Content Analysis of Conclusions of Empirical Studies in Software Engineering

Bachelors Thesis – Opening Presentation

Lars Spitzlay

Table of Contents

Original Goals

- Evaluate the quality of empirical studies in SE
- Are the conclusions „good“?
  - Do they contain generalization?
  - Is the generalization appropriate?
  - Do they just repeat results?

Goals

- Report on common structures in conclusions
  - Which codes often correlate
- Provide valuable insights for future lectures/tutorials on
  - Structure of the tutorials
  - Code system used for the Content Analysis

Content analysis – a quick overview

- Qualitative-quantitative method in empirical research
  - Qualitative input data, quantitative results
- Take or create a coding system
  - i.e. by open coding
- Search for all occurrences of each code in your material
- Count those occurrences
- This process should be conducted at least twice, by independent reviewers
- Calculate Inter-reviewer agreement
- Material is usually in written form or first transcribed

Lecture/Tutorial

- 10 Students working on the project
- Open Coding on some papers => MAXQDA
- Developing a unified code system
- Identify relevant papers from ICSE 2021 => 59 papers
- Code the conclusion + relevant parts
  - 18 papers per student
  - >=3 students per paper
- Data validation, clustering and code correlation
- Presentation and report

Final code system (excerpt)

- Generalization
  - Generalization_classification_by_reviewer
    - Adequate_generalization
    - Too_little_generalization
    - Too_strong_generalization
  - Generalization_degree
    - For_all_generalization
    - Limited_generalization
    - No_generalization
  - Generalization_kind
    - Recommendation_generalization
    - Speculative_generalization
    - Universal_statement_generalization
  - Generalization_relevance_evaluation_by_authors
    - …

Agreement by codes – Top 10

Code	Agreements	Disagreements	Agreement in %
state_first_of_a_kind	42	7	86
own_further_work	78	16	83
anyone_further_work	72	32	69
subsequent_further_work	72	40	64
repeat_research_process	224	141	61
no_classification_of_importance_further_work	55	35	61
mention_generalization	161	135	54
promising_further_work	39	35	53
universal_statement_generalization	104	96	52
unclassified_risk	4	5	44

Agreement by codes – Bottom 15

Code	Agreements	Disagreements	Agreement in %
relevant_result_without_specification	43	91	32
speculative_generalization	20	46	30
addressed_risk	2	5	29
relevant_result_for_research	17	56	23
too_strong_generalization	15	52	22
unclear_relevance	0	20	0
too_little_generalization	0	18	0
relevant_result_for_other_group	0	14	0
state_result	0	8	0
someone_further_work	0	7	0
untreated_risk	0	6	0
other_kind_further_work	0	4	0
distant_further_work	0	4	0
unimportant_further_work	0	1	0
major_risk	0	1	0

Example Sentence

“Our study shows that the use of bounding functions coincides with use scenarios where safety concerns of UAVs are addressed by UAV software developers”

Example Sentence

“Our study shows that the use of bounding functions coincides with use scenarios where safety concerns of UAVs are addressed by UAV software developers”

Codes used + reviewers:

- mention_generalization: a+b
- universal_statement_generalization: a+b
- relevant_result_for_practitioner: a
- too_strong_generalization: b
- no_generalization: c

Data quality

- Agreement between different reviewers limited (about 30%)
- Random tests show varying coding quality by most reviewers

=> Makes it hard to make strong statements about anything

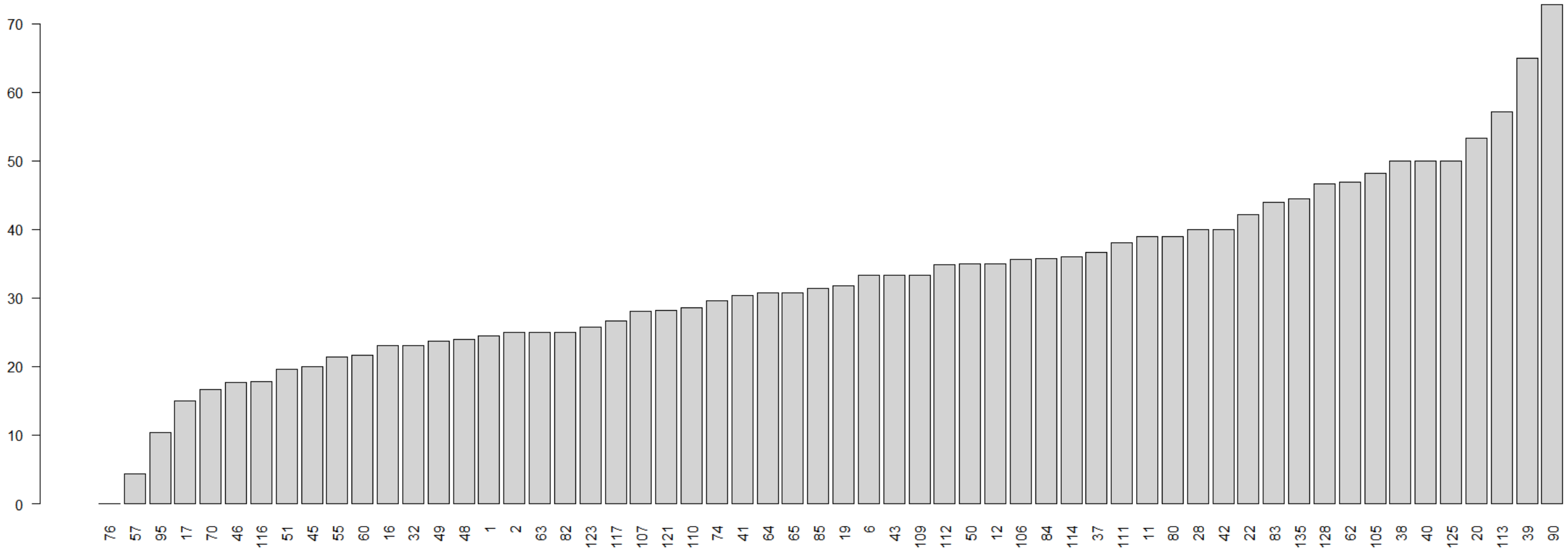
=> Can we improve data quality and salvage the data somehow?

Data quality improvements

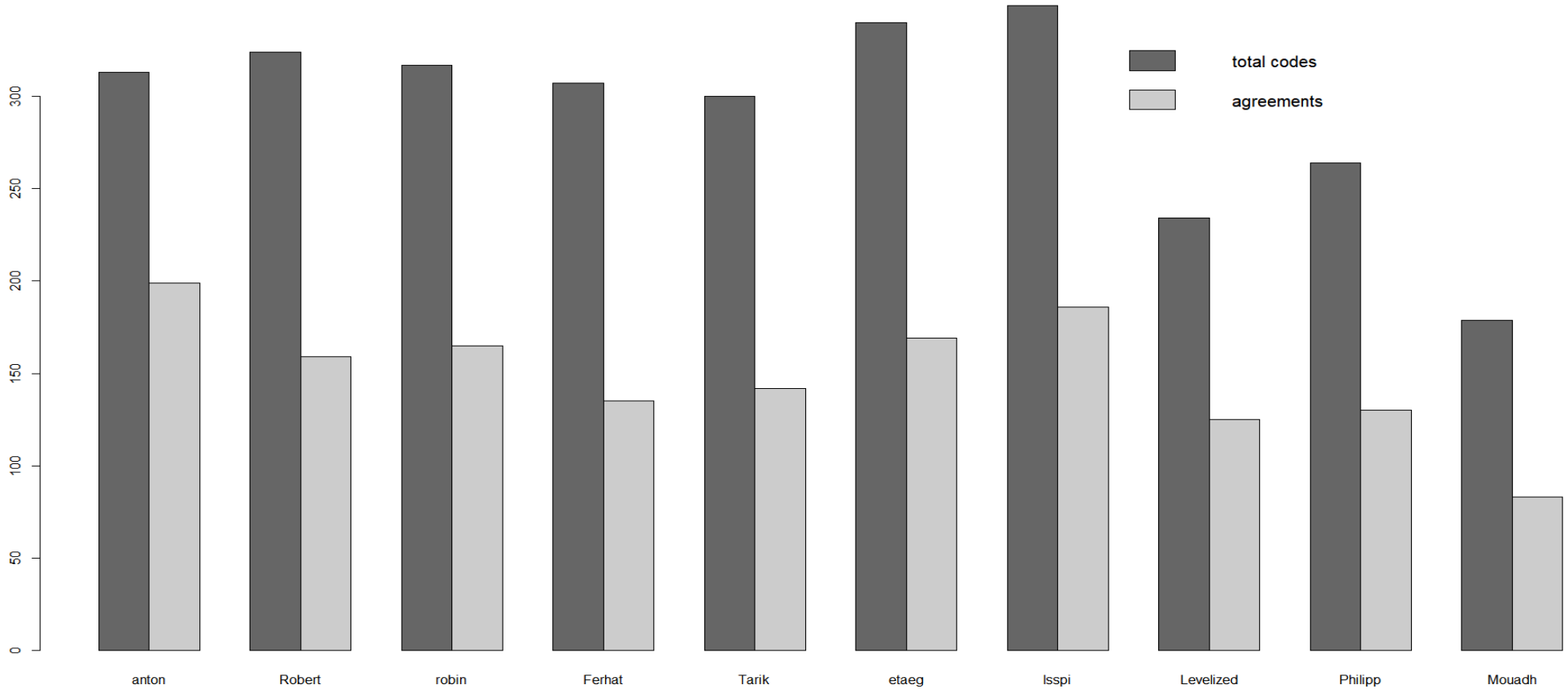
- Only consider conclusion
- Merge some codes with low agreement that fit together
  - i.e for_all_generalization + limited_generalization = mention_generalization
- Improve data validation/cleansing
  - Use a sequence matcher to find agreement
  - Check for outliers

=> Even after several improvements, agreement is still below 50%

Agreement over different papers (in %)



Codes and agreements per reviewer



Related Work

- It‘s hard to find any closely related work, my thesis is rather abstract
- Most „meta-work“ on empirical studies in SE focuses more on the content itself or on the methodology

- Perry D, et al (2000). “Empirical Studies of Software Engineering: A Roadmap”
- Höfer, A., Tichy, W.F. (2007). “Status of Empirical Research in Software Engineering”

Propositions for future lectures/tutorials

- Provide more time for the students
- Reduce the number of studies used
  - Reduces stress, improves quality of the data collected
- Reduce the complexity of the code system
  - Simplify by having more “black and white” codes
    => either code a must be used OR code b
- Have a stricter guide to coding
  - If a code is used, certain other code sub-trees must be covered as well

Thanks for your attention!

- Any questions?
- Any feedback?