Passing Advice for New Researchers

March 16, 2021

Outline

1. What's Covered?

2. Who is Research For?

3. Characteristics of the Ideal Research Topic

4. Finding an Idea

5. The Research Project

What is This?

Research can be made more enjoyable and productive through a little organization.

For many, research techniques are a product of experience rather than, unfortunately, formal training.

To improve this experience, we've provided several suggestions; it's the hope these suggestions ease some of the confusions faced when beginning a research project.

Who is Research For?

Who is Research For?

EVERYONE! No...seriously...*everyone can do it*

- **Research should facilitate what the researcher already enjoys!**
- Research should be fun! It does **not** need to be arduous and/or rigorous, rather *it should always steer towards interest*, ideally with a target goal – whatever that may be.

  It can be as simple as dedicated time – time, specifically carved out of our busy lives, that we can dedicate to something we (the researcher) find intellectually stimulating.

The ‘Good’ Researcher

Being a good research involves more than ‘merely’ positing brilliant ideas and implementing them.

Most researchers spend the majority of their time:

- reading papers
- listening to talks
- discussing ideas with others
- writing and revising papers
- staring blankly into space
- and, of course, having brilliant ideas and implementing them

Novel vs Applied

Applied Research: *I want to gain/share/discover knowledge*

- May rely on cutting edge publications
- Often involves more active work by the researcher in both *experimentation* and *investigation*, e.g.:
    - git[hub/lab] repo scraping
    - targeted web searches, specific to a [sub]domain, timeline, application, and other targeted web searches (use more than one search engine)
    - reverse engineering a closed-source technology

Novel Research: *I want to create/contribute/posit this new knowledge*

- Often rely more on cutting edge journals and publications
- Ideas, initally, tend to be more *speculative* and difficult to explain

Characteristics of the Ideal Research Topic

Before we think about research ideas,

let's discuss what the *ideal* one looks like

Characteristics of the Ideal Research Topic

The subject should be timely.

Previous groundwork should leave your research problem ripe for completion, and it should be in an active area with potential for future work and employment.

Characteristics of the Ideal Research Topic

Your work should lead to a well-defined set of results to which you can lay claim.

Results can be significantly harder to produce if you don’t have a group of people working on closely related problems with whom you can interact and share code.

Characteristics of the Ideal Research Topic

The best topics show a high level of creativity – and are often somewhat speculative.

It’s often unclear how an idea will develop, at first.

Characteristics of the Ideal Research Topic

You should really enjoy the subject, and want to spend the next several weeks/months/years with it!

An ideal project in a foreign subject is of no use without a willing person/group to help direct you.

- **The subject should be timely.**
- **Your work should lead to a well-defined set of results to which you can lay claim.**
- **The best topics show a high level of creativity – and are often somewhat speculative.**
- **You should really enjoy the subject, and want to spend the next several weeks/months/years with it!**

But...these characteristics are, to some extent, incompatible

- **The subject should be timely.**
- **Your work should lead to a well-defined set of results to which you can lay claim.**
- **The best topics show a high level of creativity – and are often somewhat speculative.**
- **You should really enjoy the subject, and want to spend the next several weeks/months/years with it!**

But...these characteristics are, to some extent, incompatible
Clearly some compromise is necessary here!

Finding an Idea

Active Readering and Listening

It's very important to make the transition **from** the **passive** mode of learning that traditional lecture courses encourage **to** an **active and critical** learning style.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related elds might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related fields might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Active Readering and Listening

Whenever you read technical material, evaluate a piece of software, or listen to a research talk, ask yourself these canonical questions:

- From where did the author seem to draw the ideas?
- What exactly was accomplished by this piece of work?
- How does it seem to relate to other work in the field?
- What would be the reasonable next step to build upon this work?
- What questions are left unanswered?
- What are the important references cited by this work?
- What ideas from related elds might be brought to bear upon this subject?
- Can the results be generalized?

Try to keep a written log of your technical reading and listening. Review it periodically to see if some of the ideas begin to fit together.

Exposing Yourself to Ideas

Set aside some time every week/month for trying to generate research ideas or improving on exisiting ideas.

- Finding and reading related work is the foundation of good research.
- If you're new to the field, you may only be familiar with buzzwords and/or material presented in textbooks. Some important initial resources for finding pertinent references are:
  - The Association for Computing Machinery (ACM) Digital Library (DL)
  - The Computing Research Association (CRA)
  - The arXiv Computing Research Repository (CoRR)
  - Github/Gitlab, Phrack, PagedOut!
  - DEF CON Media Server
  - CCC Media Server
- If there's a mailing list, subscribe to it. Trying to pwn something as part of the research? *Subscribe to the developer mailing list.*

How can these ideas help me solve my research problem?

Exposing Yourself to Ideas

Set aside some time every week/month for trying to generate research ideas or improving on exisiting ideas.

- Finding and reading related work is the foundation of good research.
- If you're new to the field, you may only be familiar with buzzwords and/or material presented in textbooks. Some important initial resources for finding pertinent references are:
  - The Association for Computing Machinery (ACM) Digital Library (DL)
  - The Computing Research Association (CRA)
  - The arXiv Computing Research Repository (CoRR)
  - Github/Gitlab, Phrack, PagedOut!
  - DEF CON Media Server
  - CCC Media Server
- If there's a mailing list, subscribe to it. Trying to pwn something as part of the research? *Subscribe to the developer mailing list.*

How can these ideas help me solve my research problem?

Exposing Yourself to Ideas

Set aside some time every week/month for trying to generate research ideas or improving on exisiting ideas.

- Finding and reading related work is the foundation of good research.
- If you're new to the field, you may only be familiar with buzzwords and/or material presented in textbooks. Some important initial resources for finding pertinent references are:
  - The Association for Computing Machinery (ACM) Digital Library (DL)
  - The Computing Research Association (CRA)
  - The arXiv Computing Research Repository (CoRR)
  - Github/Gitlab, Phrack, PagedOut!
  - DEF CON Media Server
  - CCC Media Server
- If there's a mailing list, subscribe to it. Trying to pwn something as part of the research? *Subscribe to the developer mailing list.*

How can these ideas help me solve my research problem?

Exposing Yourself to Ideas

Set aside some time every week/month for trying to generate research ideas or improving on exisiting ideas.

- Finding and reading related work is the foundation of good research.
- If you're new to the field, you may only be familiar with buzzwords and/or material presented in textbooks. Some important initial resources for finding pertinent references are:
  - The Association for Computing Machinery (ACM) Digital Library (DL)
  - The Computing Research Association (CRA)
  - The arXiv Computing Research Repository (CoRR)
  - Github/Gitlab, Phrack, PagedOut!
  - DEF CON Media Server
  - CCC Media Server
- If there's a mailing list, subscribe to it. Trying to pwn something as part of the research? *Subscribe to the developer mailing list.*

How can these ideas help me solve my research problem?

Exposing Yourself to Ideas

Set aside some time every week/month for trying to generate research ideas or improving on exisiting ideas.

- Finding and reading related work is the foundation of good research.
- If you're new to the field, you may only be familiar with buzzwords and/or material presented in textbooks. Some important initial resources for finding pertinent references are:
  - The Association for Computing Machinery (ACM) Digital Library (DL)
  - The Computing Research Association (CRA)
  - The arXiv Computing Research Repository (CoRR)
  - Github/Gitlab, Phrack, PagedOut!
  - DEF CON Media Server
  - CCC Media Server
- If there's a mailing list, subscribe to it. Trying to pwn something as part of the research? *Subscribe to the developer mailing list.*

How can these ideas help me solve my research problem?

Exposing Yourself to Ideas

Struggling with remaining active? Try one (or more) of these catalysts:

- Make a monthly investigation to read, at least, the recent abstracts from a given journal. Choose an article or two to read in-depth and critique.
- Make a weekly investigation to find technical talks/videos in your field. View selectively and critique.
- Pick a random conference (DEFCON/BH/CCC/etc.) talk or series. Listen and critique.
- Search common code repository sites for keywords relevant to the research. Learn from and/or contribute to any that you find. Critique it's approach.

Add these to your log, and ask the canonical questions.

Becoming Part of the Community

Join a community: Local IRL or online (slack/discord/zulip/etc) communities provide an informal and casual settings to learn, teach, and bounce ideas. *This is especially import for researchers new to the field.*

Attend conferences/workshops: Conferences and workshops are the best place to meet people, discuss your ideas, hear new ideas, and get a sense for the current state of research in a field.

Publish papers: Publishing papers provides a source of feedback and forces you to clarify the ideas presented and to fit them into the context of the current state of research.

Deliver talks: Talks are a great way gain visibility, share your ideas, and hear them out-loud. Submit to CFPs at the next conference; lesser-known conferences provide the ideal 'low-pressure' environment for a new speaker!

Networking: Talk about your research interests [to willing participants] every chance you get (but make sure to spend time *listening*, too – this is when you'll learn the most)

The Research Project

The Project

Research in computer science often leads to a “project” involving programming.

It’s important to remember that programming is not computer science research. Instead, for most computer scientists, programming is merely a mechanism for performing an experiement.

As with any experiment, it should be carefully planned, ahead of time...

The Project

- **Establish goals.** Know where you are headed, and approach the solution without distraction. Develop a list of milestones which demonstrate progress, and strive to accomplish them. If you cannot formulate concise goals, you should stop and reconsider the motivations for the project.

  Goals should be 'justified' regularly. As your knowledge and understanding improve, so may your ideas and motivations. Dynamic goals provide flexibility as you undergo research; *don't be afraid to scrap a goal in favour of another.*

The Project

- **Think simple.** Design your projects so that they may be completed within a reasonable period of time. An experienced programmer generates little more than a hundred lines of reliable code per month. A project that demands thousands of lines of code, therefore, will take more than a couple of two-week blocks to implement correctly. Time spent pruning the experiment to a manageable size is time well spent. Large projects do not necessarily yield large results.

The Project

- **Build prototypes.** Most projects benefit from the construction of a prototype. A well considered prototype validates assumptions, tests the value of abstractions, and motivates reconsideration of weak ideas. While there is little research value associated with polishing off a ‘product’, many research questions can be answered satisfactorily through mock-ups or partial implementations.

The Project

- **Use tools.** A programmer's performance is dramatically improved through the use of a few simple tools. There are, of course, many important and useful tools, but the main point is clear: *the correct choice of tools can reduce the total work in a project.* Find them, learn from them, and use them.

The Project

- **Collaborate.** When resources can be coordinated, groups are often more productive than individual efforts in isolation. For many, success comes from collaboration. Try to share and develop ideas in a group atmosphere. Contact others that share your interestes and collaborate! Undoubtedly they will have solved problems you are currently considering, and their solutions will influence how you achieve common goals.

  One side effect of collaboration is increased discipline, discipline that is necessary to reduce the amount of energy expended synchronizing efforts. Some tips for collaboration:

  - Keep a regular schedule of meetings
  - Establish points of synchronization where concentration is refocused on an issue
  - Consider critisim carefully; conversely, provide others only with constructive critisim

The Project

- **Document results.** Finished projects should be documented. At a minimum, a technical overview of the experiment allows others to see what motivates your research. The document should describe the problem, your assumptions, your approach, and an honest evaluation of your results. When documenting software, include illustrative examples, tutorials, and any experience gained from its use. Well written documentation greatly increases the impact of a project.

  It's vital to reserve a good portion of time for writing. An hour spent writing is an hour spent considering a problem instead of, say, grappling with a computer. You must spend time away from other distractions to document work and focus your efforts.

Questions?