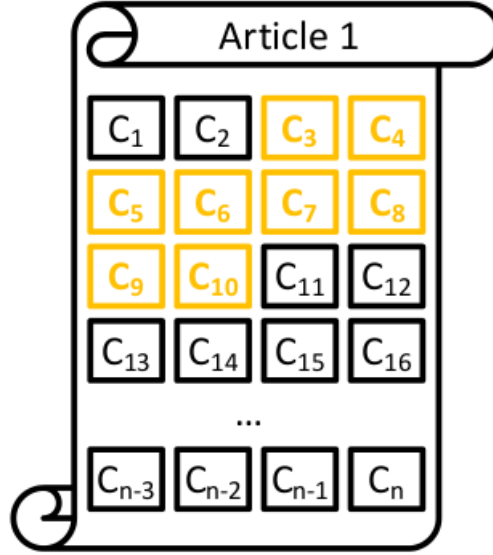Where, for any given query q_i and a golden set of items:

Figure 2.2: An example of a golden snippet starting at offset 3 and ending at offset 10.

- $|L|$ is the total number of items.
- $|L_R|$ is the number of relevant items.
- $P(r)$ is the precision for a list containing the first r items.
- $rel(r)$ is an indicator function for the existence of item r in the golden set (i.e., it returns 1 if the rth item is relevant, and 0 otherwise).

- Mean Average Precision (MAP):

$$MAP = \frac{1}{n} \sum_{i=1}^{n} AP_i$$

(To get the average precision for list of queries $q_1, q_2, \ldots, q_n$.)

- Geometric Mean Average Precision (GMAP):

$$GMAP = \sqrt[n]{\prod_{i=1}^{n} (AP_i + \epsilon)} \quad \text{(with some small } \epsilon \text{ for cases where } AP_i = 0)$$

GMAP is similar to MAP, only used to further penalize low performing queries.

M 2.2   Archetype

Typesystem

For this milestone, you are required to use the typesystem `OAQATypes.xml` included in the archetype. Note however, you are only required to use the types defined in 2.2 and their supertype ancestry. You are of course welcome to extend the typesystem and/or use any other preexisting types. Most importantly. it is imperative that you use the inherited `uri` and `rank` attributes, as we will use them for evaluation.

BioASQ Type	UIMA Type

Result (supertype)	• `edu.cmu.lti.oaqa.type.retrieval.SearchResult`
	   – `uri`
	   – `rank`

Snippet	• `edu.cmu.lti.oaqa.type.retrieval.Passage`
	   – `uri` (inherited)
	   – `rank` (inherited)

Figure 2.3: Typesystem Mapping

Web Services

Unlike in M1, there is no client for M2. To access the web service below you will simply query `http://metal.lti.cs.cmu.edu:30002/pmc/PMID` with the PMID you retrieved from the previous services.

- Full Document Sources: a service for accessing the *PMC* full text articles, with the same input parameters and output parameters as aforementioned, with the only difference being that the articles returned contain in addtion the full text.

Data

In the archetype you will also find `BioASQ-SampleData1B.json` containing an annotated version of the 29 sample questions shown in M0. We also provide you with a convenience class `JsonCollectionReaderHelper.java` to read from JSON format.

Bibliography

[1] Georgios Balikas, Ioannis Partalas, Aris Kosmopoulos, Sergios Petridis, Prodromos Malakasiotis, Ioannis Pavlopoulos, Ion Androutsopoulos, Nicolas Baskiotis, Eric Gaussier, Thierry Artieres, and Patrick Gallinari. Evaluation framework specifications. Project deliverable D4.1, 05/2013 2013.