Multi-task Deep Learning in the Software Development domain

Silvia Severini, Garching, 27.05.19
Advisor: Ahmed Elnaggar

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

- Motivation
- Introduction
- Research questions
- Methodology
- Tasks
- Model architecture overview
- Timeline of the thesis
- References

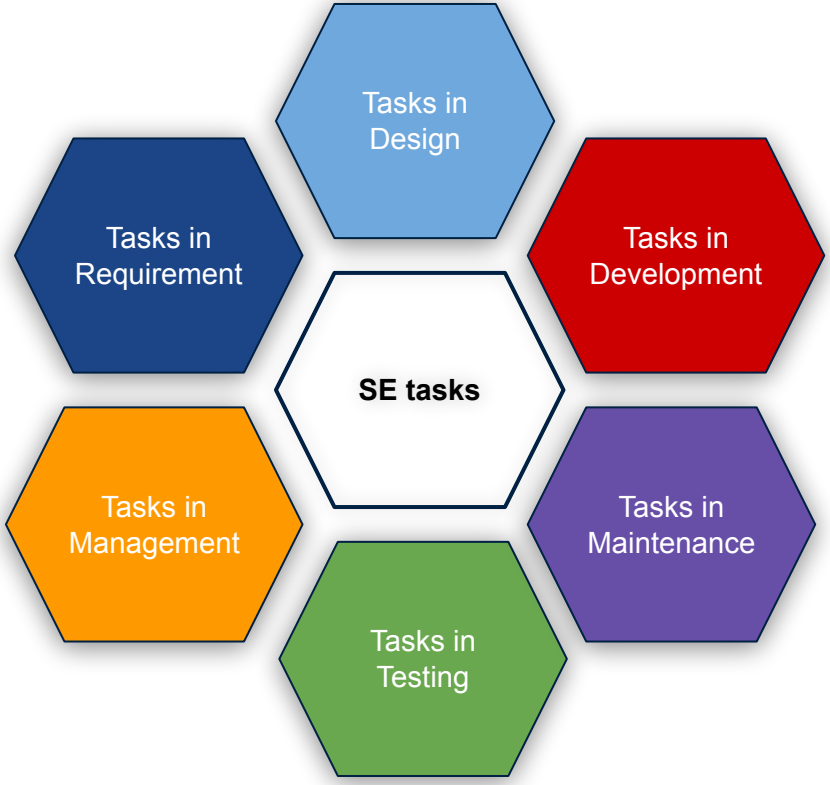
Outline

- **Motivation**
- Introduction
- Research questions
- Methodology
- Tasks
- Model architecture overview
- Timeline of the thesis
- References

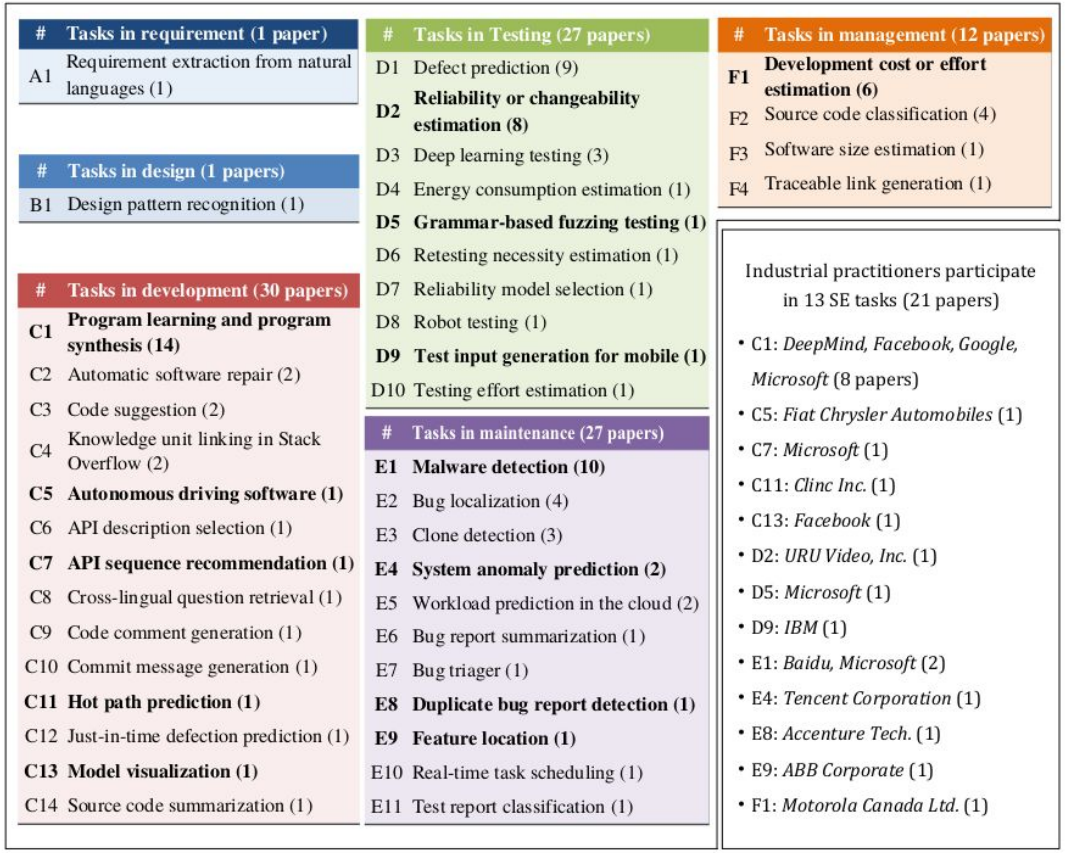
A recent growing interest

- A variety of tasks in the software development domain can benefit from the aid of Machine Learning and Deep Learning
- Deep learning has achieved competitive performance against previous algorithms on about 40 SE tasks
- Industrial practitioners are also interested in integrating Deep Learning into their SE solutions

Tasks in the Software development domain



Tasks in the Software development domain



#	Tasks in requirement (1 paper)
A1	Requirement extraction from natural languages (1)

#	Tasks in design (1 papers)
B1	Design pattern recognition (1)

#	Tasks in development (30 papers)
C1	**Program learning and program synthesis (14)**
C2	Automatic software repair (2)
C3	Code suggestion (2)
C4	Knowledge unit linking in Stack Overflow (2)
C5	**Autonomous driving software (1)**
C6	API description selection (1)
C7	**API sequence recommendation (1)**
C8	Cross-lingual question retrieval (1)
C9	Code comment generation (1)
C10	Commit message generation (1)
C11	**Hot path prediction (1)**
C12	Just-in-time defection prediction (1)
C13	**Model visualization (1)**
C14	Source code summarization (1)

#	Tasks in Testing (27 papers)
D1	Defect prediction (9)
D2	**Reliability or changeability estimation (8)**
D3	Deep learning testing (3)
D4	Energy consumption estimation (1)
D5	**Grammar-based fuzzing testing (1)**
D6	Retesting necessity estimation (1)
D7	Reliability model selection (1)
D8	Robot testing (1)
D9	**Test input generation for mobile (1)**
D10	Testing effort estimation (1)

#	Tasks in maintenance (27 papers)
E1	**Malware detection (10)**
E2	Bug localization (4)
E3	Clone detection (3)
E4	**System anomaly prediction (2)**
E5	Workload prediction in the cloud (2)
E6	Bug report summarization (1)
E7	Bug triager (1)
E8	**Duplicate bug report detection (1)**
E9	**Feature location (1)**
E10	Real-time task scheduling (1)
E11	Test report classification (1)

#	Tasks in management (12 papers)
F1	**Development cost or effort estimation (6)**
F2	Source code classification (4)
F3	Software size estimation (1)
F4	Traceable link generation (1)

Industrial practitioners participate in 13 SE tasks (21 papers)

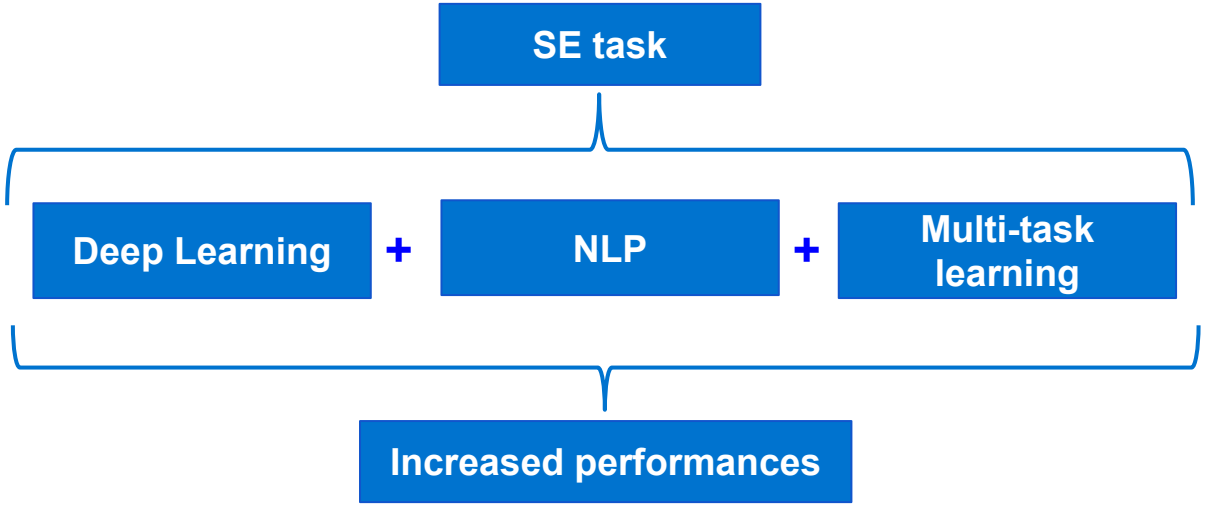
- C1: *DeepMind, Facebook, Google, Microsoft* (8 papers)
- C5: *Fiat Chrysler Automobiles* (1)
- C7: *Microsoft* (1)
- C11: *Clinc Inc.* (1)
- C13: *Facebook* (1)
- D2: *URU Video, Inc.* (1)
- D5: *Microsoft* (1)
- D9: *IBM* (1)
- E1: *Baidu, Microsoft* (2)
- E4: *Tencent Corporation* (1)
- E8: *Accenture Tech.* (1)
- E9: *ABB Corporate* (1)
- F1: *Motorola Canada Ltd.* (1)

Fig. 4. The SE tasks solved by deep learning and participated by industrial practitioners [3]

Outline

- Motivation
- **Introduction**
- Research questions
- Methodology
- Tasks
- Model architecture overview
- Timeline of the thesis
- References

Artificial Intelligence applied to SE



Natural Language Processing for Source Code

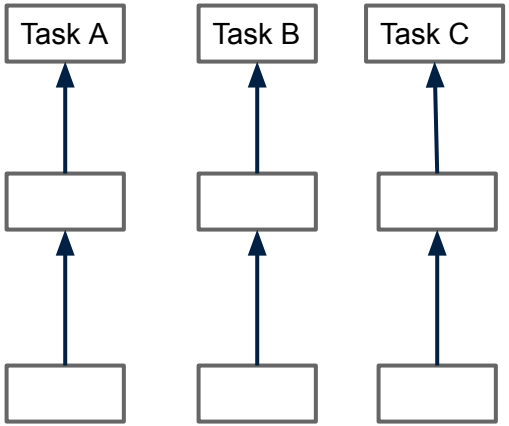
Programming language as a new language like English

- Complexity
- Context awareness
- Unlimited vocabulary
- Dataset scarcity (required GitHub scraper [5])
- Tokenization of each programming language

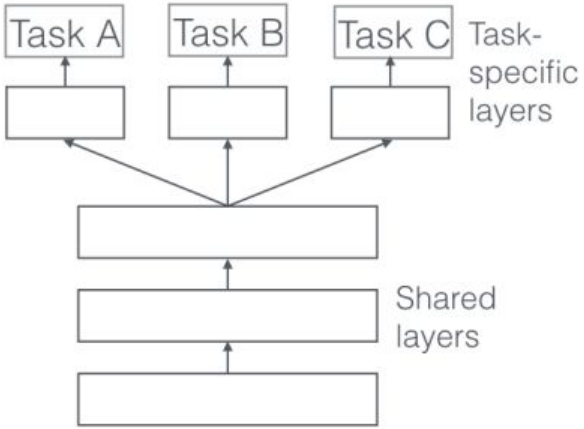

```
public class HelloWorld {

    public static void main(String[] args) {
        // Prints "Hello, World" to the terminal window.
        System.out.println("Hello, World");
    }

}
```

public class HelloWorld { public static void main (String [ ] args) { System . out . println (" Hello, World ") ; } }

Single-task vs Multi-task learning



Single-task learning



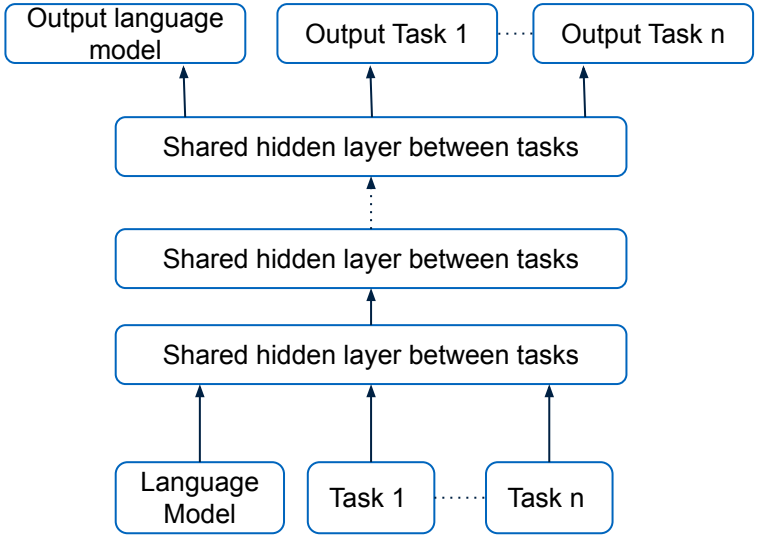
Hard parameter sharing for multi-task learning in deep neural networks [1]

Why Multi-task learning ?

"Given m learning tasks $\{T_i\}_{i=1}^{m}$ where all the tasks or a subset of them are related, multi-task learning aims to help improve the learning of a model for T_i by using the knowledge contained in all or some of the m tasks."

- Implicit data augmentation
- Regularization
- Attention focusing
- Representation bias

=> Augment of the generalization capabilities



Outline

- Motivation
- Introduction
- **Research questions**
- Methodology
- Tasks
- Model architecture overview
- Timeline of the thesis
- References

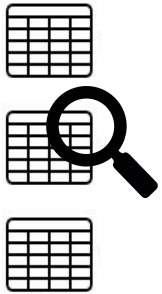
Research questions

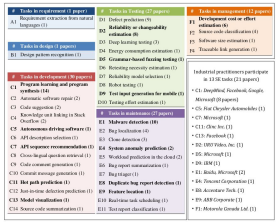
1	Can multi-task deep learning be beneficial for tasks in the software development domain?
2	How far is multi-task deep learning from state-of-the-art solutions in the software development domain?
3	Could the model be trained with English language and programming language together?
4	How does training on multiple tasks of the software development domain simultaneously compare to training on each task separately?
5	Which tasks could be combined together in order to achieve better performances?

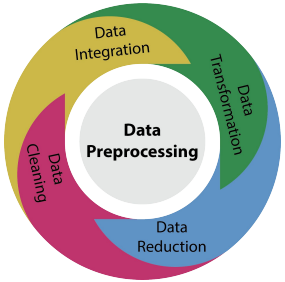
Outline

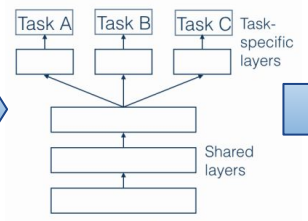
- Motivation
- Introduction
- Research questions
- **Methodology**
- Tasks
- Model architecture overview
- Timeline of the thesis
- References

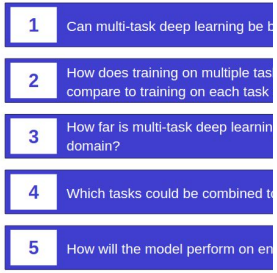
Methodology: overview











Search for available datasets

Choose relevant tasks

Preprocessing and integration of the datasets

Train deep learning models

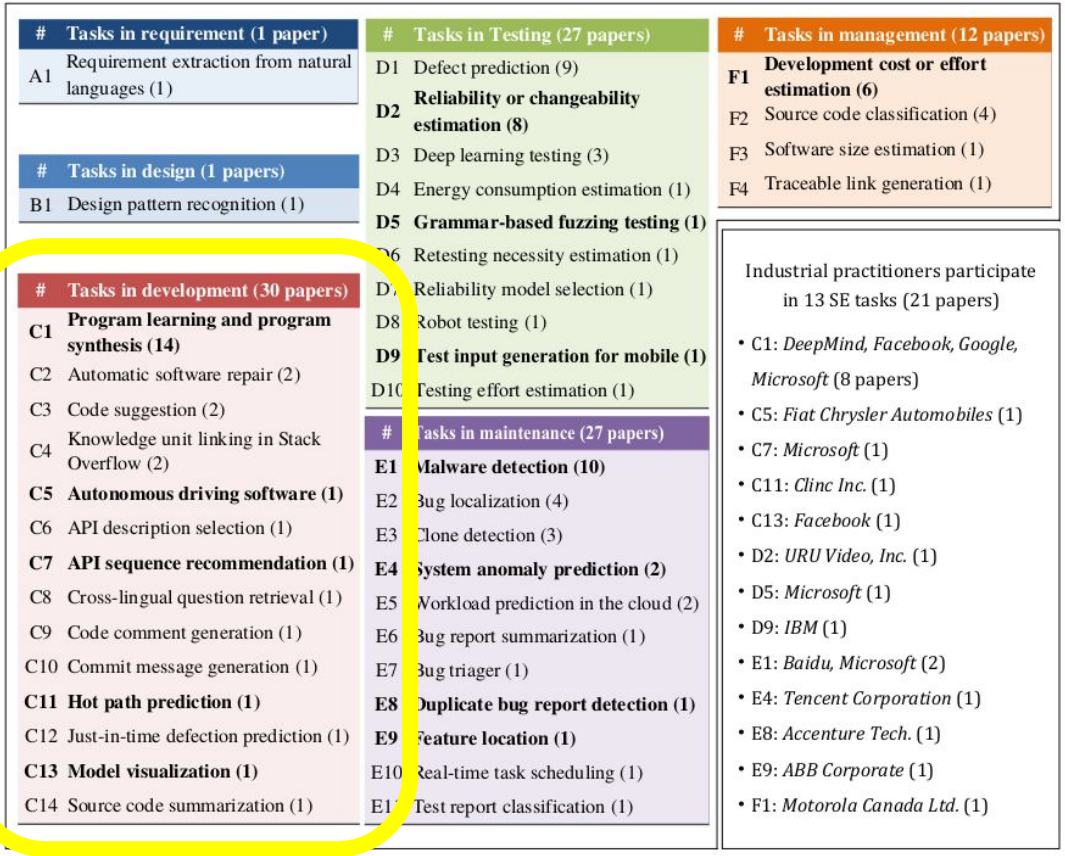
Evaluation of the results

Verification and validation of the research questions

Outline

- Motivation
- Introduction
- Research questions
- Methodology
- **Tasks**
- Model architecture overview
- Timeline of the thesis
- References

Tasks in the Software development domain



#	Tasks in requirement (1 paper)
A1	Requirement extraction from natural languages (1)

#	Tasks in design (1 papers)
B1	Design pattern recognition (1)

#	Tasks in development (30 papers)
C1	**Program learning and program synthesis (14)**
C2	Automatic software repair (2)
C3	Code suggestion (2)
C4	Knowledge unit linking in Stack Overflow (2)
C5	**Autonomous driving software (1)**
C6	API description selection (1)
C7	**API sequence recommendation (1)**
C8	Cross-lingual question retrieval (1)
C9	Code comment generation (1)
C10	Commit message generation (1)
C11	**Hot path prediction (1)**
C12	Just-in-time defection prediction (1)
C13	**Model visualization (1)**
C14	Source code summarization (1)

#	Tasks in Testing (27 papers)
D1	Defect prediction (9)
D2	**Reliability or changeability estimation (8)**
D3	Deep learning testing (3)
D4	Energy consumption estimation (1)
D5	**Grammar-based fuzzing testing (1)**
D6	Retesting necessity estimation (1)
D7	Reliability model selection (1)
D8	Robot testing (1)
D9	**Test input generation for mobile (1)**
D10	Testing effort estimation (1)

#	Tasks in maintenance (27 papers)
E1	**Malware detection (10)**
E2	Bug localization (4)
E3	Clone detection (3)
E4	**System anomaly prediction (2)**
E5	Workload prediction in the cloud (2)
E6	Bug report summarization (1)
E7	Bug triager (1)
E8	**Duplicate bug report detection (1)**
E9	**Feature location (1)**
E10	Real-time task scheduling (1)
E11	Test report classification (1)

#	Tasks in management (12 papers)
F1	**Development cost or effort estimation (6)**
F2	Source code classification (4)
F3	Software size estimation (1)
F4	Traceable link generation (1)

Industrial practitioners participate in 13 SE tasks (21 papers)

- C1: *DeepMind, Facebook, Google, Microsoft* (8 papers)
- C5: *Fiat Chrysler Automobiles* (1)
- C7: *Microsoft* (1)
- C11: *Clinc Inc.* (1)
- C13: *Facebook* (1)
- D2: *URU Video, Inc.* (1)
- D5: *Microsoft* (1)
- D9: *IBM* (1)
- E1: *Baidu, Microsoft* (2)
- E4: *Tencent Corporation* (1)
- E8: *Accenture Tech.* (1)
- E9: *ABB Corporate* (1)
- F1: *Motorola Canada Ltd.* (1)

Fig. 4. The SE tasks solved by deep learning and participated by industrial practitioners [3]

Tasks

Tasks	Description	Number of samples
Program learning and synthesis	Generate programs from natural language descriptions	100.000
API sequence recommendation	Generates relevant API usage sequences given a natural language query	7.500.000
Code comment generation	Automatic generation of code comments	400.000
Commit message generation	Automatically “translate” diffs into commit messages	30.000
Source code summarization	Summarization of source code snippets	80.000

Unsupervised Language model	English: 1 Billion world corpus [4]	300.000.000
	Java from PGA [5]	500.000
	SQL corpus [6]	135.000
	150K Python Dataset [7]	150.000
	C# from PGA [5]	500.000

Example of input-output pairs

Source code summarization:

- from pygithub3 import Github\n\nusername = raw_input("Please enter a Github username: ")\npassword = raw_input("Please enter the account password: ")\n\ngh = Github(login=username, password = password)\n\nget_user = gh.users.get()\n\nuser_repos = gh.repos.list().all()\n\nfor repo in user_repos:\n print repo.language
- Getting repository information using pygithub3 for Python

Code comment generation:

- public void handleEntryExpiredSA(EntryExpiredBusPacket packet) throws Exception {\n handleEntryExpiredCoreSA(packet.getEntryHolder(),packet.getTransaction(),packet.isFromReplication());\n}\n"
- Handles EntryExpired packets.

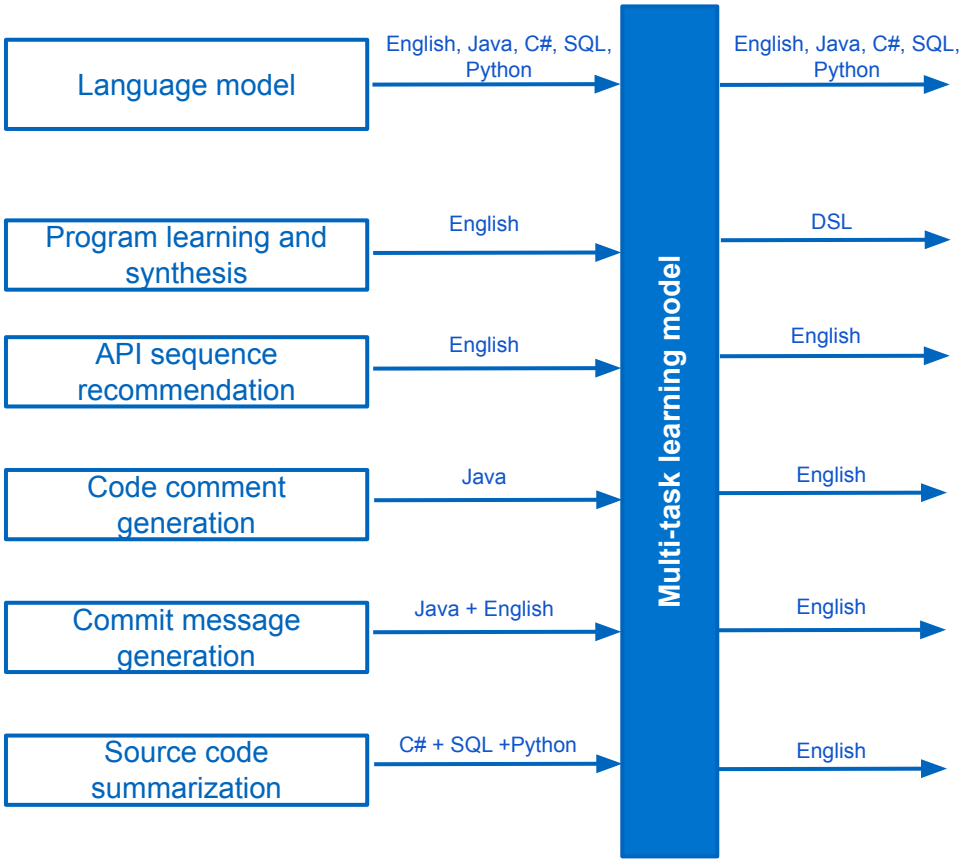
API sequence recommendation:

- return a printable representation of this exception for debugging purposes
- StringBuffer . <init> StringBuffer . append StringBuffer . toString

Outline

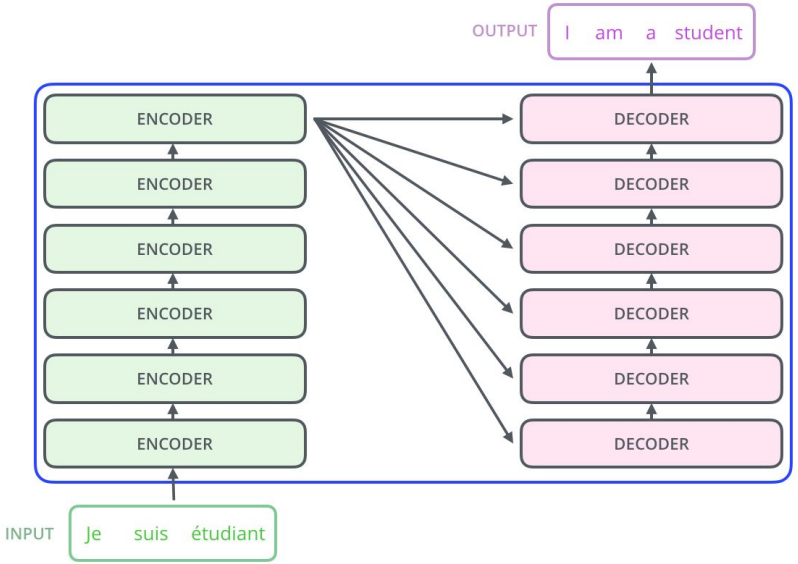
- Motivation
- Introduction
- Research questions
- Methodology
- Tasks
- **Model architecture overview**
- Timeline of the thesis
- References

Model architecture overview

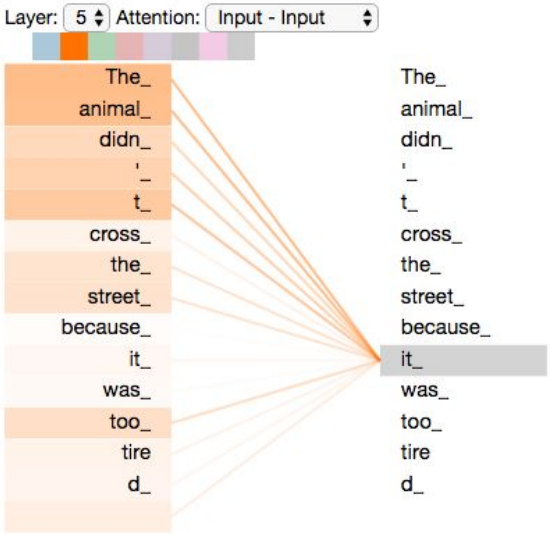


Transformer model [8] [9]

Encoder - Decoder model



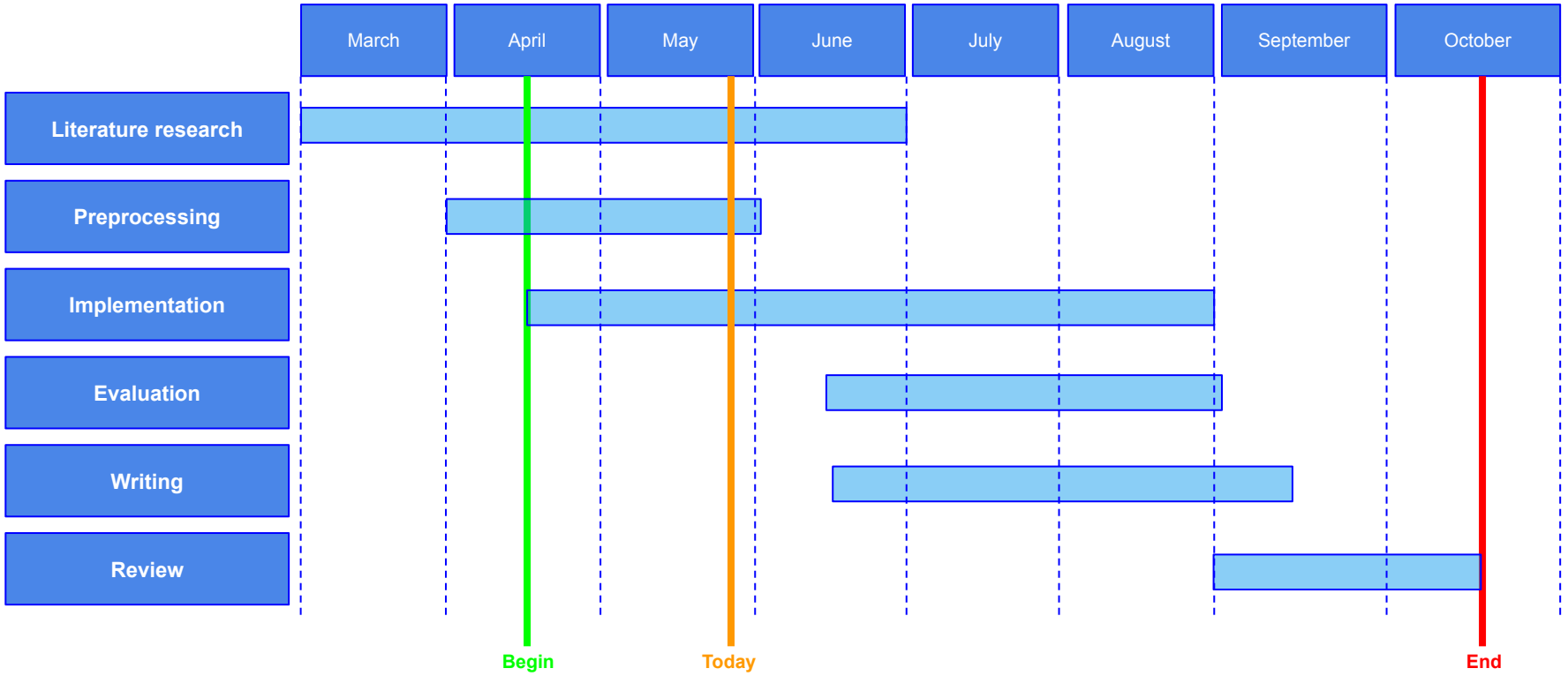
Attention mechanism



Outline

- Motivation
- Introduction
- Research questions
- Methodology
- Tasks
- Model architecture overview
- **Timeline of the thesis**
- References

Timeline of the thesis



Outline

- Motivation
- Introduction
- Research questions
- Methodology
- Tasks
- Model architecture overview
- Timeline of the thesis
- **References**

References

[1] Ruder, Sebastian. "An overview of multi-task learning in deep neural networks." arXiv preprint arXiv:1706.05098 (2017)
[2] Zhang, Yu, and Qiang Yang. "A survey on multi-task learning." arXiv preprint arXiv:1707.08114 (2017).
[3] Li, Xiaochen, et al. "Deep Learning in Software Engineering." arXiv preprint arXiv:1805.04825 (2018).
[4] http://www.statmt.org/lm-benchmark/
[5] https://github.com/src-d/datasets/tree/master/PublicGitArchive
[6]https://github.com/LittleYUYU/StackOverflow-Question-Code-Dataset/blob/master/annotation_tool/data/code_solution_labeled_data/source/sql_how_to_do_it_by_classifier_multiple_iid_to_code.pickle
[7] https://www.sri.inf.ethz.ch/py150
[8] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
[9] http://jalammar.github.io/illustrated-transformer/

Tasks related papers:

- Polosukhin, Illia, and Alexander Skidanov. "Neural program search: Solving programming tasks from description and examples." *arXiv preprint arXiv:1802.04335* (2018).
- Gu, Xiaodong, et al. "Deep API learning." *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, 2016.
- Hu, Xing, et al. "Deep code comment generation." *Proceedings of the 26th Conference on Program Comprehension*. ACM, 2018.
- Iyer, Srinivasan, et al. "Summarizing source code using a neural attention model." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2016.
- Jiang, Siyuan, Ameer Armaly, and Collin McMillan. "Automatically generating commit messages from diffs using neural machine translation." Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering. IEEE Press, 2017.

Silvia Severini

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for
Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289. 17132
Fax +49.89.289.17136

matthes@in.tum.de
wwwmatthes.in.tum.de