QuickUMLS: a Fast, Unsupervised Approach for Medical Concept Extraction

Luca Soldaini and Nazli Goharian

Information Retrieval Lab
Georgetown University

2016-07-21 MedIR ‘16
A 47 year old male who fell on his left arm presents with pain and bruising on the elbow, swelling, and inability to bend the arm.
Task

Medical Information Extraction (MIE):
Extract concepts and their location from medical document

A 47 year old male who fell on his left arm presents with pain and bruising on the elbow, swelling, and inability to bend the arm.
State of the Art

• **MetaMap** (Aronson 2001, Aronson & Lang 2010)
  - Designed for biomedical text, handles negation, word sense disambiguation

• **cTAKES** (Savova et al 2010)
  - Created for clinical notes

• Focus is on accuracy, not performance
  - OK if 1000s of documents, challenging if more
  - Is real-time analysis a goal?
This Work

• Introduce *QuickUMLS*: unsupervised IE algorithm

• Compared to state-of-the-art:
  • Similar or better performance (Prec, Rec, F1)
  • Significantly faster (2 to 135 times)
  • 500 - 1000 tokens processed per second

• Tests on three datasets: *i2b2*, *THYME*, drug reviews

• Python 2/3 implementation available at: https://github.com/Georgetown-IR-Lab/QuickUMLS
A 47 year old male who fell on his left arm presents with pain and bruising on the elbow, swelling, and inability to bend the arm.

- **left arm** → Left arm structure, C0230347
- **pain** → Pain, C0030193
- **bruising** → Contrusions, C0009938
- **inability to bend** → Ability to bend, C0560887

### UMLS concept matching

- **candidates generation**

### System Overview

Input text
Candidates generation

1. Document tokenization and PoS extraction

2. Generate all seq. of tokens with length up to \( w \) such that:

1. contains at least one word & it is not a stopword

2. not span across sentences

3. does not start or end with conjunction, adposition, determiner, or punctuation

\[
\begin{align*}
\text{Woke up} & \quad \text{nine} \\
\text{in} & \quad \text{pain} \\
\text{bruise} & \quad \text{mark on the} \\
\text{on the} & \quad \text{left. Arm} \\
\text{left arm.} & \quad \text{was bruised} \\
\text{pain in the} & \quad \text{the patient} \\
\text{right arm.} & \quad \text{was in} \\
\text{the patient} & \quad \text{pain.}
\end{align*}
\]
UMLS concept matching

- **CPMerge** (Okazaki and Tsujii, 2010) used for matching sequences to *UMLS* concepts

- For each sequence $d$, determine the set of concepts $C_K$ such that:
  
  $$StringSimilarity(d, c_{iK}) \geq \alpha \quad \forall c_{iK} \in C_K$$

- For efficiency:
  - strings are tokenized in trigrams and indexed using an inverted index
  - each trigram posting list is partitioned by length of the strings containing the trigram

- In our experiments: *Jaccard* similarity, $0.6 \leq \alpha \leq 1.0$
Experimental Setup

- **2010 i2b2/VA Challenge Dataset** (Uzuner et al., 2011)
  - 169 annotated with medical concepts for US VA dept.

- **THYME Corpus** (Styler et al., 2014)
  - 1,254 de-identified clinical reports from Mayo Clinic

- **Drug Reviews** (Yates and Goharian, 2013)
  - 2,500 reviews for **Anastrozole, Exemestane, Letrozole, Raloxifene, and Tamoxifen**
  - generated by laypeople, annotated for drugs side effects

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Tokens per doc</th>
<th>Concepts per doc</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>i2b2</em> dataset</td>
<td>1,040</td>
<td>99</td>
</tr>
<tr>
<td><em>THYME</em> corpus</td>
<td>1,035</td>
<td>172</td>
</tr>
<tr>
<td><em>Drug Reviews</em></td>
<td>131</td>
<td>2</td>
</tr>
</tbody>
</table>
Experimental Setup

- *SpaCy* for tokenization, parsing, and chunking
  - v.0.100.7, https://spacy.io/

- *MetaMap*
  - v.2016, UMLS 2015AB release, NegEx processing
  - Phrase chunking done with SpaCy (much faster)

- *cTAKES*
  - v.3.2.2, FastUMLSPrecessor pipeline.
Results – i2b2

- **cTAKES** has the best precision
- **QuickUMLS** has best recall, close to cTAKES when $\alpha = 1.0$
- **F1**: QuickUMLS = 0.63, cTAKES = 0.61, MetaMap = 0.48
- Small $\alpha$: more matches, better recall, lower precision, slower
• Results are similar to i2b2
• \texttt{cTAKES} has still the best precision, \texttt{QuickUMLs} best recall
• F1: \texttt{QuickUMLs} = 0.72, \texttt{cTAKES} = 0.68*, \texttt{MetaMap} = 0.61*
• \texttt{QuickUMLs} is 2-26 times faster than \texttt{cTAKES}
• Results are worse
  • laypeople content is harder to parse
  • only adverse symptoms are annotated

• F1: QuickUMLS = 0.48, cTAKES = 0.22*, MetaMap = 0.14*
• QuickUMLS has the best precision and recall
Conclusions

• *QuickUMLS*: unsupervised concept extraction
  • Uses approximate dictionary mapping to match sequences of tokens to UMLS concepts

• Proposed method performs similarly or better than the state of the art

• 2 to 135 times faster than cTAKES or MetaMap

• Available at: [https://github.com/Georgetown-IR-Lab/QuickUMLS](https://github.com/Georgetown-IR-Lab/QuickUMLS)

@soldni  luca@ir.cs.georgetown.edu
<table>
<thead>
<tr>
<th>Method</th>
<th>Prec</th>
<th>Rec</th>
<th>F-1</th>
<th>ms/doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaMap</td>
<td>0.49*</td>
<td>0.48*</td>
<td>0.48*</td>
<td>19,295*</td>
</tr>
<tr>
<td>cTAKES</td>
<td>0.71*</td>
<td>0.53*</td>
<td>0.61</td>
<td>3,852*</td>
</tr>
<tr>
<td>QuickUMLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha = 0.6$</td>
<td>0.50*</td>
<td>0.75</td>
<td>0.60</td>
<td>1,594*</td>
</tr>
<tr>
<td>$\alpha = 0.7$</td>
<td>0.60*</td>
<td>0.66*</td>
<td>0.63</td>
<td>680*</td>
</tr>
<tr>
<td>$\alpha = 0.8$</td>
<td>0.63*</td>
<td>0.60*</td>
<td>0.61</td>
<td>332*</td>
</tr>
<tr>
<td>$\alpha = 0.9$</td>
<td>0.64*</td>
<td>0.56*</td>
<td>0.60</td>
<td>193*</td>
</tr>
<tr>
<td>$\alpha = 1.0$</td>
<td>0.67*</td>
<td>0.54*</td>
<td>0.60</td>
<td>143</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Prec</th>
<th>Rec</th>
<th>F-1</th>
<th>ms/doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaMap</td>
<td>0.71*</td>
<td>0.53*</td>
<td>0.61*</td>
<td>15,935*</td>
</tr>
<tr>
<td>cTAKES</td>
<td>0.89*</td>
<td>0.55*</td>
<td>0.68*</td>
<td>3,765*</td>
</tr>
<tr>
<td>QuickUMLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha = 0.6$</td>
<td>0.68*</td>
<td>0.77</td>
<td>0.72</td>
<td>1,536*</td>
</tr>
<tr>
<td>$\alpha = 0.7$</td>
<td>0.78*</td>
<td>0.67</td>
<td>0.72</td>
<td>646*</td>
</tr>
<tr>
<td>$\alpha = 0.8$</td>
<td>0.83*</td>
<td>0.61</td>
<td>0.70</td>
<td>340*</td>
</tr>
<tr>
<td>$\alpha = 0.9$</td>
<td>0.85*</td>
<td>0.57</td>
<td>0.68</td>
<td>174*</td>
</tr>
<tr>
<td>$\alpha = 1.0$</td>
<td>0.87*</td>
<td>0.55</td>
<td>0.67</td>
<td>141</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Prec</th>
<th>Rec</th>
<th>F-1</th>
<th>ms/doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaMap</td>
<td>0.12*</td>
<td>0.16*</td>
<td>0.14*</td>
<td>1,774*</td>
</tr>
<tr>
<td>cTAKES</td>
<td>0.16*</td>
<td>0.37*</td>
<td>0.22*</td>
<td>301*</td>
</tr>
<tr>
<td>QuickUMLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha = 0.6$</td>
<td>0.16*</td>
<td>0.6</td>
<td>0.25*</td>
<td>116*</td>
</tr>
<tr>
<td>$\alpha = 0.7$</td>
<td>0.38*</td>
<td>0.51</td>
<td>0.44*</td>
<td>57*</td>
</tr>
<tr>
<td>$\alpha = 0.8$</td>
<td>0.43</td>
<td>0.51</td>
<td>0.47</td>
<td>32*</td>
</tr>
<tr>
<td>$\alpha = 0.9$</td>
<td>0.47</td>
<td>0.50</td>
<td>0.48</td>
<td>22</td>
</tr>
<tr>
<td>$\alpha = 1.0$</td>
<td>0.47</td>
<td>0.45*</td>
<td>0.46</td>
<td>18</td>
</tr>
</tbody>
</table>