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

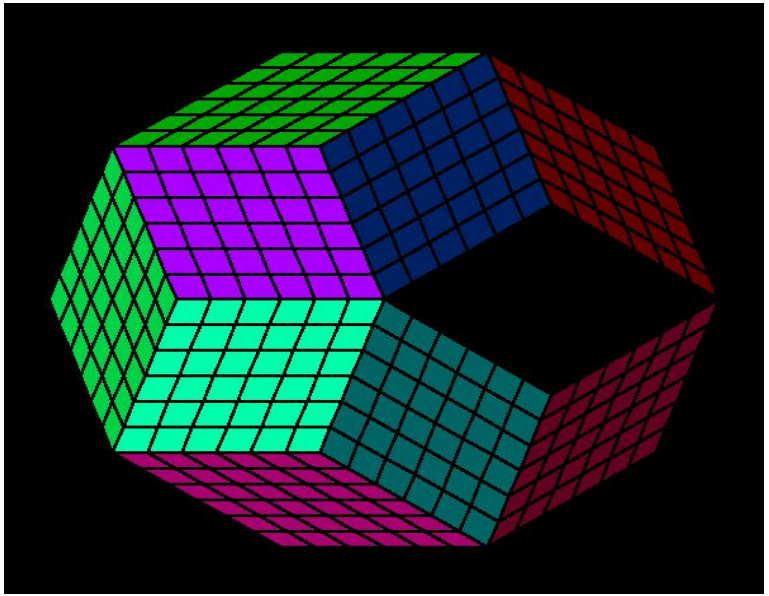
Requirements in Natural Language

- 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)

“Natural Language Processing for Requirements Engineering : The Best is yet to Come”, Dalpiaz et al. 2018

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 (eg. 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

Results

P_1 vs. P_3		P_1 vs. P_3'		P_1 vs. P_2		P_3 vs. P_3'		P_2 vs. P_3		P_2 vs. P_3'	
bieten	0.9874	bieten	0.9884	system	0.9717	verbund	0.9940	bieten	0.9705	ermöglichen	0.9696
möglichkeit	0.9874	möglichkeit	0.9881	bieten	0.9690	service	0.9938	möglichkeit	0.9705	möglichkeit	0.9693
nutzer	0.9771	nutzer	0.9770	möglichkeit	0.9690	möglichkeit	0.9933	nutzer	0.9691	bieten	0.9689
fähig	0.9422	fähig	0.9622	nutzer	0.9639	bieten	0.9932	ermöglichen	0.9584	bereitstellen	0.9685
chat	0.9397	konfigurieren	0.9618	stanag	0.9588	nutzer	0.9931	bereitstellen	0.9510	nutzer	0.9655
⋮		⋮		⋮		⋮		⋮		⋮	
mission	0.9177	anzeigen	0.9052	ermöglichen	0.9343	informationen	0.9251	services	0.9008	maximal	0.9101
automatisch	0.8941	durchzuführen	0.9012	bereitstellen	0.9258	endgerät	0.9148	gemäß	0.8966	beim	0.8966
informationen	0.8637	entsprechend	0.8961	informationen	0.9250	clients	0.8703	mobilen	0.8911	services	0.8929
service	0.8625	nutzung	0.8899	nato	0.9070	planning	0.8701	informationen	0.8730	priorisierung	0.8717
durchzuführen	0.8540	service	0.8750	service	0.8978	durchzuführen	0.8436	plattform	0.8621	plattform	0.8269

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)?

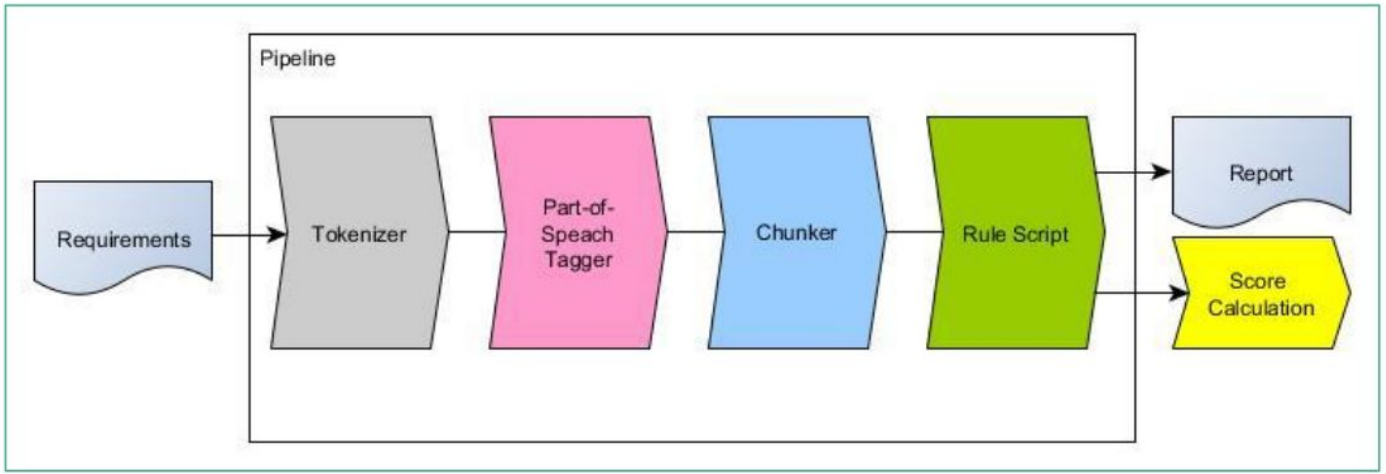
"Detection of Defective Requirements using Rule-based scripts", Hasso et al., 2019

Kinds of defects

Defects and their occurrences in the Test Corpus

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

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

	True Positive	False Positive	False Negative	Precision	Recall	F1
Total	108	40	38	0.73	0.74	0.753
Empty Verbphrase	23	13	12	0.639	0.657	0.648
Incomplete Condition	0	5	4	0.0	0.0	0.0
No Atomicity	66	22	12	0.75	0.846	0.795
Passive	17	0	0	1.0	1.0	1.0
Quantor	1	0	3	1.0	0.25	0.4
Vague Adjective	1	0	7	1.0	0.125	0.222

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)

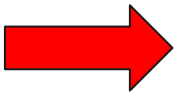
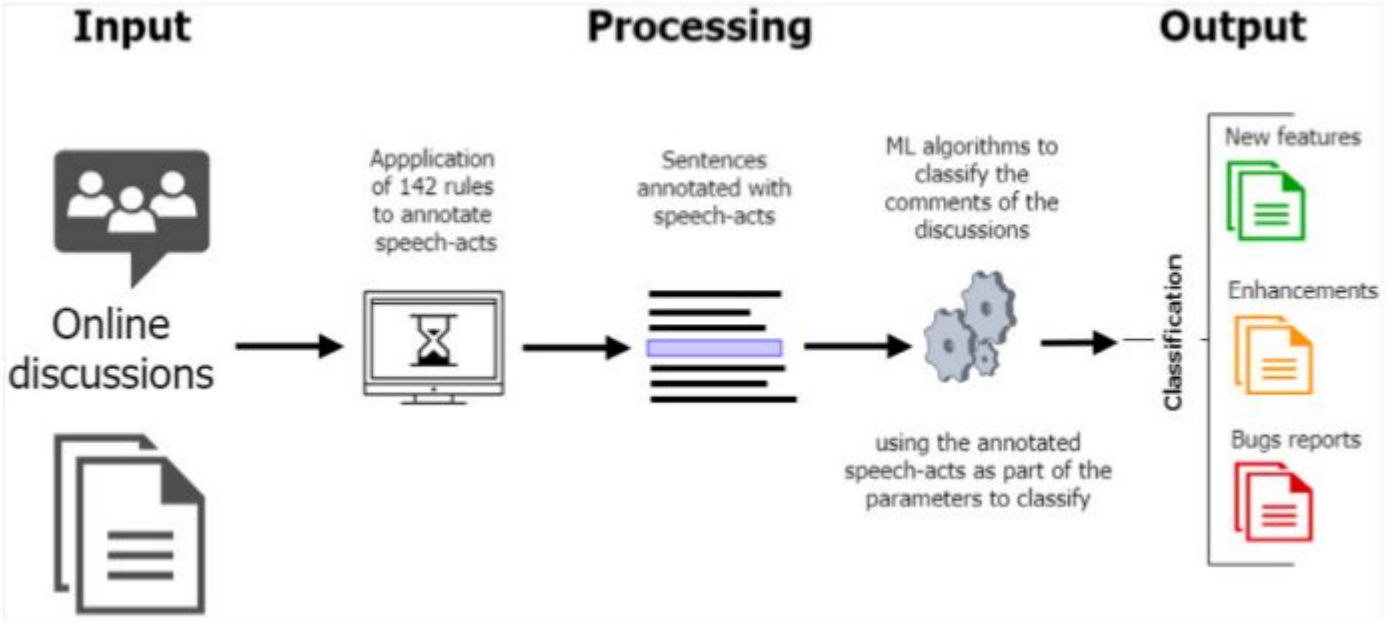


Table 1: RE research using NLP techniques at SE-FBK

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

"Research on NLP for RE at the FBK-Software Engineering Research Line : A Report", Meshesha Kifetew et al., 2019.

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

RE 4 NLP

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...

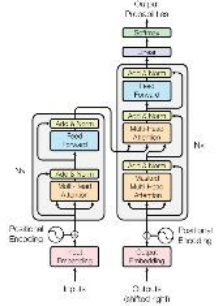
- Earlier (Pre mid-2018), solutions proposed were specific to a downstream task
  - State-of-the-art for a dataset or at max a set of datasets
- The models were usually trained from scratch over pre-trained word vectors
- RNNs and CNNs were widely used
- 2018 onwards Pre-trained models :
  - ULMFiT, 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
[2] “BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding”, Devlin et al., 2018
[3] “Improviong Language Understanding with Unsupervised Learning”, Radford et al., 2018
[4] “XLNet : Generalized Auto-regressive pre-training for Language Understanding”, Yang et al., 2019

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
[2] "SuperGLUE : A Stickier Benchmark for General-Purpose Language Understanding Systems", Wang et al., 2019
[3] "Optimal Brain Damage", LeCun, 1998

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

System	Citation	Performance
System A	Smith et al. 2018	**76.05**
System B	Li et al. 2018	75.85
System C	Petrov et al. 2018	75.62

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
[2] "Winner's Curse? On pace, progress and empirical rigor", Sculley et al. 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 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