Natural Language Processing For Requirements Engineering

Presenter: Ashutosh Adhikari
Mentor: Daniel Berry
Outline

- Research in NLP 4 Requirements Engineering (Part I)
  - 4 dimensions for NLP in RE
  - Reviewing and analysing the NLP4RE’19 workshop
- Identifying Requirements in NL research (Part II)
  - Trends in NLP-research
  - Requirements for betterment of research in NLP
- Conclusion
Requirements in Natural Language

- Requirements have been traditionally documented in Natural Language...
- However, NL has its own caveats
  - Ambiguous
  - Cumbersome to examine manually
  - Rich in variety
- RE can reap benefits from the NLP algorithms
Natural Language Requirements Processing

4 dimensions (Ferrari et al. 2017):

- Discipline
- Dynamism
- Domain Knowledge
- Datasets

"Natural Language Requirements Processing: A 4D Vision", Ferrari et al. 2017
Dynamism

- Requirements change/modify during the development phase
- Requirements traceability
  - Cross-linking requirements with other requirements
- Requirements categorization
  - Aids in managing large number of requirements
- Apportionment of requirements to specific software components
- Partition requirements into security, availability, usability ..... 
- Useful during transition from requirements to architectural design
Discipline

- Requirements are abstract conceptualization of system needs
  - and are open to interpretation
- Software developments standards like CENELEC-50128 (railway software), DO-178C (avionics), 830TM-1998(IEEE standard), etc ask requirements to be unequivocal
  - None provide language guidelines
- Enter ambiguity (remember Dan’s lectures?)
  - Research on ambiguity
  - Pragmatic analysis and disambiguation is being taken up by NLPeople
- Solution: *Templates and common requirement languages*
Domain Knowledge

- Requirements are mostly loaded with domain-specific or technical jargons
- Domain-knowledge is needed in requirements elicitation
- NL techniques can be used to find topic clusters
  - Discover fine-grained relationships among relevant terms
  - "Text-to-knowledge"
- Solution:
  - Mine Slack, Trello or Workplace
  - Domain-specific ontologies can be developed
    - Can further help with traceability and categorization (dynamism)
Datasets

- "Modern NLP techniques are data hungry, and datasets are still scarce in RE"
- Sharing is caring
  - Take-away from the NLP-community
- Standardized datasets
  - Leaderboards
  - Competitive and Collaborative Research
- Active Learning to the rescue
Reviewing NLP4RE19 Workshop (Major Projects)

- A workshop initiated to record and incentivize research in NLP4RE
- Coming up: Possible collaborations with the Association of Computational Linguistics (ACL)
  - “The Best is Yet to Come” (Dalpiaz et al. 2018)-NLP4RE workshops with *ACL
- Good starting point for us!

- Let’s look at some papers (from all the 4 dimensions)
NLP4RE Workshop (What are they looking at?)

- Resource Availability:
  - Techniques in NLP depend on data quality and quantity
- Context Adaptation
  - NLP techniques need to be tuned for the downstream tasks in RE
- Player Cooperation
  - Mutual cooperation between the players is essential
Resource Availability

- Creation of reliable data corpora
  - The data is usually companies’ requirements
  - Annotations from experts needed for training ML algorithms

- Data quality and heterogeneity
  - The sources of NL (e.g., app reviews) may exhibit poor quality
  - Variety of formats (rigorous NL specifications, diagrammatic models to bug reports)

- Validation metrics and workflows
  - RE has traditionally borrowed validation approaches from IR
  - Need to device metrics for RE specifically (Dan’s concerns)
Context Adaptation

- **Domain Specificity**
  - Each domain has its own jargon
  - NLP tools need to handle specificity
- **Big NLP4RE**
  - NLP4RE tools need to take into account artifacts like architecture, design diagram, evolution of software, etc
  - Companies may have large number of artifacts
- **Human-in-the-loop**
  - AI not at a cost of but for aiding humans
  - Active Learning
- **Language Issues**
  - non-english data
  - Low resources tools
Player Cooperation

- RE researchers
  - RE researchers need to be well versed with NLP algorithms and their usage
- NLP experts
  - NLP experts need to be introduced to problems in RE
- Tool vendors
- Industries
  - Strong interaction with industries is needed
Domain Specific Polysemous Words (Domain Knowledge and Discipline)

- Motivation:
  - Managing multiple related projects may lead to ambiguity
  - Goal is to determine if a word is used differently in different corpora

- Approach:
  - Given 2 corpora $D_1$, $D_2$ and a word $t$
  - Calculate context centers and similarity between them based on word vectors $v$. *(skipping the technicalities)*

- Strengths:
  - Need not train domain-specific word-vectors

- Weaknesses:
  - Old techniques (is it 2014?)

"Determining Domain-specific Differences of Polysemous Words Using Context Information", Toews and Holland, 2019
Table 5: Highest and lowest context similarity scores of a pairwise comparison of four requirement datasets $P_1, P_2, P_3$ and $P'_3$, where the latter two originate from a single project with requirements split in two parts.
Detection of Defective Requirements (Discipline)

- Carelessly written requirements are an issue
  - Can be misleading, redundant or lack information
- An automatic way of identifying defects is desirable
- Solution Proposed: Rule-based scripts
  - Advantages: Rules are easy to maintain
    - Enforce narrow linguistic variations in requirements
  - Disadvantages: Lacks generalization
    - Can you really enforce rules on non-technical clients (unreasonable)?

“Detection of Defective Requirements using Rule-based scripts”, Hasso et al., 2019
# Kinds of defects

<table>
<thead>
<tr>
<th>Defect</th>
<th>Example</th>
<th>Concern</th>
<th>Occurrence per 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty Valuephrase</td>
<td>&quot;The system should <strong>perform a data transfer</strong> regularly.&quot;</td>
<td>The action should be expressed through the main verb.</td>
<td>35</td>
</tr>
<tr>
<td>Incomplete Condition</td>
<td>&quot;<strong>In a state of emergency</strong>, the system needs to transfer data via radio.&quot;</td>
<td>How should data be transferred normally?</td>
<td>4</td>
</tr>
<tr>
<td>No Atomicity</td>
<td>&quot;The application should transmit data via radio <strong>and</strong> run on every operating system.&quot;</td>
<td>This should be two requirements.</td>
<td>78</td>
</tr>
<tr>
<td>Passive</td>
<td>&quot;The system should <strong>be updated</strong>.&quot;</td>
<td>Doesn’t specify who’s responsible.</td>
<td>17</td>
</tr>
<tr>
<td>Quantor</td>
<td>&quot;<strong>All users should have access to the database.</strong>&quot;</td>
<td>Should really all the users have access?</td>
<td>4</td>
</tr>
<tr>
<td>Vague Adjective</td>
<td>&quot;The system should transmit data <strong>quickly</strong>.&quot;</td>
<td>How quick is considered quickly?</td>
<td>8</td>
</tr>
<tr>
<td>Indefinite Article</td>
<td>&quot;Ein Soldat muss das System <strong>besitzen</strong> können.&quot;</td>
<td>In German, the indefinite article and the numeral one are homonymous.</td>
<td>0</td>
</tr>
<tr>
<td>Temporal Clause</td>
<td>&quot;<strong>While</strong> the system is booting up, data mustn’t be sent.&quot;</td>
<td>What is actually meant is a condition.</td>
<td>0</td>
</tr>
<tr>
<td>Redundant Clause</td>
<td>&quot;The administrator needs to change data <strong>at any time</strong> <strong>in order to help the user with his problems.</strong>&quot;</td>
<td>No need to justify a requirement at this place.</td>
<td>0</td>
</tr>
<tr>
<td>Incomplete Comparison</td>
<td>&quot;The system needs to be <strong>faster</strong>.&quot;</td>
<td>Faster than what?</td>
<td>0</td>
</tr>
</tbody>
</table>
Solution Proposed
Examples of rules

- Rules for identifying passive voice: based on strict word-order which has to be followed.
- Rules for empty verb phrase: presence of verb with broad meaning and a noun which expresses the process
# Results

<table>
<thead>
<tr>
<th></th>
<th>True Positive</th>
<th>False Positive</th>
<th>False Negative</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong></td>
<td>108</td>
<td>40</td>
<td>38</td>
<td>0.73</td>
<td>0.74</td>
<td>0.753</td>
</tr>
<tr>
<td><strong>Empty Verbphrase</strong></td>
<td>23</td>
<td>13</td>
<td>12</td>
<td>0.639</td>
<td>0.657</td>
<td>0.648</td>
</tr>
<tr>
<td><strong>Incomplete Condition</strong></td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>No Atomicity</strong></td>
<td>66</td>
<td>22</td>
<td>12</td>
<td>0.75</td>
<td>0.846</td>
<td>0.795</td>
</tr>
<tr>
<td><strong>Passive</strong></td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td><strong>Quantor</strong></td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1.0</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td><strong>Vague Adjective</strong></td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>1.0</td>
<td>0.125</td>
<td>0.222</td>
</tr>
</tbody>
</table>
Analysis of the work

- The rule-based scripts did pretty well
- However, can't generalize
- Such rules can't be developed for all languages
**NLP4RE at FBK-Software (Dynamism)**

<table>
<thead>
<tr>
<th>NL Artefact</th>
<th>RE Task</th>
<th>Technique</th>
<th>Application Domain</th>
<th>Use Case</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements documents, free textual NL in English</td>
<td>Semi-structured specification of Requirements</td>
<td>Rule-based and Controlled Natural Language</td>
<td>European Railways Signaling System</td>
<td>Validation and verification of requirements specifications</td>
<td>[CRST12, CRST11]</td>
</tr>
<tr>
<td>Online discussion, as in user forum. Thread of textual messages in English</td>
<td>Elicitation of Requirements’ relevant information</td>
<td>Speech-Act based analysis techniques, ML classification algo.</td>
<td>OSS Software development</td>
<td>Stakeholder feedback analysis for software maintenance and evolution in OSS Requirements management</td>
<td>[MPC14, MRKP18]</td>
</tr>
<tr>
<td>User-feedback, short textual messages in English</td>
<td>Elicitation of Requirements relevant information</td>
<td>Sentiment analysis and Speech-Act based analysis techniques, ML classification algo.</td>
<td>Home energy management apps</td>
<td>Requirements management for software evolution</td>
<td>[MRKP18]</td>
</tr>
<tr>
<td>User-feedback, short textual messages in German</td>
<td>Elicitation of Requirements relevant information</td>
<td>Sentiment analysis, ML classification algo.</td>
<td>Home energy management apps</td>
<td>Requirements management for software evolution</td>
<td>[KPS18]</td>
</tr>
</tbody>
</table>

Analysis of online comments (Dynamism)
Future work

- Issue prioritization
  - Associating feedbacks to issues
  - Extract properties of feedback
  - Infer issue rankings based on associated feedback’s properties
What about datasets?

- No paper found at NLP4RE covering this aspect
- The community needs retrospection for the datasets which must be created
Note:
In the light of ML being rampantly applied for NLP tasks, I shall try to have different content than the previous presenters in the course (Bikramjeet, Priyansh, Shuchita, Varshanth and ChangSheng)
Previously in Natural Language Processing...

- Earlier (Pre mid-2018), solutions proposed were specific to a downstream task
  - State-of-the-art for a dataset or at max a set of datasets
- The models were usually trained from scratch over pre-trained word vectors
- RNNs and CNNs were widely used
- 2018 onwards Pre-trained models:
  - ULMFiT, BERT, GPT, XL-NET
- Basic Idea: learn embeddings such that the model understands the language
  - Fine-tune for any downstream tasks
- “Beginning of an era?”
The rise of the Transformer

- Transformers (2017) (Vaswani et al.)
- Open AI GPT (2018) (Radford et al.)
- BERT (2018) (Devlin et al.)
- Open AI GPT-2 (2018-19)
- XL-NET (2019)

Basic Idea : A one-for-all model!

TL;DR : Develop huge parallelizable models!

[1] "Attention is all you need", Vaswani et al. 2017
[3] "Improving Language Understanding with Unsupervised Learning", Radford et al., 2018
Requirements in the Transformer Era

- Go Small!!
  - The models are getting larger and larger (> billions of parameters)
  - Most of the labs in universities can't afford to even finetune the pre-trained models
  - Current transformers are fit for industrial use only
  - Very little attempt for compressing these models (LeCun 1998)

- Verifiable claims:
  - “We crawled the net, used x billion parameters, we beat everything!!”

- Leaderboard chasing:
  - MSMARCO (Passage ranking, RC, QA)
  - HOTPOT-QA (RC and QA)
  - GLUE (Natural Language Understanding), etc

[1] "MS MARCO : A MAchine Reading COmprehension dataset", Bajaj et al., 2016
Wait, aren’t Leaderboards good?

- Only reward SOTA
  - Need more metrics like: size of the model used, data samples used, hours for training, etc.
- Leaderboards hamper interpretability
- Participants aren’t forced to release models
- Huge models trained on thousands on GPUs overshadow contributions

TL;DR: Leaderboards aren’t a good way of doing Science (Anna Rogers, UMASS)
Where is the empirical gain coming from?

- Varshanth’s, Priyansh’s and Bikramjeet’s presentation
  - Basically, we need to get out act right while applying ML
- Lipton et al., Sculley et al. argue that many of the gains are just noise!
  - Induced from excessive hyperparameter tuning
- We (our research group) found that LR, SVM and BiLSTM were beating many other complex models for Document Classification
- With increasing hyperparameters, come increasing noise
  - Difficult to credit the component which is giving performance gains
- TL; DR: Requirement to do more analysis than just reporting “good” results for interpretability

[1] “Troubling trends in Machine Learning Scholarship”, Lipton and Steinhardt, 2018
Learnt models need to be Fair!

- Shuchita’s presentation
- Pretrained models like BERT have been shown to have learnt biased embeddings
- Requirement to either:
  - Debias the learnt models
  - Use unbiased data
- TL;DR: Requirements for models to be unbiased
RE for [NLP for RE] (Dan's concerns)

- Already covered in ChangShen's presentation
- TL;DR: We have to come up with RE-specific metrics
  - Not blindly borrow metrics from IR/NLP domain
Conclusion (NLP4RE)

- Need better models (rule-based techniques aren’t good enough)
- Need ways to share data, models, and code for rapid development
- Good days are coming
**Conclusion (RE4NLP)**

- Requirements for:
  - Fair, robust and interpretable models
  - Feasible models
  - Reliable evaluation criteria (leaderboards aren't going to cut it)
  - Models need to be evaluated rigorously (empirical rigor)
    - Proper ablation studies
Thank you