Requirements Classification with Interpretable Machine Learning and Dependency Parsing

F. Dalpiaz[1], D. Dell'Anna[1], F. B. Aydemir[2], S. Çevikol[2]

1. RE-Lab, Utrecht University, Utrecht, The Netherlands
2. Boğaziçi University, İstanbul, Turkey

RE 2019

Requirements Classification

The system shall refresh the display every 60 seconds.

Requirements Classification

The system shall refresh the display every 60 seconds.

functionality quality

Requirements Classification

The system shall re

Requirements Eng (2007) 12:103–120
DOI 10.1007/s00766-007-0045-1

ORIGINAL ARTICLE

Automated classification of non-functional requirements

Jane Cleland-Huang · Raffaella Settimi ·
Xuchang Zou · Peter Solc

Received: 3 November 2006 / Accepted: 22 February 2007 / Published online: 23 March 2007

Abstract This paper describes a technique for automating the detection and classification of non-functional requirements related to properties such as security, performance, and usability. Early detection of non-functional requirements enables them to be incorporated into the initial architectural design instead of being refactored in at a later date. The approach is used to detect and classify stakeholders' quality concerns across requirements speci-

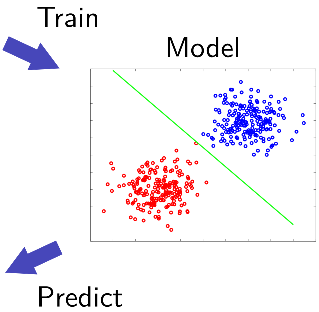
is useful for supporting an analyst in t
manually discovering NFRs, and furt
to quickly analyse large and complex
search for NFRs.

Keywords Non-functional requirem
Quality requirements · Classification

Automated requirements classification

A supervised learning task

		Class
Requirement	**Features**	**Functional**
… *print a report* …	…	Yes
… *save the page* …	…	Yes
… *every three days* …	…	No



… *refresh the display* …	…	**?**

State-of-the-art automated requirements classifiers[1]

Hundreds of features at word level:
text n-grams, Part-Of-Speech n-grams, ...

Requirement	**print**	**report**	**(print, a)**	**page**	**VB**	**DT**	**(VB, DT)**	...	**Functional**
... *print a report* ...	Yes	Yes	Yes	No	Yes	Yes	Yes	...	Yes
... *save the page* ...	No	No	No	Yes	Yes	Yes	Yes	...	Yes
... *every three days* ...	No	No	No	No	Yes	No	No	...	No
... *refresh the display* ...	No	No	No	No	Yes	Yes	Yes	...	**?**

[1] e.g., (Kurtanović *et al.*, 2017), (Winkler *et al.*, 2016), (Knauss *et al.*, 2011)

State-of-the-art automated requirements classifiers[1]

Hundreds of features at word level:
text n-grams, Part-Of-Speech n-grams, ...

Requirement	print	report	(print, a)	page	VB	DT	(VB, DT)	...	Functional
... *print a report* ...	Yes	Yes	Yes	No	Yes	Yes	Yes	...	Yes
... *save the page* ...	No	No	No	Yes	Yes	Yes	Yes	...	Yes
... *every three days* ...	No	No	No	No	Yes	No	No	...	No
... *refresh the display* ...	No	No	No	No	Yes	Yes	Yes	...	**?**

High performance (precision and recall up to $\sim 90\%$)

[1] e.g., (Kurtanović *et al.*, 2017), (Winkler *et al.*, 2016), (Knauss *et al.*, 2011)

State-of-the-art automated requirements classifiers[1]

Limitations

L1 Absence of validation **benchmarks**

- Slicing same dataset for training and testing

L2 **Dichotomous** classification Functional vs Quality

- How to cope with "*I want to print a report every 30 seconds*"?

L3 Low **interpretability** and **generality**

- Many low-level features are used to decide the class

[1] e.g., (Kurtanović *et al.*, 2017), (Winkler *et al.*, 2016), (Knauss *et al.*, 2011)

Annotation of 1500+ requirements from 8 datasets

Addressing dataset scarcity (L1) and requirements classes (L2)

Requirements can have *both functional and quality aspects* (Li *et al.*, 2014).

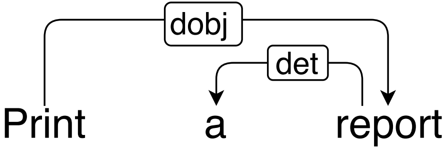
4 types of requirements: **OnlyF**, **OnlyQ**, **F+Q**, **None**

Dataset	**Domain**	**Public**	**Reqs**
PROMISE	Misc	Yes	625
ESA Euclid	Satellite	No	236
Helpdesk	IT	No	172
User mgmt	IT	No	138
Dronology	UAS	Yes	97
ReqView	IT	Yes	87
Leeds library	IT	Yes	85
WASP	IT	Yes	62
Total			1,502

Dependency Types: fewer and higher-level features

Addressing low generality and interpretability (L3)

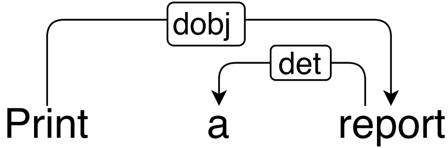
Dependency types describe the **relationship** between (possibly **non-contiguous**) words.



Dependency Types: fewer and higher-level features

Addressing low generality and interpretability (L3)

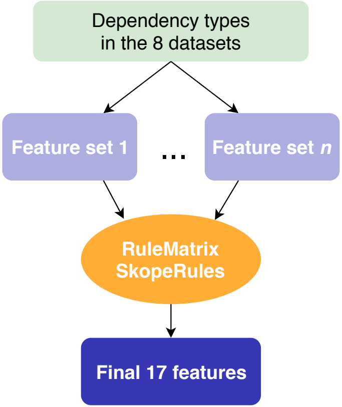
Dependency types describe the **relationship** between (possibly **non-contiguous**) words.



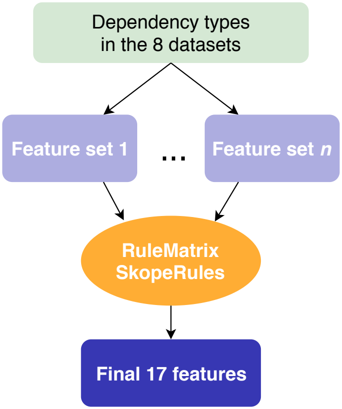
12 word-level features:
print, a, report, (print, a), (a, report), (print, a, report), VB, DT, NN, (VB, DT), (DR, NN), (VB, DT, NN).

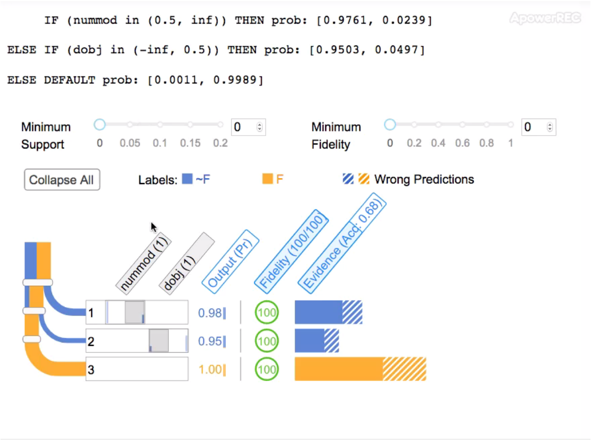
Only 2 dependency types:
dobj and *det*

Feature engineering with Interpretable ML



Feature engineering with Interpretable ML





Experimental Setting

- **Reconstruction** of (Kurtanović and Maalej, 2017) word-level high-dimensional classifier
- **Comparison** of the reconstruction against our 17 higher-level features
- **Training** always on PROMISE NFR dataset (for comparison purposes)
- **Testing** on different slicing of PROMISE NFR & 7 industrial datasets
- Experiments for **F**, **Q**, **OnlyQ**, **OnlyF**, **F+Q** requirements

500 word-level features vs 17 higher-level features

Comparison with reconstruction of (Kurtanović *et al.*, 2017) classifier

Similar performances:

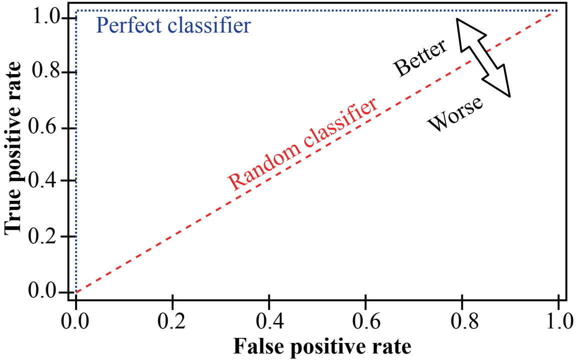
- On **PROMISE NFR**:
  precision and recall worsen, but the degradation is limited (circa -0.1).

- On the **industry datasets**:
  recall improved for **F** ($+0.16$); precision improved for **OnlyQ** ($+0.31$) and **OnlyF** ($+0.28$).

Higher level features provide more generality

Comparison with reconstruction of (Kurtanović *et al.*, 2017) classifier

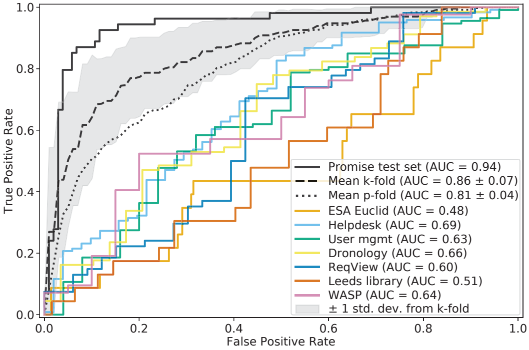
ROC plot to study performance of classifier.



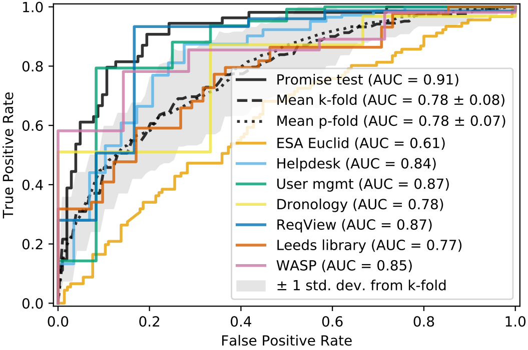
Higher level features provide more generality

Comparison with reconstruction of (Kurtanović *et al.*, 2017) classifier

Classification of **OnlyF** requirements (ROC plot).



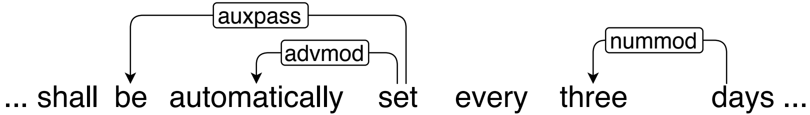
(a) SVM 500 word-level features



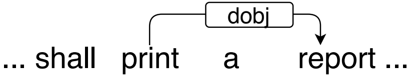
(b) SVM 17 higher-level interpretable features

Some interpretable findings with the 17 identified features

- *Adverbial modifiers*, *numerical modifiers*, *passive sentences* typically indicate **qualities**



- *Direct objects* typically indicate **functional** aspects



Conclusion and Implications on RE practice and research

- Annotation of **1500+ requirements** from 8 datasets
- Openly available classifiers
- Few **higher-level linguistic dependencies** as features for requirements classification instead of many word-level hard-to-interpret features.

Conclusion and Implications on RE practice and research

- Annotation of **1500+ requirements** from 8 datasets
- Openly available classifiers
- Few **higher-level linguistic dependencies** as features for requirements classification instead of many word-level hard-to-interpret features.

  Practical uses:

  - Bootstrapping a classifier with limited data
  - Interpretability and guidelines for requirements authoring
  - Approach appplicable also to: bug vs features vs praises, requirements vs information, qualities categorization, etc.

Thank you for your attention.

Download our artifacts!

Limitations of our approach

- Additional validation is needed
- Training on PROMISE (for comparison purposes)
- Hard(er) to determine high level features that distinguish **qualities**
- Reconstruction of the state-of-the-art to the extent the paper describes it