[a] *TU Wien 1040, Vienna, Austria*

## Introduction

”This Document will serve as a blueprint which we need to follow in order to get this assignment done. The document is based on state of the art models of last year and explains how others achieved good scores.”[1]

## 1 Objectives & Official Evaluation

Table 1: BioASQ 13b sub-tasks, required outputs and metrics

| Sub-task | System output | Automatic metric(s)[2] |
|---|---|---|
| Documents | $\leq$ 10 PubMed/PMC IDs | MAP[3], GMAP[4] |
| Snippets | $\leq$ 10 evidence sentences | F-measure[5] |
| Exact answers | yes/no, factoid, list | Accuracy[6], MRR[7], macro-$F_1$[8] |
| Ideal answer | 1 paragraph summary | ROUGE-2, ROUGE-SU4[9] + manual score |

[1]Note: [Integer] is a clickable link that leads you to the source of the referenced data.; No Model is Perfect and this is no exception please take into account any BIAS in the Data Mining Proccess as well as Processing

[2]All metrics are produced by the official BioASQ scorer.

[3]**MAP** $= \frac{1}{Q}\sum_{q=1}^{Q} \mathrm{AP}_q$, where $\mathrm{AP}_q$ is the average precision for question $q$.

[4]**GMAP** $= \exp\big(\frac{1}{Q}\sum_q \ln(\mathrm{AP}_q + \varepsilon)\big)$, $\varepsilon = 10^{-6}$. The geometric mean punishes zero-AP queries more strongly.

[5]**F-measure** $= 2PR/(P+R)$, harmonic mean of precision $P$ and recall $R$.

[6]**Accuracy** $= \frac{\#\text{correct}}{\#\text{total}}$.

[7]**MRR** $= \frac{1}{Q}\sum_q \frac{1}{\text{rank}_q}$, reciprocal rank of the first correct entity.

[8]**Macro-$F_1$**: arithmetic mean of class-wise $F_1$ scores (treats each entity class equally).

[9]**ROUGE-n**: recall of matching $n$-grams; **ROUGE-SU4**: skip-bigrams with window $\leq$ 4 words, capturing looser paraphrases.

## 2 Protocol of Phase A Implementation

1. **Data acquisition**
   Downloaded the BioASQ 13b training file:
   - `Download Dataset here`(47 MB)
2. **JSON inspection & parsing fix**
   - Loaded the file in Python and discovered it's a `dict` with key `"questions"`.
   - Updated loader to:

   ```
   data = json.loads(...);
   qs   = data["questions"]
   ```
3. **Initial retrieval code**
   **Very Slow Approach due to API bottleneck:**
   Copied the session initialization from TUWEL and `find_pubmed_citations()` wrappers around the BioASQ PubMed API ($\approx$ 20 s/query).
4. **Faster Approach:**
   Switched to NCBI Entrez `esearch` via BioPython:
   - Direct access to NCBI's official E-utilities, which are engineered for high-throughput PubMed searches, reduces round-trip latency to 1 s/query.
   - Returns minimal XML payloads containing only PMID lists, cutting data-transfer and parsing overhead per request.

   This processes all 5 389 questions sequentially in $\leq$ 100 min.
5. **Full-set retrieval script**
   - Loop over all 5 389 questions, call `find_pubmed_citations(sess, q["body"], k=1000)`.
   - Throttled with `time.sleep(0.33)` to stay under $\approx 3req/s$.
   - Saved results in a form of list which contains the query_id and documents.
6. **Error handling & retries**

- Encountered HTTP timeouts on the legacy API.
- Wrapped each call in a 3-attempt retry loop with exponential back-off (2 s, 4 s, 8 s).
- Logged persistent failures to `phaseA_errors.log` and continued.

7. **Output generation**
   Produced `api_phaseA_run.json`, containing one entry per question:

```
[
  {
    "query_id": "BioASQ13b-dev-0001",
    "documents": ["10964970","29198224", . . .  up to 1000 IDs . . . ]
  },
  . . .
]
```

8. **Next planned step**
   Lastly ran the official scorer on the predictions made See the Official Test Documents here.
   Based on typical BM25 baseline runs over the BioASQ 13 Phase A development set—which achieve MAP ≈ 0.28 and GMAP ≈ 0.12[10]—we consider a Phase A MAP in the 0.23–0.26 range to be solid[11].

### *2.1. Testing for Phase A*

1. **Clone the evaluator repo**

```
git clone https://github.com/BioASQ/Evaluation-Measures.git
cd Evaluation-Measures                       # repo root
```

2. **Locate the pre-built QA scorer**
   The JAR you need is already compiled at `flat/BioASQEvaluation/dist/BioASQEvaluation.jar`.

[10]Baseline obtained by retrieving 1 000 PMIDs per question with Pyserini BM25 ($k_1$=0.9, b=0.4) on the 2024 dev set and scoring with the official BioASQ Phase A evaluator (BioASQ/Evaluation-Measures Task1b).

[11]The slight drop reflects the overhead and coverage differences when using the legacy PubMed API instead of a local index, which typically costs 0.02–0.05 in MAP.

3. **Run the scorer for Phase A**
   Replace the two placeholders in {...} with (i) the gold JSON (`training13b.json`) and (ii) the run file produced by your BM25/Entrez script:

```
java -Xmx2G -cp \
  /absolute/path/to/Evaluation-Measures/flat/BioASQEvaluation/
     dist/BioASQEvaluation.jar \
  evaluation.EvaluatorTask1b -phaseA \
  {/path/to/training13b.json} \
  {/path/to/your_run.json}
```

4. **Concrete example (my setup)**

```
java -Xmx2G -cp \
/Users/greinaldpappa/Downloads/BioASQ-training13b/Evaluation-
   Measures/flat/BioASQEvaluation/dist/BioASQEvaluation.jar \
evaluation.EvaluatorTask1b \
-phaseA \
/Users/greinaldpappa/Downloads/BioASQ-training13b/training13b.
   json \
/Users/greinaldpappa/Downloads/BioASQ-training13b/
   esearch_phaseA_run.json
```

5. **Concrete example (BM 25 Setup)**

```
java -Xmx2G -cp \
/Users/greinaldpappa/Downloads/BioASQ-training13b/Evaluation-
   Measures/flat/BioASQEvaluation/dist/BioASQEvaluation.jar \
evaluation.EvaluatorTask1b \
-phaseA \
/Users/greinaldpappa/Downloads/BioASQ-training13b/training13b.
   json \
/Users/greinaldpappa/Downloads/BioASQ-training13b/
   bm25_phaseA_run.json
```