Porting a Natural Language Processing Algorithm to Extract Findings from Colonoscopy Pathology Reports

Jennifer A. Pacheco\textsuperscript{1}; Andrew J. Gawron, MD, PhD\textsuperscript{2}; Kenneth Borthwick\textsuperscript{3}; Peggy L. Peissig, PhD\textsuperscript{4}; David S Carrell, PhD\textsuperscript{5}; Luke V. Rasmussen\textsuperscript{1}; Huan Mo, MD\textsuperscript{6}; William K. Thompson, PhD\textsuperscript{1}

\textsuperscript{1}Northwestern University, Chicago, IL; \textsuperscript{2}University of Utah, Salt Lake City, UT; \textsuperscript{3}Geisinger Health System, Danville, PA; \textsuperscript{4}Marshfield Clinic, Marshfield, WI; \textsuperscript{5}Group Health Research Institute, Seattle, WA; \textsuperscript{6}Vanderbilt University, Nashville, TN

March 22, 2016
Disclosure

- Will Thompson is co-founder of Textractor Technologies LLC
This work has been supported in part by funding from PhEMA (R01 GM105688) and eMERGE (U01 HG006379, U01 HG006378 and U01 HG006388).
Electronic Medical Records and Genomics Network

- NHGRI funded project (currently in 3rd funding cycle)
- Development of methods and best practices for using the EHR as a tool for genomic research
- Network members link EHR data to DNA biobanks, pool results across sites
- Several dozen electronic phenotypes completed and in progress (www.PheKB.org)

Example Phenotypes: Type 2 Diabetes, Peripheral Arterial Disease, QRS Duration, Colon Polyps, Resistant Hypertension
Phenotype: Colon Polyps

- Colorectal cancer (CRC) is the 2nd leading cause of cancer-related mortality in the United States
- Colonoscopy is a widely used CRC screening modality, results in polyp biopsies with histologies described in pathology notes
- Serrated vs. adenoma cancer pathway
- Goal: EHR-based phenotype and GWAS across eMERGE sites
Developing and Porting the NLP-based Algorithm

NLP Modules

Enterprise Data Warehouse

Pathology Notes

Histology, Location

UIMA Pipeline

Groovy Script Annotator
Sentence Splitter
Groovy Script Annotator
Groovy Script Annotator
Groovy Script Annotator

Sectioning Script

cTAKES sentence model

Polyp Histology and Location Concept Script

NegEx Script

Relation Extractor Script
Groovy Script UIMA Annotator

Groovy script UIMA annotators are very easily configured, and scripts can be updated and re-loaded without recompiling the entire NLP library. Built-in functionality for Groovy scripts includes:

- Support for cTAKES common type system
- Selecting and filtering annotations
- Creating annotations
- Matching patterns of annotations and text
@Override
void initialize(UimaContext context) {
    super.initialize(context)
    config = new CompilerConfiguration()
    config.setScriptBaseClass("org.northshore.cbri.UIMAUril")
    shell = new GroovyShell(config)
    // load in script file contents
    this.script = shell.parse(scriptContents)
}

@Override
void process(JCas jcas) {
    UIMAUril.setJCas(jcas)
    this.script.run()
}
Annotation Selection

```plaintext
// select all Sentences containing an EntityMention
sents = select
  type: Sentence,
  filter: contains(EntityMention)
```

- Built-in support for cTAKES common type system
- Pre-defined and on-the-fly filters
- Compositional filters w. boolean functions
- First class support for regex strings & operators
// select all Sentence annotations contained in
// "Findings" Segment that also contains an
// EntityMention annotation and ends with
// text "tubular adenoma"

select(type: Segment).grep { seg ->
    seg.id == "FINDINGS"}.each {
    select(type: Sentence, filter: (
        and (coveredBy(seg),
            {it.coveredText==~/.* tubular \s+ adenoma},
            contains(EntityMention))
    )}
Annotation Creation

```python
create(type=EntityMention,
    begin:0, end:10,
    polarity:1, uncertainty:0,
    ontologyConcepts:[
        create(type:UmlsConcept, cui:"C01"),
        create(type:UmlsConcept, cui:"C02")
    ]
)
```

- Built-in support for cTAKES common types
- Can nest calls to `create` on the fly
Annotation Matching

```python
sents = select(type: Sentence)
patterns = [~/(?i)(tubular|villous)\s+adenoma/]  

match(sents, patterns,  
{  
create(type: EntityMention,  
  begin: it.start(1), end: it.end(1),  
  polarity: 1, uncertainty: 0,  
  ontologyConcepts: [  
    create(type: UmlsConcept, cui: "C01")  
  ]  
})
```

- Patterns can be specified over text and/or annotations
- Specified functions are applied to every match
- Action taken can be anything (create annotation one possibility)
Annotations Matching (More Complex Example)

```java
pat = (∼/ (?s)(?<h1>@Head)(?= (?<h2>@Head)| \Z))/
AnnotationMatcher matcher =
    pat.matcher(includeText: false)

matcher.each{ Map binding ->
    create(type: Segment,
        begin: binding.get("h1").begin,
        end: (binding.get("h2") ?
            binding.get("h2").begin :
            jcas.documentText.length()))
}
```
Abstractor Tool

https://github.com/lrasmus/DocumentAbstraction
KNIME Execution Workflow

https://www.knime.org
KNIME Integration with NLP Pipeline

Table Creator

SNOMED findings

Joiner

Column Filter

Java Snippet

Column Filter

Ungroup

Java Snippet

parse JSON

Row Splitter

GroupBy

de-dup

Table Creator

SNOMED locations

Dialog - 0:276:160 - Java Snippet (NLP)

Column List

ROWID

ROWINDEX

ROWCOUNT

S pat_id

i note_csn_id

S path_text

10 // system imports
11 // Your custom imports:
12 import edu.northwestern.fsm.ColonPathReportProcessor;
13
14 // system variables
15 // Your custom variables:
16
17 // expression start
18 // Enter your code here:
19 out_nlp_re = ColonPathReportProcessor.processText(c_path_text);
KNIME Validation Workflow
Sharing KNIME Workflows

Exporting a KNIME workflow

Importing a KNIME workflow
Results

- Using the annotator tool at a single site, a corpus of 200 randomly selected pathology notes was created. At the polyp finding level (histology + location match), the algorithm achieved sensitivity of 0.95 and PPV 0.95.
- The algorithm and tools were subsequently shared with three additional eMERGE sites
- Porting to new sites required modifications to script files regulating sectionizing and relation extraction.
- After running the algorithm across all four sites, we were able to extract a genotyped cohort of 5839 patients.
- Our qualitative evaluation indicated that using the tools significantly sped up the porting process, and is a promising approach to similar tasks involving NLP-based phenotype algorithms.
Conclusion

- Porting NLP-based phenotype algorithms across sites presents time-consuming challenges, limiting cross-site collaboration opportunities.
- We developed a set of tools to help make it easier to share, execute, modify, and validate NLP algorithms.
- In future work we will perform a quantitative evaluation of the benefits of using these tools for porting NLP-based phenotype algorithms.
- Pathology notes are relatively uniform and lend themselves to pattern-based rules; more complex notes or tasks may not lend themselves as well to this approach.
- The integration of NLP with KNIME can be improved by developing extension nodes that allow for direct editing of script files without requiring re-generation of NLP libraries.
Thanks!