TRANSLATIONAL UMLS VOCABULARY ALIGNMENT

By:
Bernal Jimenez
MOTIVATION

- UMLS is a medical knowledge base which combines over 200 medical terminologies.
  - Valuable repository for medical knowledge and useful resource for inter-operability
- UMLS contains
  - ~4 million concepts
  - ~16 million atoms (1 atom = 1 string from a specific source)
- UMLS grows larger every year
  - Approx. a million atoms are added every year
MOTIVATION

- UMLS editors integrate new atoms into UMLS
  - Every new atom needs to be associated with atoms already in UMLS.
  - Task is called UMLS Vocabulary Alignment or UVA
  - The task can range from very simple to extremely challenging and often requires in-depth domain knowledge.

- Simple Examples:
  - “ascorbic acid” => “vitamin C”
  - “lung cancer” => ”pulmonary carcinoma”

- Complex Example:
  - “SPRL1B” => “LCE2B gene”
Motivation

- Given that there are $\sim 10^{10}$ atom comparisons to be done for each new batch of terms, updating UMLS can become extremely expensive.
  - A medium sized team is unable to keep up with the task by itself.

- Currently, UMLS editors rely on rule-based tools.
  - Thus, our team has been exploring deep learning methods to alleviate the burden on UMLS editors and improve UMLS quality.
Our lab’s previous UVA work is formulated as a task where every possible pair would need to be classified.

Due to the massive size of this task ($10^{10} - 10^{14}$ pairs), current work is done on idealized (but still large-scale) subsets ($10^8$ pairs).

- 166 million negative edges when there are $10^{10}$ possible negative edges.
CURRENT TEST DATASET

Non-Synonym Edges
~166 million

Synonym Edges
~5 million
REAL WORLD DATASET

Non-Synonym Edges

$\sim 10^{14}$ edges (Not to scale)

Synonym Edges

$\sim 30$ million edges
Current datasets use a prevalence which is $10^6$ times lower than real world datasets
- Prevalence: % of positive samples in dataset

This prevalence gap means that even models which perform well in current datasets would likely yield poor results in the real-world scenario.
- Many negative edges would be incorrectly predicted as positive.
Main Goal: Build a system which can be directly deployed to support UMLS editors for UMLS construction and updating.

Research Aims:
1. Define task and datasets which faithfully represent the real-world task.
2. Design a baseline system that can address the task and dataset designed.
3. Perform quantitative and qualitative evaluation of baseline system.
TRANSLATIONAL UVA GOALS

Research Aims:

1. Define **task** and **datasets** which faithfully represent the real-world task.
   - **Task Definition:** For each new atom to be introduced to UMLS, find all synonymous atoms in the current UMLS. (analogous to real-world task)
   - **Evaluation Dataset:**
     - 430k new atoms were introduced between the first and second version of (2020AA vs 2020AB).
     - For each of these 430k new atoms in 2020AB, we are looking to determine which atoms are likely to be its synonyms in UMLS 2020AA.

2. Design a **baseline system** that can address the task and dataset designed.

3. Perform **quantitative** and **qualitative evaluation** of baseline system.
Research Aims:
1. Define task and datasets which faithfully represent the real-world task.
2. Design a baseline system that can address the task and dataset designed.
   - Two-step system:
     - High recall candidate generation (fast but misses few potential synonyms)
     - High precision synonymy classification (slower but more discriminative)
3. Perform quantitative and qualitative evaluation of baseline system.
TRANSLATIONAL UVA GOALS

Research Aims:
1. Define task and datasets which faithfully represent the real-world task.
2. Design a baseline system that can address the task and dataset designed.
3. Perform quantitative and qualitative evaluation of baseline system.
   • Quantitative Evaluation
   • High Recall Step
     • Recall at K - % of true synonyms that can be found within the first K atoms retrieved from the original UMLS.
   • High Precision Step
     • F1, Precision and Recall on true synonym pairs
TRANSLATIONAL UVA GOALS

Research Aims:

1. Define task and datasets which faithfully represent the real-world task.
2. Design a baseline system that can address the task and dataset designed.
3. Perform quantitative and qualitative evaluation of baseline system.
   - Qualitative Evaluation
     - Sample output should be carefully examined by biomedical experts
     - UMLS has some ambiguities and errors, thorough analysis is required to ascertain the quality of the predictions compared to the “gold standard”.

METHODOLOGY

- High recall candidate generation (fast but gets many false positives)
- High precision synonymy classification (slower but more discriminative)
METHODOLOGY

- High recall candidate generation (fast but gets many false positives)
- High precision synonymy classification (slower but more discriminative)
HIGH RECALL STEP: FORMULATION

- Task formulation:
  - 430k query terms
  - ~10 million term database
  - Retrieve a limited # of candidates from the database for each query which hopefully contain relevant candidates.

- Similar tasks:
  - Information retrieval (IR) (finding relevant documents with respect to a query)
  - Entity linking (finding relevant concepts with respect to a term mentioned in text)
HIGH RECALL STEP: FORMULATION

- Both IR and entity linking use modern textual encoders (often pre-trained language models) and a fast implementation of the k-nearest neighbors (k-NN) algorithm to achieve a fast and high recall candidate retrieval step.
- Spurred on by PLMs as well as k-NN speedups using GPUs (Johnson et al. 2017)
HIGH RECALL STEP: FORMULATION

- We model our high recall approach directly on biomedical entity linking:
  - In this task, a concept is mentioned within a sentence:
    - “The patient was diagnosed with pulmonary carcinoma.”
  - We then link this mention to a UMLS entity which represents “pulmonary carcinoma”. 

"The patient was diagnosed with pulmonary carcinoma"

Encoder

Mention Vector

UMLS Concept Vectors

k-NN algorithm

Ranked UMLS Concepts

UMLS Concepts
HIGH RECALL STEP: OUR APPROACH

New UMLS Term

Old UMLS Term

Encoder

New UMLS Term Vector

Old UMLS Term Vectors

k-NN algorithm

Ranked Old Terms
• Any system that produces a dense vector from text can be an encoder.
• Examples:
  • Current UVA models like LexLM, ConLM and UBERT.
  • BioWordVec embeddings
  • Pretrained language models (PLMs) like BERT, RoBERTa
  • Biomedical PLMs (PubMedBERT)
  • Biomedical PLMs with infused UMLS information (SAPBERT and KRISSBERT)
  • (Liu et al. 2020, Zhu et al. 2020, Bhowmik et al. 2021, Zhang et al. 2022, Xu et al. 2022)
HIGH RECALL STEP: GPU K-NN SPEEDUP

- Benchmarking k-NN for LexLM on GPUs vs CPU
  - Database: 8,521,220 AUIs
  - Queries: 430,135 AUIs
  - Dimension: 50
- CPU Time: 3 hours (180 minutes)
- GPU Time: 3 minutes
- GPU offers around a 60 times speedup to the k-NN
### HIGH RECALL STEP: RESULTS

<table>
<thead>
<tr>
<th>Model</th>
<th>R @ Source Synonymy</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>R@50</th>
<th>R@100</th>
<th>R@200</th>
<th>R@500</th>
<th>R@1000</th>
<th>R@2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>PubMedBERT</td>
<td>-</td>
<td>9%</td>
<td>16%</td>
<td>18%</td>
<td>23%</td>
<td>25%</td>
<td>28%</td>
<td>31%</td>
<td>34%</td>
<td>37%</td>
</tr>
<tr>
<td>LexLM</td>
<td>-</td>
<td>10%</td>
<td>22%</td>
<td>28%</td>
<td>42%</td>
<td>47%</td>
<td>51%</td>
<td>56%</td>
<td>59%</td>
<td>62%</td>
</tr>
<tr>
<td>KRISSBERT</td>
<td>-</td>
<td>13%</td>
<td>25%</td>
<td>30%</td>
<td>43%</td>
<td>48%</td>
<td>53%</td>
<td>59%</td>
<td>64%</td>
<td>68%</td>
</tr>
<tr>
<td>SAPBERT</td>
<td>-</td>
<td>20%</td>
<td>44%</td>
<td>53%</td>
<td>71%</td>
<td>76%</td>
<td>81%</td>
<td>86%</td>
<td>88%</td>
<td>89%</td>
</tr>
<tr>
<td>PubMedBERT (Source Syn)</td>
<td>35%</td>
<td>41%</td>
<td>46%</td>
<td>48%</td>
<td>52%</td>
<td>53%</td>
<td>55%</td>
<td>57%</td>
<td>59%</td>
<td>61%</td>
</tr>
<tr>
<td>LexLM (Source Syn)</td>
<td>35%</td>
<td>42%</td>
<td>51%</td>
<td>56%</td>
<td>66%</td>
<td>70%</td>
<td>73%</td>
<td>77%</td>
<td>79%</td>
<td>81%</td>
</tr>
<tr>
<td>KRISSBERT (Source Syn)</td>
<td>35%</td>
<td>43%</td>
<td>51%</td>
<td>55%</td>
<td>64%</td>
<td>68%</td>
<td>71%</td>
<td>76%</td>
<td>79%</td>
<td>82%</td>
</tr>
<tr>
<td>SAPBERT (Source Syn)</td>
<td>35%</td>
<td>46%</td>
<td>63%</td>
<td>70%</td>
<td>83%</td>
<td>86%</td>
<td>90%</td>
<td>93%</td>
<td>95%</td>
<td>95%</td>
</tr>
</tbody>
</table>
## HIGH RECALL STEP: RESULTS

<table>
<thead>
<tr>
<th>Model</th>
<th>R @ Source Synonymy</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>R@50</th>
<th>R@100</th>
<th>R@200</th>
<th>R@500</th>
<th>R@1000</th>
<th>R@2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>PubMedBERT</td>
<td>-</td>
<td>9%</td>
<td>16%</td>
<td>18%</td>
<td>23%</td>
<td>25%</td>
<td>28%</td>
<td>31%</td>
<td>34%</td>
<td>37%</td>
</tr>
<tr>
<td>LexLM</td>
<td>-</td>
<td>10%</td>
<td>22%</td>
<td>28%</td>
<td>42%</td>
<td>47%</td>
<td>51%</td>
<td>56%</td>
<td>59%</td>
<td>62%</td>
</tr>
<tr>
<td>KRISSBERT</td>
<td>-</td>
<td>13%</td>
<td>25%</td>
<td>30%</td>
<td>43%</td>
<td>48%</td>
<td>53%</td>
<td>59%</td>
<td>64%</td>
<td>68%</td>
</tr>
<tr>
<td>SAPBERT</td>
<td>-</td>
<td>20%</td>
<td>44%</td>
<td>53%</td>
<td>71%</td>
<td>76%</td>
<td>81%</td>
<td>86%</td>
<td>88%</td>
<td>89%</td>
</tr>
<tr>
<td>PubMedBERT (Source Syn)</td>
<td></td>
<td>35%</td>
<td>41%</td>
<td>46%</td>
<td>48%</td>
<td>52%</td>
<td>53%</td>
<td>55%</td>
<td>57%</td>
<td>59%</td>
</tr>
<tr>
<td>LexLM (Source Syn)</td>
<td></td>
<td>35%</td>
<td>42%</td>
<td>51%</td>
<td>56%</td>
<td>66%</td>
<td>70%</td>
<td>73%</td>
<td>77%</td>
<td>79%</td>
</tr>
<tr>
<td>KRISSBERT (Source Syn)</td>
<td></td>
<td>35%</td>
<td>43%</td>
<td>51%</td>
<td>55%</td>
<td>64%</td>
<td>68%</td>
<td>71%</td>
<td>76%</td>
<td>79%</td>
</tr>
<tr>
<td>SAPBERT (Source Syn)</td>
<td></td>
<td>35%</td>
<td>46%</td>
<td>63%</td>
<td>70%</td>
<td>83%</td>
<td>86%</td>
<td>90%</td>
<td>93%</td>
<td>95%</td>
</tr>
<tr>
<td>SAPBERT (Source Syn + LUI)</td>
<td></td>
<td>58%</td>
<td>25%</td>
<td>60%</td>
<td>71%</td>
<td>88%</td>
<td>91%</td>
<td>94%</td>
<td>96%</td>
<td>97%</td>
</tr>
</tbody>
</table>
HIGH RECALL STEP: TAKEAWAYS

- **SAPBERT** is by far the most effective encoder.
- Leveraging **basic lexical similarity** and **source synonymy** greatly improves candidates obtained from only dense representations.
  - Throwing away rule-based signal is detrimental to performance.
- We achieve above **90% recall at above 100 candidates** with the best system.
  - This is high enough for a useful real-world system to support UMLS editors (humans in the loop are still vital).
METHODOLOGY

- High recall candidate generation system (fast but gets many false positives)
- High precision synonymy classification (slower but more discriminative)
HIGH PRECISION STEP: FORMULATION

- **Output from first step:**
  - ~100-200 (query term, candidate term) pairs for each query term
  - **Imbalanced distribution:** only 5-10% of these pairs are synonymous (much higher prevalence than natural one)

- **Task formulation:**
  - Same formulation used by previous UVA methods (LexLM, UBERT))
  - Binary synonymy classification for each (query term, candidate term) pair.

- **Approach:**
  - Given the success of PLM fine-tuning in a wide range of NLP tasks, we leverage PLMs.
  - To deal with the heavy class imbalance, we sample a balanced number of positive and negative pairs for training.
**HIGH PRECISION STEP: DATASETS**

- **Dev and Test Datasets**
  - Top 100 SAPBERT candidates from the 430k new 2020AB term dataset.
  - Set aside 1000 and 2000 concepts for dev and test sets, respectively.
  - Use all 100 candidates for each concept to create dev and test sets
    - 100,000 dev set and 200,000 test set pairs
  - For this setting, we also add whatever synonyms are missing from the candidate list (not fully realistic but upper bound on performance)
HIGH PRECISION STEP: DATASETS

- **Training Datasets**
  - **Ideal Distribution**
    - Rest of 2020AB new terms (Same semantic group distribution as dev and test set)
    - By using this training set, we are inadvertently introducing information about the new terms that we would not have in the real-world setting.
  - **Realistic Distribution**
    - We separate 400k UMLS 2020AA terms as a different “new” dataset.
    - Find 100 k-NNs for each of these terms within what remains of 2020AA.
    - Create 40 million training dataset to sample balanced datasets from.
HIGH PRECISION STEP: DATASETS

- Training Dataset Types
  - Balanced
  - Stratified
  - Dev Set Equivalent: 100 candidates for each query term (not shuffled)

- Training Dataset Sizes
  - 10k
  - 100k
  - 200k
  - 500k
HIGH PRECISION STEP: MODELS

- **Fine-tuned Models**
  - PubMedBERT

- **Baselines**
  - UBERT (Original + SAPBERT)
  - LexLM
  - ConLM
## HIGH PRECISION STEP: RESULTS

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Dataset</th>
<th>Training Data Type</th>
<th>Training Data Size</th>
<th>F1</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAPBERT + UBERT Synonymy Prediction</strong></td>
<td>UVA Train</td>
<td>Stratified</td>
<td>166 M</td>
<td>27.4%</td>
<td>16.3%</td>
<td>87.4%</td>
</tr>
<tr>
<td><strong>UBERT MLM + Synonymy Prediction</strong></td>
<td>UVA Train</td>
<td>Stratified</td>
<td>166 M</td>
<td>35.0%</td>
<td>22.0%</td>
<td>85.0%</td>
</tr>
<tr>
<td><strong>PubMedBERT Fine Tuning</strong></td>
<td>Ideal</td>
<td>Balanced</td>
<td>10k</td>
<td>38.2%</td>
<td>24.6%</td>
<td>85.4%</td>
</tr>
<tr>
<td><strong>PubMedBERT Fine Tuning</strong></td>
<td>Ideal</td>
<td>Stratified</td>
<td>10k</td>
<td>40.6%</td>
<td>38.1%</td>
<td>43.3%</td>
</tr>
<tr>
<td><strong>PubMedBERT Fine Tuning</strong></td>
<td>Ideal</td>
<td>Balanced</td>
<td>100k</td>
<td>46.7%</td>
<td>31.7%</td>
<td>88.8%</td>
</tr>
<tr>
<td><strong>PubMedBERT Fine Tuning</strong></td>
<td>Ideal</td>
<td>Stratified</td>
<td>100k</td>
<td>33.4%</td>
<td>49.1%</td>
<td>25.3%</td>
</tr>
<tr>
<td><strong>PubMedBERT Fine Tuning</strong></td>
<td>Ideal</td>
<td>Dev Set Equivalent</td>
<td>100k</td>
<td>25.9%</td>
<td>54.7%</td>
<td>17.0%</td>
</tr>
</tbody>
</table>
HIGH PRECISION STEP: TAKEAWAYS

- Both UBERT versions (which outperform other models in UVA work) underperform small-scale fine-tuning
  - Small (100k samples) but more realistic datasets yield better real-world performance than training on millions of synonym pairs.

- Training set distribution drastically affects performance
  - Balanced datasets (1:1) yield high recall but low precision
  - Stratified datasets (1:~10) yield higher precision but very low recall
## HIGH PRECISION STEP: RESULTS

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Dataset</th>
<th>Training Data Type</th>
<th>Training Data Size</th>
<th>F1</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>PubMedBERT Fine Tuning</td>
<td>Ideal</td>
<td>Balanced</td>
<td>100k</td>
<td>46.7%</td>
<td>31.7%</td>
<td>88.8%</td>
</tr>
<tr>
<td>PubMedBERT Fine Tuning</td>
<td>Realistic</td>
<td>Balanced</td>
<td>100k</td>
<td>41.5%</td>
<td>27.3%</td>
<td>86.6%</td>
</tr>
<tr>
<td>PubMedBERT Fine Tuning</td>
<td>Ideal</td>
<td>Balanced</td>
<td>200k</td>
<td>42.8%</td>
<td>32.0%</td>
<td>64.7%</td>
</tr>
<tr>
<td>PubMedBERT Fine Tuning</td>
<td>Realistic</td>
<td>Balanced</td>
<td>200k</td>
<td>43.8%</td>
<td>29.0%</td>
<td>90.2%</td>
</tr>
<tr>
<td>PubMedBERT Fine Tuning</td>
<td>Ideal</td>
<td>Balanced</td>
<td>500k</td>
<td>52.5%</td>
<td>37.1%</td>
<td>90.0%</td>
</tr>
<tr>
<td>PubMedBERT Fine Tuning</td>
<td>Realistic</td>
<td>Balanced</td>
<td>500k</td>
<td>37.1%</td>
<td>23.1%</td>
<td>94.6%</td>
</tr>
</tbody>
</table>
HIGH PRECISION STEP: TAKEAWAYS

- Original distribution training datasets underperform ideal distribution training.
- The correlation between dataset size and performance is not as strong as expected.
- Training is quite noisy
  - Training metrics keeps increasing but dev set performance drops after epoch 1 in most cases

- Precision is only at ~30-40%, not high enough for deployable system
  - 2/3 of all predicted synonym pairs are not synonymous according to gold labels
  - Qualitative evaluation is necessary to determine how this model performs in practice
QUALITATIVE EVALUATION: FALSE POSITIVES

- Low precision problem is due to the high number of false positives
- For every 1 synonym pair predicted correctly, 2 are incorrect according to UMLS
- Unfortunately, or fortunately, it is very difficult to determine whether each of these false positives is a true error or a UMLS error
- The amount of time spent on each term would be very large (even for a person with some biomedical training)
## Qualitative Evaluation: False Positives

<table>
<thead>
<tr>
<th>Query</th>
<th>Candidate</th>
<th>Label</th>
<th>Pred</th>
</tr>
</thead>
<tbody>
<tr>
<td>arginine/serine-rich protein 1</td>
<td>SRA1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ATP synthase, H+ transporting, mitochondrial Fo complex, subunit F2</td>
<td>ATP5MC3 gene</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>pseudogene 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDYS</td>
<td>SDYS</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>BENZALKONIUM CHLORIDE 1 mg in 1 g TOPICAL CLOTH [Antiseptic Towel</td>
<td>Pro Pet Dental Wipes 0.1 % Medicated Pad</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Benzalkonium Chloride]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>transfer RNA tyrosine 1 (anticodon GUA)</td>
<td>TRT-AGT2-1 gene</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>WW domain binding protein 1-like pseudogene 4</td>
<td>WBP4 gene</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>protirelin</td>
<td>Thyrotropin-releasing factor, prepro-</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sodium hyaluronate 23 MG/ML Injectable Solution</td>
<td>HYALURONATE NA (GEL-ONE) 10MG/ML SYR 3ML</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sf9 Cell</td>
<td>SR cell line</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>fragile histidine triad diadenosine triphosphatase</td>
<td>Hemin-Controlled Translational Repressor</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Meningismus</td>
<td>hemiballisms</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Endocervix</td>
<td>Endocervical epithelium structure (body</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>structure)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CONCLUSION

Complete:
1. Define task and datasets which faithfully represent the real-world task.
2. Design a baseline system that can address the task and dataset designed.

In Progress:
1. Perform quantitative and qualitative evaluation of baseline system.
   1. More baselines are necessary for classification system
   2. Thorough qualitative evaluation is crucial
FUTURE CHALLENGES

- Addressing the more moderate but still important class imbalance problem
  - Distributionally robust optimization (Levy et al. 2020) or other similar techniques
- UMLS is noisy and synonymy task is often ambiguous
- Data scarcity
  - Only data point for determining synonymy is a short phrase and the source it comes from.
  - This is unrealistic since other data points are used by humans to make determination (other source synonyms, source semantic categories, descriptions, hierarchical structure, etc.).
  - Adding this information is crucial for models to perform better.
REFERENCES

- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, Arnold Overwijk: “Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval”, 2020;
- Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, HaiFeng Wang: “RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering”, 2020;