TECHNISCHE UNIVERSITÄT DRESDEN

ACADEMIC SKILLS IN COMPUTER SCIENCE

Lecture 5: Planning Your Research

Markus Krötzsch
Knowledge-Based Systems

TU Dresden, 30th April 2019

Review: Exploiting report structure in reading

General structure of research reports:
1. **Title, authors, affiliations**
2. **Abstract**
3. **Introduction**
4. The Research: details, methodology, results
5. **Conclusion**
6. **References**

Read only as far as it serves your current goals

Re-read sentences, paragraphs, sections, and whole papers as needed

Review: Reading results

For every work you read, answer at least the following questions:
(A) Where and when was the work published?
(B) Who are the authors, and what are their affiliations?
(C) Which problem did they address? What was their research question?
(D) What did they propose as a solution?
(E) Which methods did they use to validate their research?
(F) What was the outcome? What did they learn?
(G) Strong points of the work?
(H) Weak points of the work?
(I) Where would you place this work in comparison to its area/your own research?
(J) What did you find hard to understand? Which further information do you need?

Beginners often do the following:
- Find a PDF online, without taking note about who, when and where.
- Spend all time & effort on understanding the proposed solution (D).

Goals for today

Learning goals of this lecture:
(1) Appreciate the importance of defining the right scope for a research project
(2) Judge the quality of research questions & hypotheses
(3) Learn about common problems in the scope of research
(4) Get an overview on how to structure your written report

Planning Your Research

From research direction to hypothesis

Research direction: a concrete area of investigation

- Query answering under description logic constraints
- Human-computer interfaces in cars
- Compact data structures for efficient algorithms

Research problem: a specific question

- "Is conjunctive query answering over OWL 2 ontologies decidable?"
- "How do voice-controlled smart assistants impact driving performance?"
- "How can compactly serialised trie structures be updated efficiently?"

Research hypothesis: a proposed answer that can be (in)validated

- "Conjunctive query answering over OWL 2 ontologies is decidable but complete for non-elementary time."
- "Performance in critical driving situations is significantly affected when smart agents demand additional information from the driver."
- "The proposed new data structure achieves better write performance than previous approaches while maintaining comparable read and storage characteristics."

What is a good hypothesis?

A good research hypothesis is:

- precise (not vague or unclear)
- useful (can give rise to transferable predictions)
- testable (can be reasonably validated or invalidated)
- as general as possible, as specific as needed (to solve research problem)

The last requirement is related to Occam's Razor, the guideline to prefer simple explanations over complicated ones.

Note: There are some types of research-related publications without such a hypothesis:

- Surveys: reports that give a comprehensive, balanced, and well-informed overview of a research area or sub-field
- Infrastructure & tools: reports about new tools that are valuable for research (emphasis on practical utility rather than on fundamental insights)

$\rightsquigarrow$ venues will ask for such contributions explicitly if desired

Hypotheses in design-oriented research

Many computer science works are design-oriented:

- Some artefact is designed (a software, an algorithm, a model, a mathematical theory, a methodology, etc.)
- Hypothesis: "This is a 'good' design"

The research hypothesis is often not stated, but implicit in what the work claims to show.

Examples:

- An algorithm is shown to be correct and to run in worst-case polynomial time (hypothesis: a computing problem can be solved efficiently)
- Software is shown to be faster than existing tools (hypothesis: a new design can lead to performance gains over the state of the art)
- A new methodology is tested in a practical case study, and shown to lead to better results than previous approaches (hypothesis: the new method can improve quality in practical applications)

Design-orientation can lead to weak hypotheses

Design-only research:

- researchers put significant effort into a complex design and its implementation,
- they therefore feel that they made a big contribution to research,
- but no strong hypothesis can be found behind all this work.

Separating research from other work can be tricky in computer science:

- not every problem one can solve is a research problem
- not every activity that somebody finds useful is research

Note: There are special (research) venues that ask for reports about artefacts that can be useful to research

Design-orientation & weak hypotheses: example

Example: A complex software product with a user interface, web API, and database backend is developed. It is reported that the software was used in some project of the authors.

- Vague: what is actually claimed here? Several different things about several research fields? Or is the novelty in the combination?
- Not transferable: the more complex the software, the harder it is to transfer insights to other settings. Even if the software is great, it is impossible to say what really caused this.
- Not testable: the general overall quality of software is very hard to evaluate (simple tests of one component are not enough)
- Overly specific: the focus on a single implementation hides the generality of individual innovations (if there are any)

Ways out

Elaborate artefacts can be an excellent basis for research – if the hypotheses are chosen appropriately. Some approaches for turning complex artefacts into research:

- Select one contribution and thoroughly study it in separation

  Example: Focus on user interface; describe design ideas; conduct a laboratory study to compare it with other designs; conduct a field study with real users in the context of the whole software product

- Find over-arching research hypothesis motivating several aspects of your design

  Example: Conjecture that ontology reasoning can be faster by better, parallelised algorithms; design several techniques to achieve this; evaluate impact of each in isolation; show that the overall system outperforms competitors

- Develop an application-oriented or product-oriented research hypothesis

  Example: Argue that logic-based AI can be beneficial in medical diagnosis; conduct an extensive field study with MPs using your tool; evaluate their feedback as well as objective measures of work quality

Lack of focus can lead to weak hypotheses

The general rule for papers and smaller theses: one contribution per report

Salad of weak results

- researchers often combine several weak contributions into one paper,
- hoping that they will somehow “add up” under a common high-level motivation,
- but mixing several bad ingredients is not a recipe for good outcomes.

Why mixing unrelated contributions does not work:

- several inconclusive parts can hardly form a conclusive whole
- depth is more important than breadth in individual research works
- mixing unrelated concerns means that readers care only about part of the work

Note: one report can study several hypotheses, if they are closely related and contribute to the same research problem

How to do research with a broad focus

Wrong: Focus paper on end-user problems; present three pages on UI design and three pages on performance improvements by parallelisation.

- Most likely, neither contribution is analysed & evaluated properly
- Experts in UI resp. parallel algorithms can only appreciate three pages each

It is always possible to research one aspect in depth, but it can also be feasible to maintain a broad focus, if a matching hypothesis is chosen:

Solution 1 (interdisciplinary): Argue that better UI concepts are prevented by computing demands; solve this problem by co-designing a new UI and tailor-made parallel algorithms; evaluate how faster response times improve user experience

Solution 2 (applied): Focus on a relevant, general application area; argue that new application designs are needed to bring research/technology advances to this field; conduct several case studies to evaluate contribution.

$\rightsquigarrow$ both strategies can make the whole work relevant to UI and implementation experts

Superficial problem-orientation can lead to weak hypotheses

Not just with complex designs it is easy to mistake a solution (for one problem) for a research contribution:

Black box research [Zobel: Writing for Computer Science; 2015]

- researchers propose an algorithm whose properties are poorly understood ("black box"),
- and show that it can solve a problem better than some baseline (e.g., better than random),
- but there are no transferable insights (how good is this in other cases? which cases work well? which parts of the design caused this?)

Main weakness: authors are satisfied too easily by some apparent success rather than asking what is behind it:

- strong research is not about solving a particular problem instance,
- but about learning how to best solve problem instances in general

Superficial problem-orientation: examples

This phenomenon is typical in many areas where

1. solutions are hard to analyse and fully understand
2. a small number of known benchmarks are the main tool for evaluating success

Example: Areas that use machine learning and other complex data analysis processes are prone to this fallacy. Typical pattern:

- New algorithms are proposed, often by combining/modifying known ideas
- A known benchmark dataset is used to show improvements over previous approaches

Author: "We significantly improve upon the state of the art!"
Expert: "You just tweaked side-conditions to tune performance on this benchmark."

Ways out: insightful analysis of new approach (why should it be better, and in which cases?); broader evaluation (do not rely on benchmarks only)

Benchmarks can lead to weak hypotheses

Possibly the most common kind of superficial problem-orientation:

Benchmark-driven research

- researchers propose an algorithm or system,
- and show that it leads to improvements over standard benchmarks,
- but do not present insights or evidence beyond the benchmark

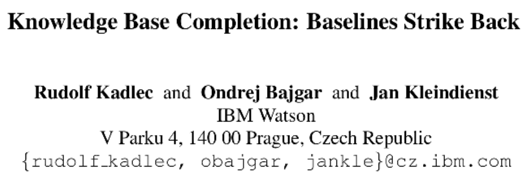
Benchmarks can become a perverse incentive:

- Original goal: enable standardised evaluation of relevant practical qualities
- Perverted goal: benchmark performance becomes main research purpose
  (improving performance on the benchmark is usually much easier than improving it in general)

$\rightsquigarrow$ benchmarks, once published, loose their predictive quality

Solution: Benchmarks must be dynamic and under continuous development

(Note: developing benchmarks can be a relevant research contribution)


Knowledge Base Completion: Baselines Strike Back

Rudolf Kadlec and **Ondrej Bajgar** and **Jan Kleindienst**
IBM Watson
V Parku 4, 140 00 Prague, Czech Republic
{rudolf_kadlec, obajgar, jankle}@cz.ibm.com

"Many papers have been published on the knowledge base completion task in the past few years. Most of these introduce novel architectures for relation learning that are evaluated on standard datasets such as FB15k and WN18. This paper shows that the accuracy of almost all models published on the FB15k can be out-performed by an appropriately tuned baseline – our reimplementation of the DistMult model. Our findings cast doubt on the claim that the performance improvements of recent models are due to architectural changes as opposed to hyper-parameter tuning or different training objectives. This should prompt future research to re-consider how the performance of models is evaluated and reported."

– R. Kadlec, O. Bajgar, J. Kleindienst: Knowledge Base Completion: Baselines Strike Back. Proc. 2nd Workshop on Representation Learning for NLP (Rep4NLP@ACL), 2017: 69-74

Note: What was shown here?

(A) A simple algorithm can be tuned to outperforming more advanced approaches on the benchmarks.

(B) The success of previous approaches was only achieved by actively tuning them to the benchmarks.

⇝ only (A) was shown, but it still seems high time to give up on these benchmarks!

Lack of planning can lead to weak hypotheses

In any case, research needs a plan:

Frankenstein's research reports

- researchers explore an area (by literature study, modelling, prototyping, evaluation, etc.),
- and combine it all into one report when they feel they spent enough time,
- but the result lacks a coherent idea or plan

⇝ lack of logical structure and clear focus

Risk factors:

- Author teams that are not coordinated well
- Deadline pressure
- Lack of expertise (more unsuccessful approaches need to be tried and discarded)

Summary: anti-patterns in research planning

Recipes for bad research:

- Design-only: Describing the design of an artefact without evaluating it sufficiently
  - often even without clear claims that could be evaluated
- Salad: Mixing small contributions without main hypothesis
- Black box: Favouring "black box" performance over transferable insights
  - Benchmark-driven research
- Frankenstein's: Presenting incoherent explorations as planned research

Other major flaws that occur in practice:

- Irrelevance: it might be valid research, but nobody cares
- Ignorance: assumptions and basic claims fail a reality check; key developments in the field are ignored

So how can we arrive at good research then?

Refining hypotheses

Research is about being optimistic and sceptical at the same time:

- Optimistic: researchers have to trust their intuition that their approach has merit
- Sceptical: researchers must focus on the weak spots of their hypothesis to improve it

The more enthusiastic you are, the more sceptical you must be.

A good approach might be an (inner) dialogue:

- Try to think of ways in which your work can be attacked by a critical adversary
- Think of possible answers: defend your hypothesis, fix your approach, or acknowledge weaknesses (understanding limitations is a strength)
- Works well with collaborators and friends, but you can also talk to a rubber duck
- A good research paper incorporates the insights from such critical analysis

Hypothesis and evaluation

Previous examples already showed the close connection between hypothesis and evaluation.

Indeed, the claim often dictates the evaluation – when choosing a hypothesis, one must be prepared to evaluate it!

Claim	Example evaluation method
Better usability	User study (with human subjects)
Faster algorithm	Comparative performance evaluation
Fundamentally better algorithm	Theoretical analysis and proof
Useful system	Relevant case study
Improved architecture	Requirements analysis

Evaluation approaches

Main types of evaluation in computer science:

- Proof: rigorous mathematical analysis that establishes a claim
- Experiment: controlled empirical study that provides evidence for a claim

Conclusive proofs and realistic experiments might be hard or impossible

Auxiliary techniques:

- Modelling: abstraction of reality or some artefact to simplify evaluation

  Example: Model security protocols as transition systems for formal analysis.

- Simulation: experimental study under synthetic conditions

  Example: To evaluate a database management system, use a benchmark that simulates characteristics of real queries.

$\rightsquigarrow$ when using models and simulations, we need to ask if they are adequate for investigating the truth of our hypothesis

Evaluation feasibility

Some valid research is not feasible since evaluation is not doable (with reasonable effort):

- Missing evaluation data/benchmarks
- Insufficient availability of users for user studies
- Complexity of subject too high for complete mathematical analysis
- No access to relevant application scenarios/use cases
- Lack of own equipment or methodological expertise
- Ethical restrictions

$\rightsquigarrow$ must be taken into account when defining research hypothesis

Planning your research: final tips

Summary of previous hints:

- Be conscious about what your hypothesis actually is
- Avoid common pitfalls and weaknesses
- Refine your hypothesis by critical enquiry
- Plan your evaluation along with your hypothesis
- Be optimistic and sceptical

Moreover: use your strengths!

- Choose work that is easier for you than for others (your “unfair advantage”)
- Take advantage of your environment (supervisor/colleagues, resources, infrastructure)
- Seek role models; read good papers
- Dare to be original (don’t follow the hype, unless for motivating readers)

Summary

Finding the proper scope for your research project is important

Weak hypotheses lead to weak research, but common fallacies are not hard to avoid

Research question, hypothesis, and evaluation must develop together

What's next?

- Research reports: structure & style
- Persuasive writing
- Typesetting with LaTeX