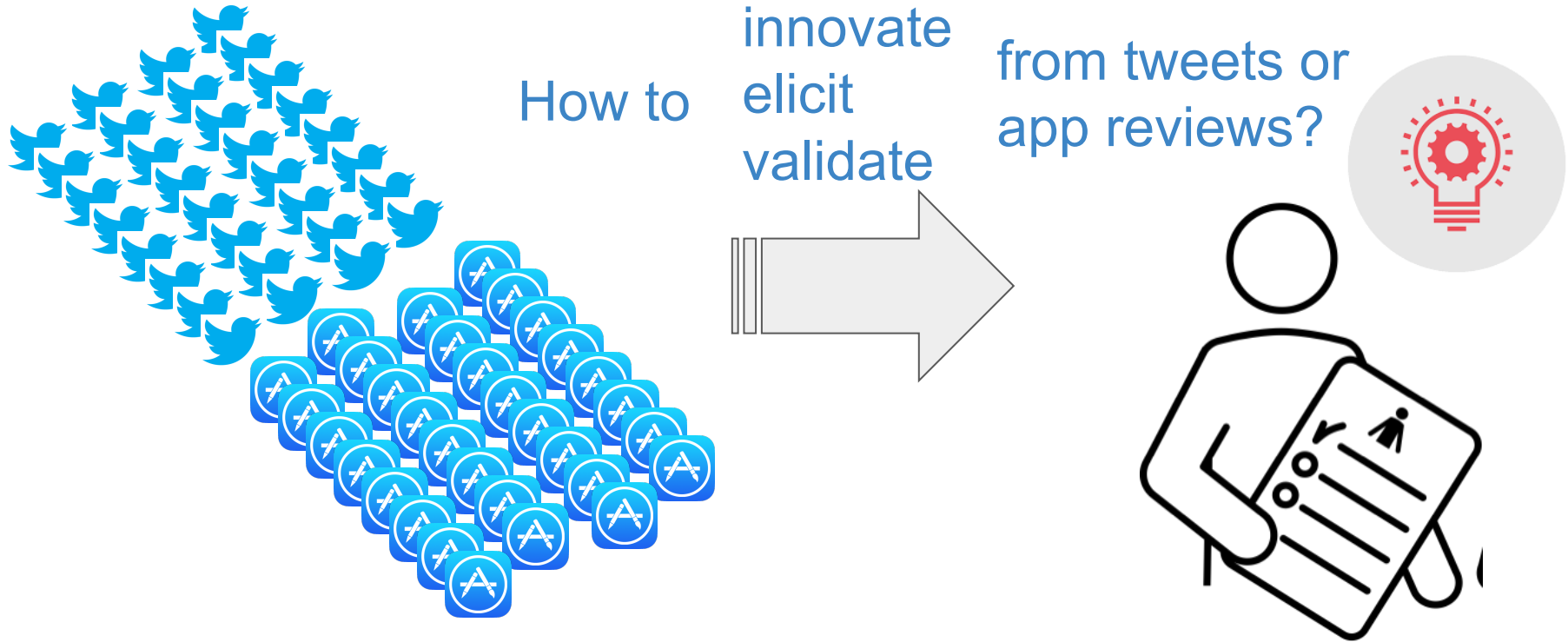


Mining and Analyzing User Feedback

Catrin Loch & Florian Fischer

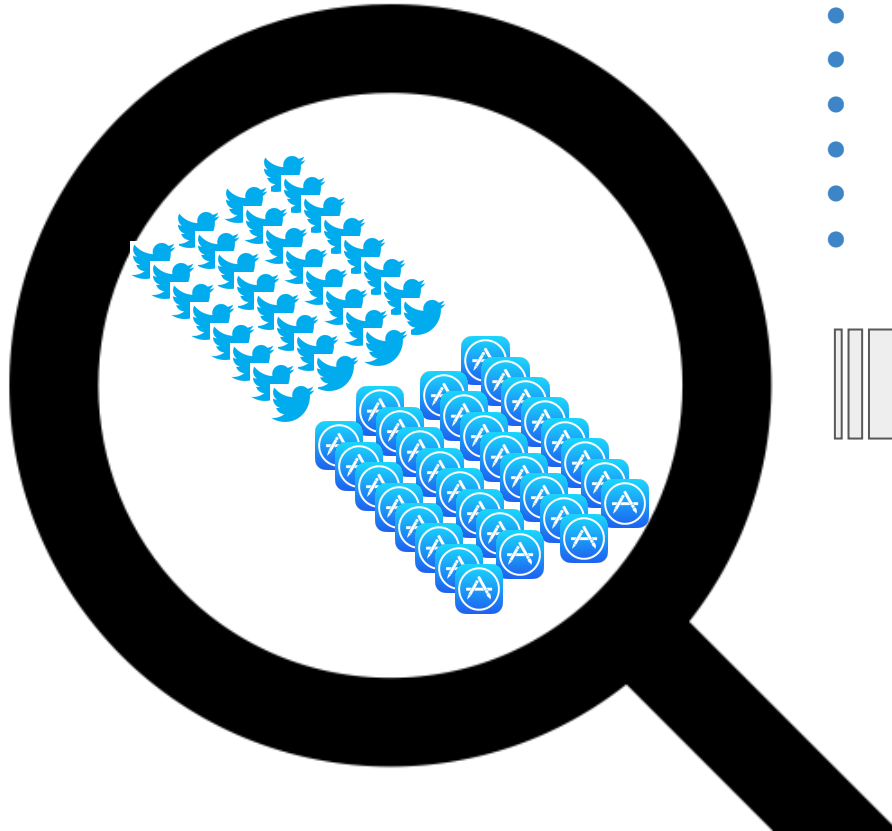
Social Computing and User Feedback in Requirements Engineering



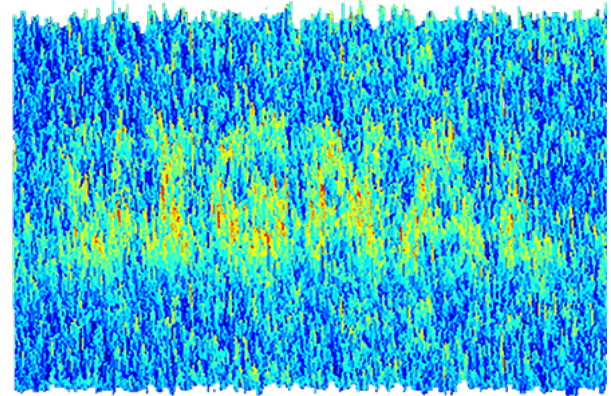
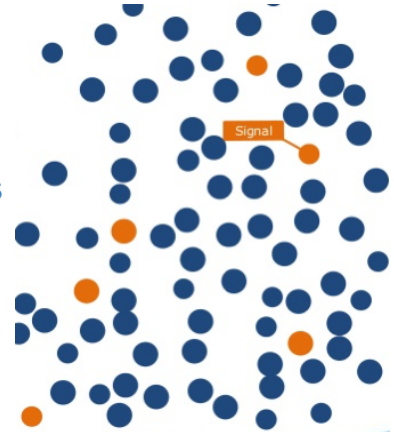
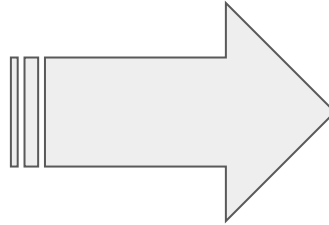
Social Computing and User Feedback in Requirements Engineering

- Social computing and user feedback are very rich sources for eliciting or validating requirement. The richness of these sources come from the fact that very large amounts of people are involved in these activities, resulting in enormous amounts of raw data.
- In order to process the raw data created by both social computing und user feedback, mining techniques have been developed (or still are in development) to classify, analyze and finally “make sense” of this mass of information.

Mining and Analyzing User Feedback



- descriptive statistics
- content analysis
- sentiment analysis
- machine learning techniques
- probabilistic techniques
- ...



Mining and Analyzing User Feedback

- The need for mining and analyzing user feedback comes from the potential insights that might help developers to create increased market value but also from the fact that a feedback alone might not be insightful or even representative of the user's community. In this sense, using the image of radio communication, the role of mining of analysing user feedback is to separate signal from noise.
- The main challenge for mining techniques today is to classify in a way that will produce a valuable output for analysis. Even though there is a broad consensus about the richness of these sources, the methods for classifying (descriptive statistics, content analysis, machine learning techniques, probabilistic techniques, etc.) are still to be evaluated.

How can user feedback in app reviews be analyzed automatically? (Maalej & Nabil, 2015)

1. real reviews from app stores
(1.1 million reviews from 1100 apps in AppStore + 150'000 from GoogleStore half paid, half free)
2. manually analysed representative truth set and train techniques
3. implement the classifiers on real reviews
4. classification is evaluated

Methods	Accuracy
string matching (keyword)	0.27-0.74
bag of words (dictionary and frequency)	0.66-0.86
natural language processing (stemming, stop words)	0.62-0.9
metadata (rating, verb tense and review length)	
sentiment scores (SentiStrength)	
naive Bayes (supervised learning)	0.5-0.75
decision tree (supervised learning)	0.42-0.66
multinomial logistic regression (supervised learning)	0.33-0.57

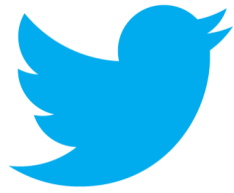
How can user feedback in app reviews be analyzed automatically?

Introduction: Reviews are an important part and a predictor for app success through increased visibility. Users use reviews as navigator. Beside the users' benefit, developers gain information on bugs or issues, summary of user experience, request or ideas for new features. But not all reviews are useful and tools are needed to sort them out.

Method: Design, evaluate and compare different classifiers with the task of classifying into the four categories: Bug report, feature request, user experience, ratings. Classifiers: string matching (keyword), bag of words (dictionary and frequency), natural language processing (stemming, stop words), metadata (rating, verb tense and review length), sentiment scores (SentiStrength), supervised learning (binary classification, naive Bayes, decision tree, multinomial logistic regression).

Research question: assess the **accuracy** of the classifications techniques

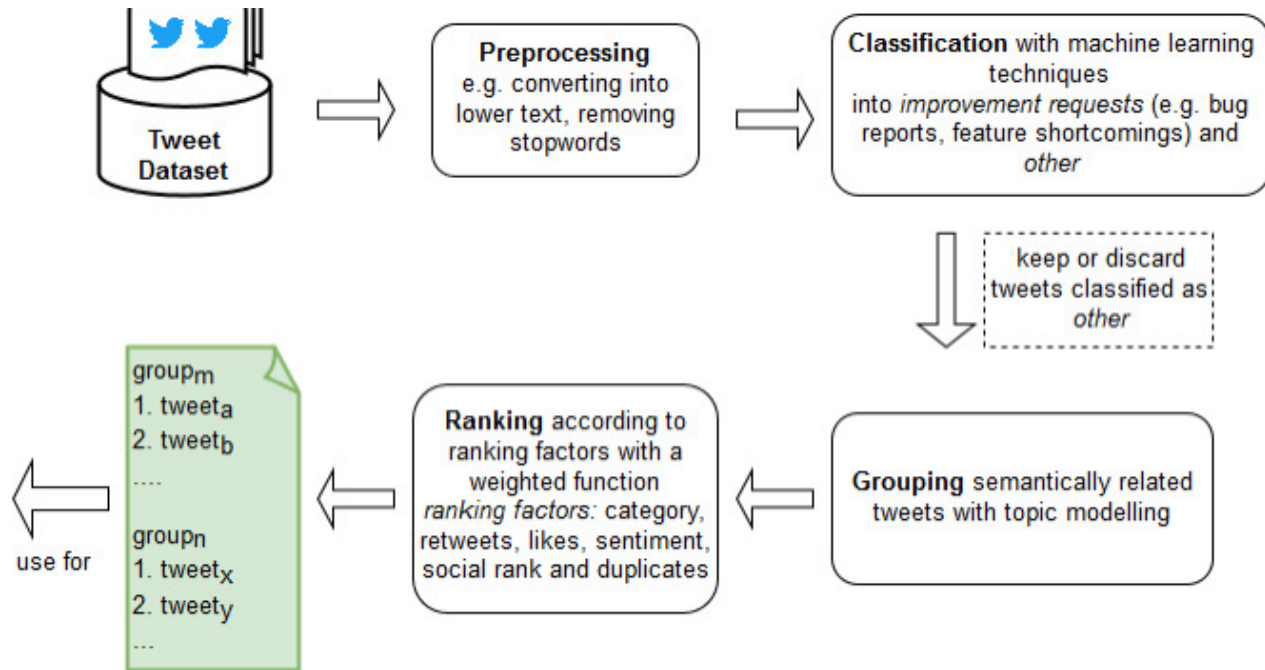
Data processing: 1) real reviews from app stores (1.1 million reviews from 1100 apps in AppStore + 150'000 from GoogleStore; half paid, half free), 2) manually analysed representative truth set and train techniques 3) implement the classifiers on real reviews, 4) classification is evaluated



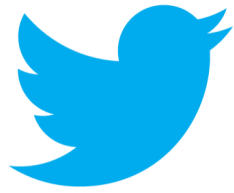
How can Twitter be used as a channel for user feedback and how can tweets be analyzed automatically?

(Guzmann, Ibrahim & Glinz, 2017)

- ALERTme
- classify, group & rank tweets
- identify requirements



(1) **filter tweets** that do not contain information relevant
(2) simplify **identifying requirements** or other information relevant for software evolution
(3) obtain information about the **relevance of the tweets**, e.g. use for prioritizing requirements elicited from the tweets



How can Twitter be used as a channel for user feedback and how can tweets be analyzed automatically?

Goal: investigate to which extent the techniques used in research about mining of user feedback from app stores can be applied to tweets about software applications

Why Tweets? Twitter allows bidirectional communication between e.g. user and developer (advantage over e.g. App Stores)

Approach: ALERTme, automatically classify, group and rank tweets about software applications. ALERTme enables the exploitation of Twitter as a feedback channel for information relevant to software evolution, including end-user requirements.

Results: applying techniques that are the same or similar to those applied on app store reviews for classification and groupings, as well as novel ranking function (ALERTme) are promising directions for mining user feedback from Twitter data

Benefits & Limitations

Benefits	Limitations
- high number, unbiased and natural (no expression constraints) - real user, real user problems, real user suggestions - being up-to-date - social networks as feedback channels, easy to use	- expensive and time consuming (when not automated) - methods are still in development - gap between what user wanted to express vs. what researchers understood - filtering of what is important/useful and what not → subjective - not organized, unstructured, random, natural language

Example: discussion in a forum

- find topic for prototype with up-to-date issue

Questions for discussion

- Who of you had already heard of using social networks as feedback channel?
- Can you think about other social networks which could be used for eliciting requirements?
What are their benefits and limitations?
- What is your approach when you have an issue with a software?
Have you used Twitter to communicate your problem?
Will you use Twitter in the future?

References

Guzmán, E., R. Alkadhi, N. Seyff (2016). A Needle in a Haystack: What Do Twitter Users Say about Software? In *24th IEEE International Requirements Engineering Conference (RE'16)*, Beijing, China. 96-105.

Guzmán, E., M. Ibrahim, M. Glinz (2017). A Little Bird Told Me: Mining Tweets for Requirements and Software Evolution. In *25th IEEE International Requirements Engineering Conference (RE'17)*, Lisbon, Portugal. 11–20.

Maleej, W., & Nabil, H. (2015) Bug Report, Feature Request, or Simply Praise? On Automatically Classifying App Reviews. In *23th IEEE International Requirements Engineering Conference (RE'15)*, Ottawa, Canada. 116-125.