Exploring the Possibilities of Applying Transfer Learning Methods for Natural Language Processing in Software Development

Wei Ding, 22.02.2021, Master Thesis

Advisor: Ahmed Elnaggar

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

Outline

Outline

- **Motivation and Introduction**
- Research Questions
- Model Architecture
- Tasks and Datasets
- Evaluation Metrics
- Results and Discussion
- Conclusion and Future Work
- References

Introduction and Motivation

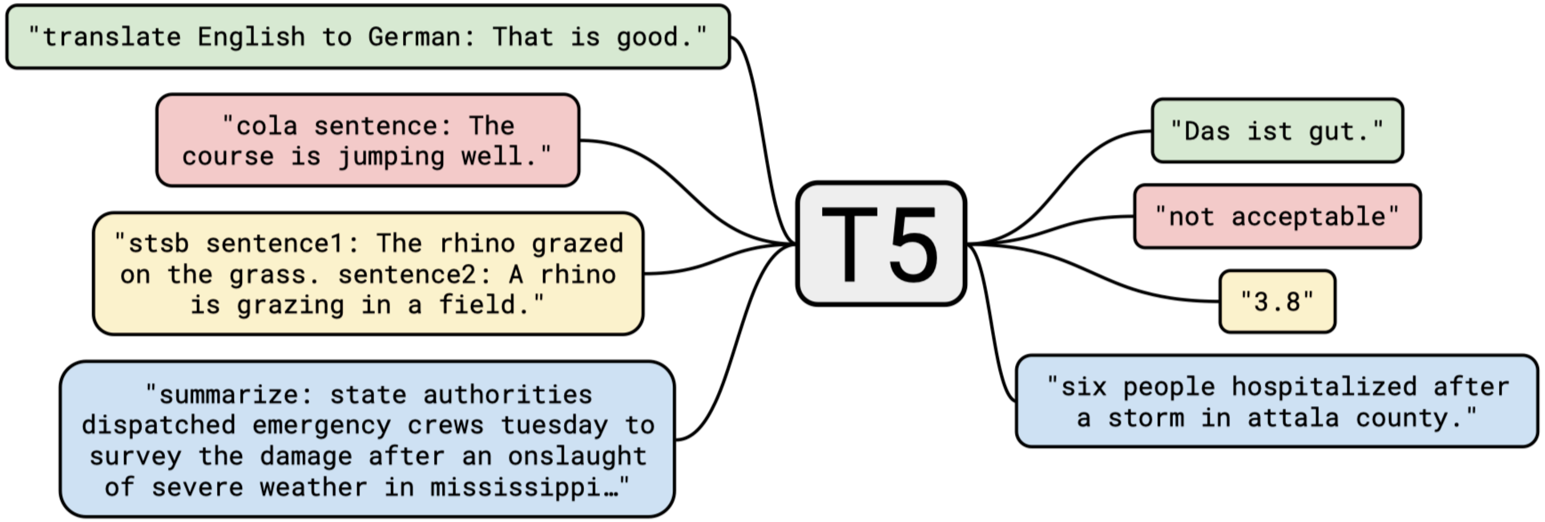
- Software Development in NLP
  - Software code is a language

  - NLU and NLG applications:
    - Analytics Dashboards
    - Chatbot
    - Content Creation

  - Visions of NLP in the Software Development world
    - Better readability of the code
    - Easier to compare and evaluate the code
    - Smoother developing process

Introduction and Motivation

- Recent success of Transfer Learning
  - Pre-Training Large Dataset → Fine-Tuning Small Dataset
  - Text-To-Text Transfer Transformer Google
  - Advantages of T5
    - same model, loss function, and hyperparameters on *any* NLP task



Outline

- Motivation and Introduction
- **Research Questions**
- Model Architecture
- Tasks and Datasets
- Evaluation Metrics
- Results and Discussion
- Conclusion and Future Work
- References

Research Questions

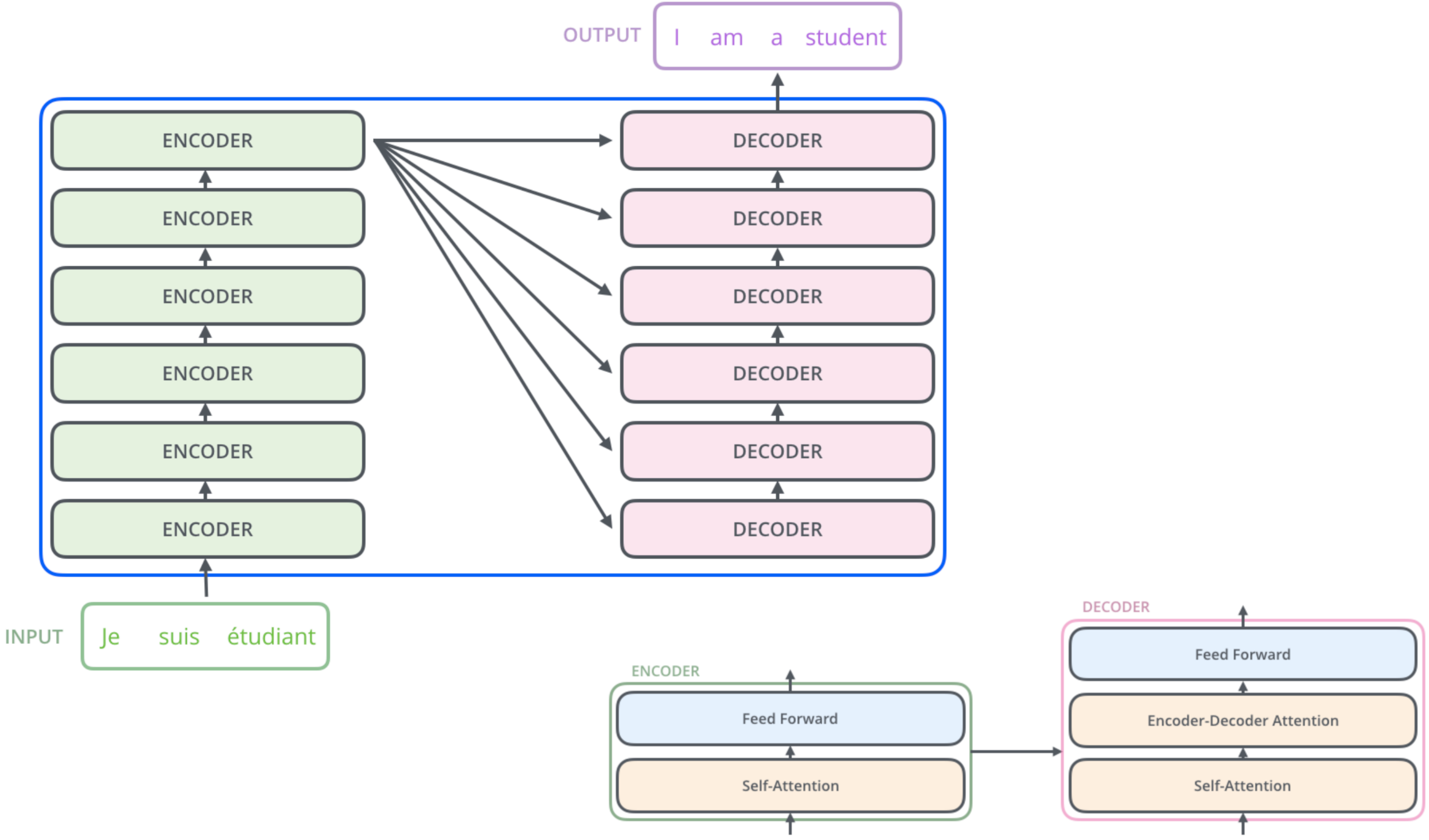
What kind of natural language processing models would work best for tasks in the software development domain?

How would transfer learning improve the performance comparing with only training on the labeled data alone?

Would transfer learning perform better than multi-task learning for the similar tasks?

Outline

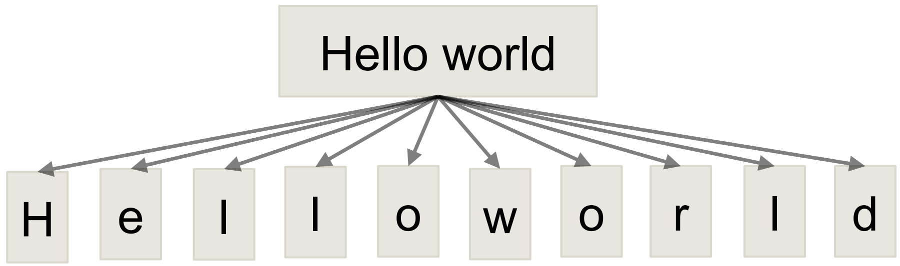
Model Architecture



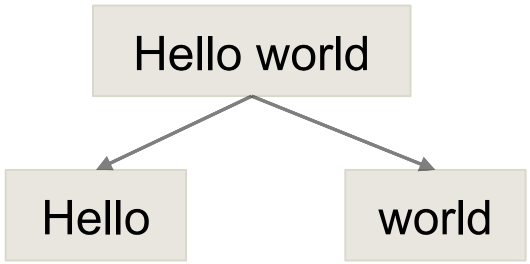
Model Architecture

- Vocabulary Tokenization

1. Character-Level



2. Word-Level



3. Subword-Level

32,000 Tokens, including: “function”, “String”, “var”, “import”
……

Model Architecture

- Unsupervised objective

Input	Output
Thank you for inviting	me to your party last week .
Thank you <M> <M> me to your party <M> week .	Thank you for inviting me to your party last week .
Thank you <M> <M> me to your party apple week .	Thank you for inviting me to your party last week .
Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>

Outline

- Motivation and Introduction
- Research Questions
- Model Architecture
- **Tasks and Datasets**
- Evaluation Metrics
- Results and Discussion
- Conclusion and Future Work
- References

Tasks and Datasets

- Fine-tune tasks
  1. Code documentation generation
     - Code Language: Python, Java, Go, Php, Ruby, Javascript
     - Code Source: Github
     - Data Example:

Input: Code Function/Method

```
def parse_query_param(url, param):
    try:
        return parse.parse_qs(parse.urlparse(url)
                  .query)[param][0]
    except:
        return None
```

Target: Natural Language Documentation Text

Parses the query string of a URL and returns the value of a parameter.

Tasks and Datasets

- Fine-tune tasks

  2. Source summarization
     - Code Language: SQL, CSharp, Python
     - Code Source: StackOverflow
     - Data Example:

Input: Code snippet	Target: Questions from Stackoverflow
select count (col0) , col1 from col2 group by col1 order by CODEINTEGER desc limit CODEINTEGER;	Sql group by with an order by

Tasks and Datasets

- Fine-tune tasks

  3. Code comment generation
     - Code Language: Java
     - Code Source: Github
     - Data Example:

Input: Java Function

```
public boolean isCritical(){
    return false;
}
```

→

Target: Comment from the function

is this a critical command that can only be executed when no other command is running?

Tasks and Datasets

- Fine-tune tasks

  4. Commit message generation
     - Code Language: Java
     - Code Source: Github
     - Data Example:

Input: Diffs from Github	Target: Commit Message
mmm a / CHANGELOG . md ppp b / CHANGELOG . md # Changelog - # 2 . 2 . 0 (16 / 07 / 2015) - SNAPSHOT + # 2 . 1 . 1 (29 / 02 / 2016) - SNAPSHOT - Added AppCompat Styles (AppCompatTextView will now pickup textViewStyle etc) . Thanks @ paul - turner - Fix for Toolbar not inflating ` TextView ` s upfront .	Fix snapshot version

Tasks and Datasets

- Fine-tune tasks

  5. Api sequence recommendation
     - Code Language: Java
     - Code Source: Github
     - Data Example:

Input: Task description in natural language

Convert from normal rgb to java hsb

→

Target: API sequence

Color.RGBtoHSB Color.getHSBColor

Tasks and Datasets

- Fine-tune tasks

  6. Programming Language and Synthesis:

     - Code Language: LISP
     - Code Source: Computer Science Student Homework
     - Data Example:

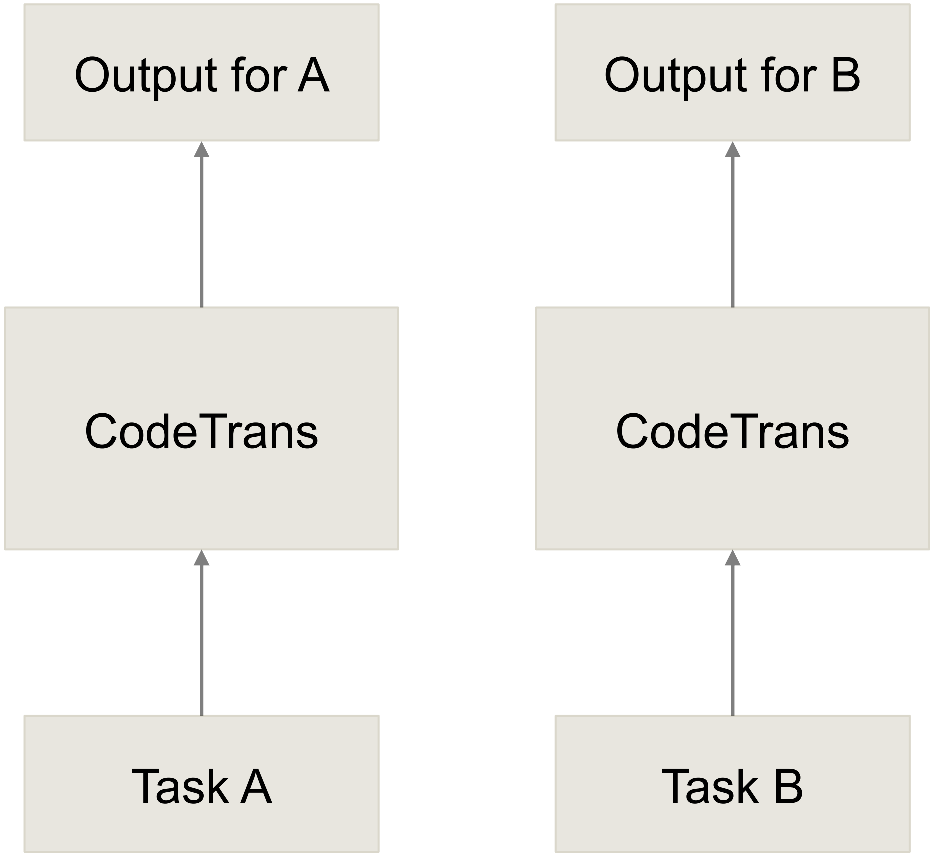
Input: Task description in natural language		Target: Programming Language Synthesis
consider an array of numbers a , compute elements of a that are odd	→	[filter a [lambda1 [== [% arg1 2] 1]]]

Tasks and Datasets

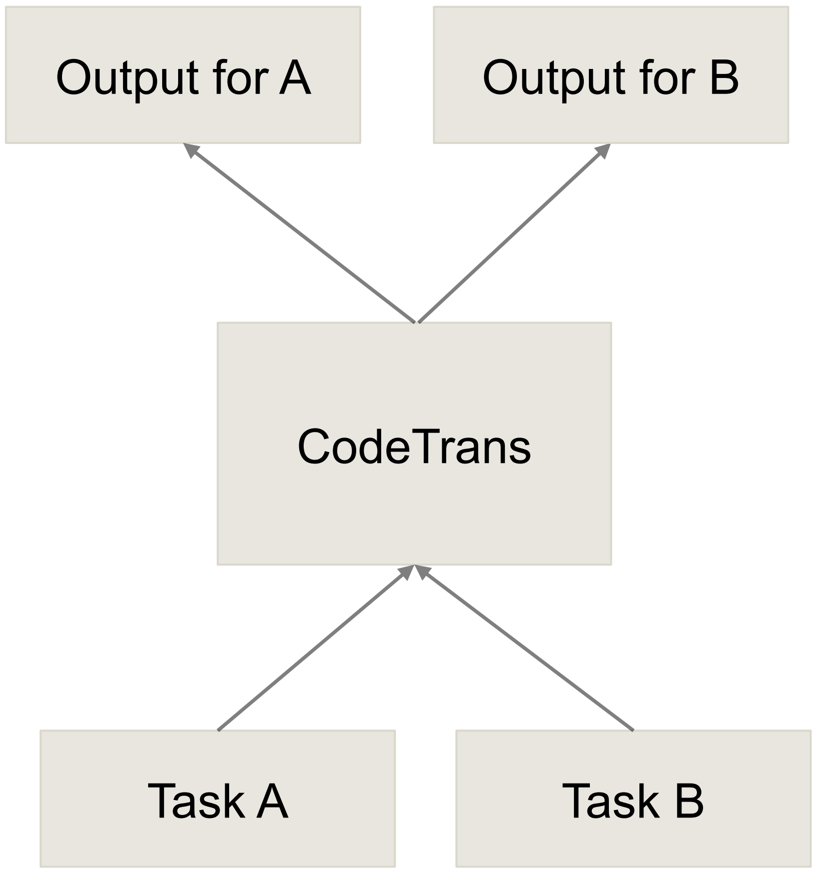
Language	Code Documentation Generation		Source Summarization		Code Comment Generation		Commit Message Generation		API Sequence Recommendation		Programming Language and Synthesis		Unlabeled
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train
Java	164,923	10,955			468,000	58,638	26,208	3,000	7,475,850	10,000			2,163,807
Python	251,820	14,918	12,004	2,783									1,181,354
JavaScript	58,025	3,291											1,817,579
Go	167,288	8,122											679,985
Ruby	24,927	1,261											154,354
Php	241,241	14,014											767,981
CSharp			52,943	6,629									469,038
SQL			25,671	3,340									133,191
LISP											79,214	9,967	122,602
English													30,913,716

Training Strategies

Single Task Learning

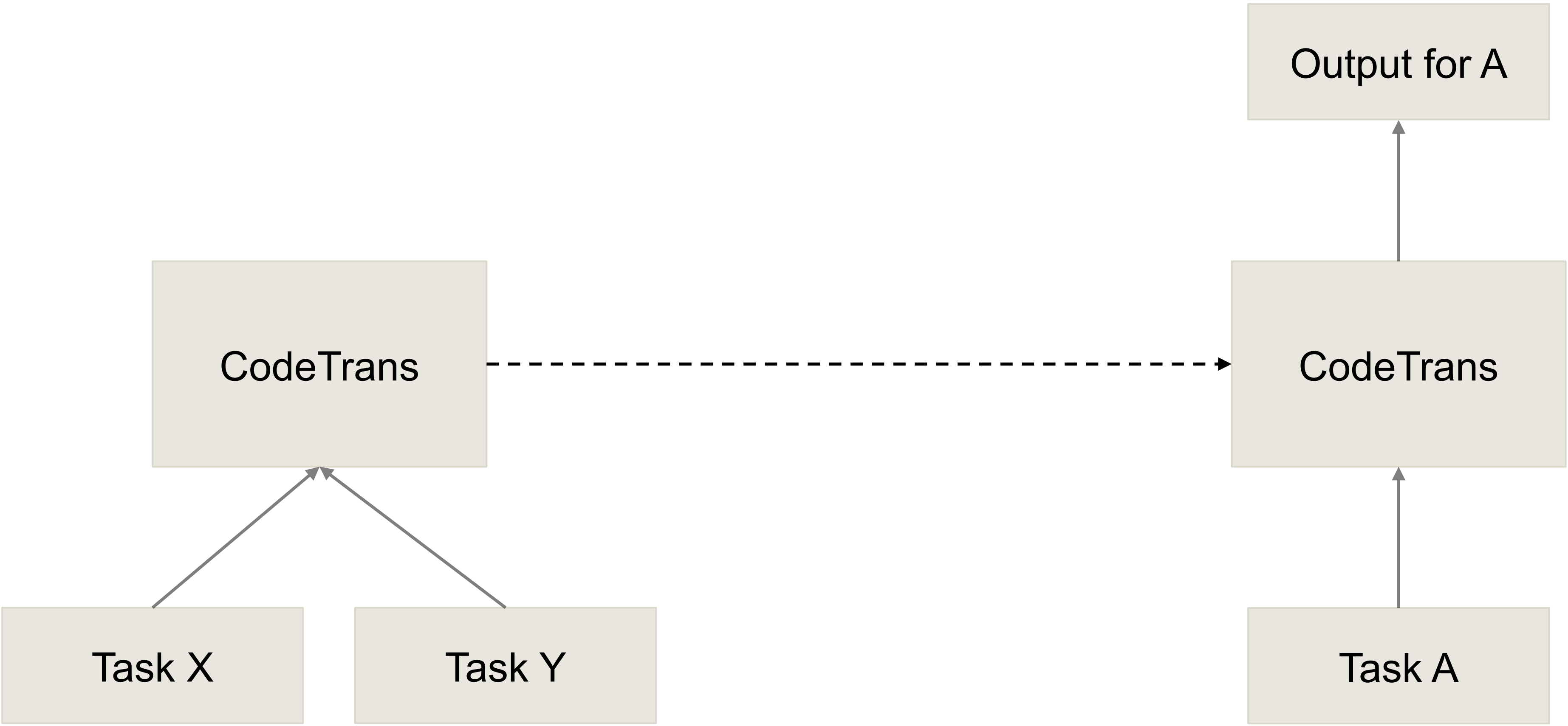


Multi-Task Learning



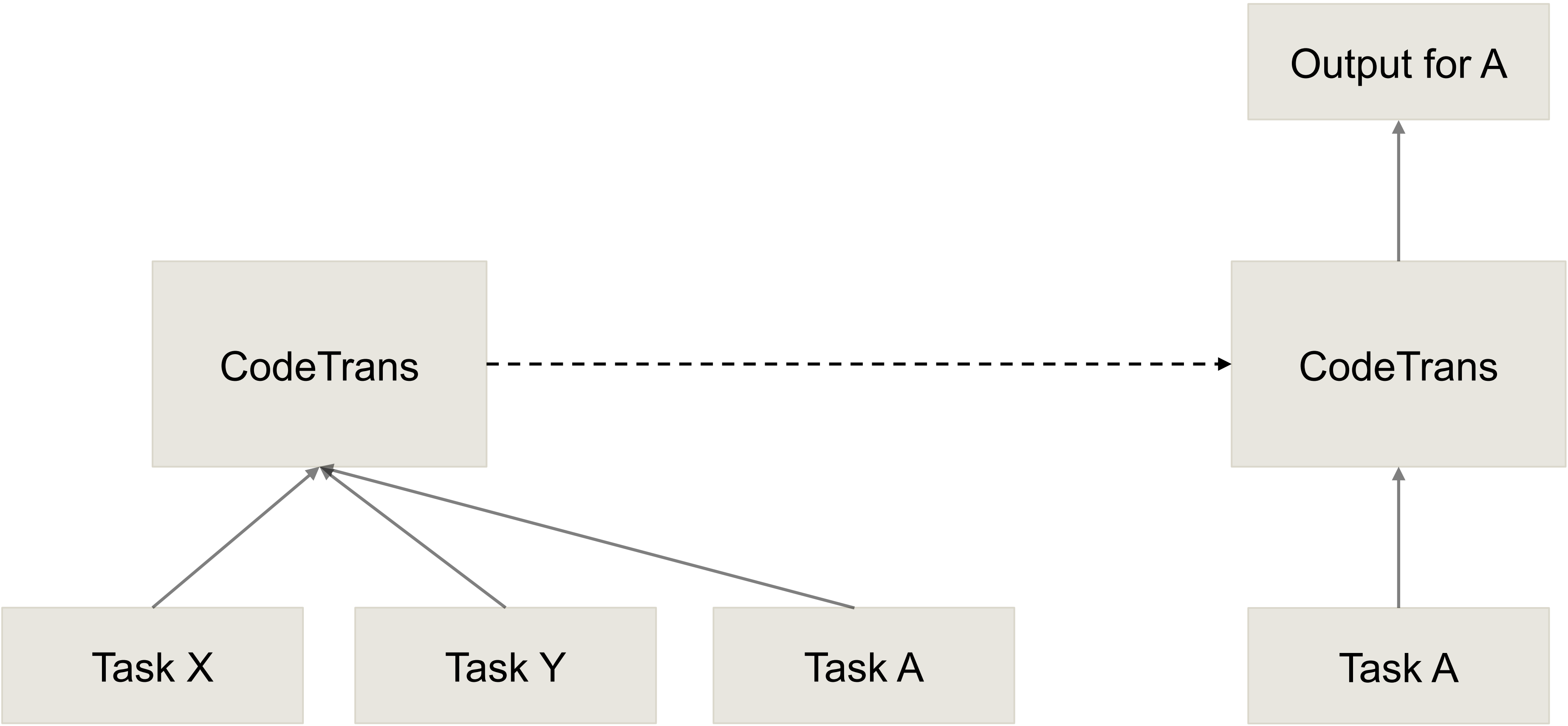
Training Strategies

Transfer Learning



Training Strategies

Multi-task Learning Fine-tuning



Outline

- Motivation and Introduction
- Research Questions
- Model Architecture
- Tasks and Datasets
- **Evaluation Metrics**
- Results and Discussion
- Conclusion and Future Work
- References

Evaluation Metrics

- BLEU
  - Formula:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Modified Precision:

$$p_n = \frac{\text{Word maximal occurrence of output in reference}}{\text{Word occurrence in output}}$$

$$w_n = 1/N$$

N: n-gram

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

c: length of the machine output
r: length of the reference sentence

Evaluation Metrics

- Bigrams example for calculating modified precision in BLEU:

Machine generated output: the cat the cat on the mat.
Human reference 1: the cat is on the mat.
Human reference 2: there is a cat on the mat.

Word	Occurrence in machine output	Occurrence in reference 1	Occurrence in reference 2	Maximal occurrence in reference
the cat				
cat the				
cat on				
on the				
the mat				
Total				

Evaluation Metrics

- Bigrams example for calculating modified precision in BLEU:

Machine generated output: the cat the cat on the mat.
Human reference 1: the cat is on the mat.
Human reference 2: there is a cat on the mat.

Word	Occurrence in machine output	Occurrence in reference 1	Occurrence in reference 2	Maximal occurrence in reference
the cat	2	1	0	1
cat the				
cat on				
on the				
the mat				
Total				

Evaluation Metrics

- Bigrams example for calculating modified precision in BLEU:

Machine generated output:	the cat the cat on the mat.
Human reference 1:	the cat is on the mat.
Human reference 2:	there is a cat on the mat.

Word	Occurrence in machine output	Occurrence in reference 1	Occurrence in reference 2	Maximal occurrence in reference
the cat	2	1	0	1
cat the	1	0	0	0
cat on	1	0	1	1
on the	1	1	1	1
the mat	1	1	1	1
Total	6			4

- Modified precision is 4/6

Evaluation Metrics

- BLEU

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

- Accuracy

$$Accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

Outline

- Motivation and Introduction
- Research Questions
- Model Architecture
- Tasks and Datasets
- Evaluation Metrics
- **Results and Discussion**
- Conclusion and Future Work
- References

Results

Code Documentation Generation								
Model \ Programming Language			Python	Java	Go	Php	Ruby	Javascript
CodeTrans	Single Task Learning	Small	17.31	16.65	16.89	23.05	9.19	13.70
		Base	16.86	17.17	17.16	22.98	8.23	13.17
	Transfer Learning	Small	19.93	19.48	18.88	25.35	13.15	17.23
		Base	20.26	20.19	19.50	25.84	14.07	18.25
		Large	20.35	20.06	**19.54**	26.18	14.94	**18.98**
	Multi-task Learning	Small	19.64	19.00	19.15	24.68	14.91	15.26
		Base	**20.39**	21.22	19.43	**26.23**	**15.26**	16.11
		Large	20.18	**21.87**	19.38	26.08	15.00	16.23
	Multi-task Learning Fine-tuning	Small	19.77	20.04	19.36	25.55	13.70	17.24
		Base	19.77	21.12	18.86	25.79	14.24	18.62
		Large	18.94	21.42	18.77	26.20	14.19	18.83
CodeBert			19.06	17.65	18.07	25.16	12.16	14.90

(Metrics: Smoothed BLEU-4)

Results

Source Code Summarization					
Model \ Programming Language			Python	SQL	CSharp
CodeTrans	Single Task Learning	Small	8.45	17.55	19.74
		Base	9.12	15.00	18.65
	Transfer Learning	Small	10.06	17.71	20.40
		Base	10.94	17.66	21.12
		Large	12.41	18.40	21.43
	Multi-task Learning	Small	13.11	19.15	22.39
		Base	**13.37**	19.24	**23.20**
		Large	13.24	19.49	23.57
	Multi-task Learning Fine-tuning	Small	12.10	18.25	22.03
		Base	10.64	16.91	21.40
		Large	12.14	**19.98**	21.10
Code-NN			-	18.40	20.50

(Metrics: Smoothed BLEU-4)

Results

Code Comment Generation			
Model \ Programming Language			Java
CodeTrans	Single Task Learning	Small	37.98
		Base	38.07
	Transfer Learning	Small	38.56
		Base	39.06
		Large	**39.50**
	Multi-task Learning	Small	20.15
		Base	27.44
		Large	34.69
	Multi-task Learning Fine-tuning	Small	38.37
		Base	38.90
		Large	39.25
DeepCom			38.17

(Metrics: Smoothed BLEU-4)

Results

Git Commit Message Generation			
Model \ Programming Language			Java
CodeTrans	Single Task Learning	Small	39.61
		Base	38.67
	Transfer Learning	Small	44.22
		Base	44.17
		Large	**44.41**
	Multi-task Learning	Small	36.17
		Base	39.25
		Large	41.18
	Multi-task Learning Fine-tuning	Small	43.96
		Base	44.19
		Large	44.34
NMT			32.81

(Metrics: BLEU-4)

Results

API Sequence Recommendation			
Model \ Programming Language			Java
CodeTrans	Single Task Learning	Small	68.71
		Base	70.45
	Transfer Learning	Small	68.90
		Base	72.11
		Large	73.26
	Multi-task Learning	Small	58.43
		Base	67.97
		Large	72.29
	Multi-task Learning Fine-tuning	Small	69.29
		Base	72.89
		Large	**73.39**
DeepAPI			54.42

(Metrics: Smoothed BLEU-4)

Results

Program Synthesis			
Model \ Programming Language			DSL
CodeTrans	Single Task Learning	Small	89.43
		Base	89.65
	Transfer Learning	Small	90.30
		Base	90.24
		Large	90.21
	Multi-task Learning	Small	82.88
		Base	86.99
		Large	90.27
	Multi-task Learning Fine-tuning	Small	**90.31**
		Base	90.30
		Large	90.17
Seq2Tree			85.80

(Metrics: Accuracy)

Results

- Output examples for the task Code Documentation Generation - Javascript

Model	Size	Model Output
CodeTrans Single-Task Learning	Small	Returns true if the browser is a native element .
	Base	Returns whether the givenEnv should be focused .
CodeTrans Transfer Learning	Small	Checks if the current browser is on a standard browser environment .
	Base	Check if browser environment is a standard browser environment
	Large	Check if the environment is standard browser .
CodeTrans Multi-task Learning	Small	Returns true if the browser environment is a standard browser environment .
	Base	Checks if the current browser environment is a standard browser environment .
	Large	Determines if the current environment is a standard browser environment
CodeTrans Multi-task Learning Fine-tuning	Small	Standard browser environment has a notion of what React Native does not support it .
	Base	Check if the browserEnv is standard .
	Large	Checks if the browser is in a standard environment .
Code Snippet as Input		function isStandardBrowserEnv ( ) { if (typeof navigator !== 'undefined' && (navigator . product === 'ReactNative' | | navigator . product === 'NativeScript' | | navigator . product === 'NS')) { return false ; } return (typeof window !== 'undefined' && typeof document !== 'undefined') ; }
Golden Reference		Determine if we re running in a standard browser environment

Discussion

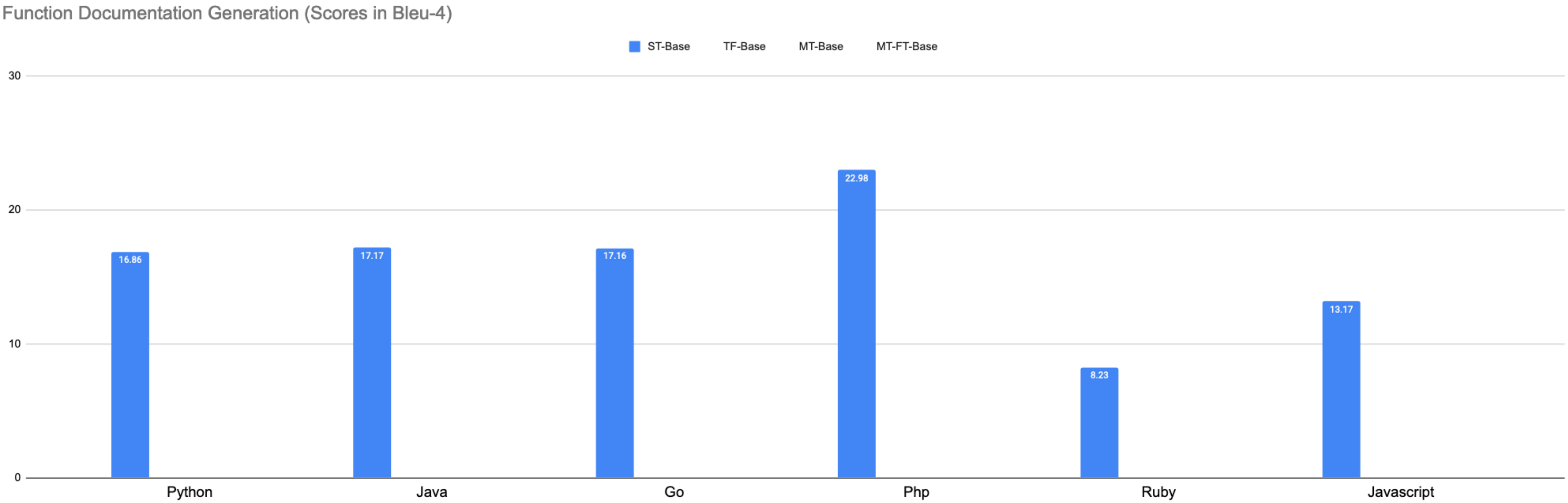
- Model Size:

		Small	Base	Large
Model Parameter (in Million)		60	220	770
Training Steps	Transfer Learning	500,000	500,000	240,000
	Multi-task Learning	500,000	500,000	260,000
Final Loss	Transfer Learning	0.926	0.586	0.476
	Multi-task Learning	0.887	0.590	0.4707
Time Cost	Transfer Learning	17 days	53 days	82 days
	Multi-task Learning	17 days	53 days	87 days

(Batch Size: 4096)

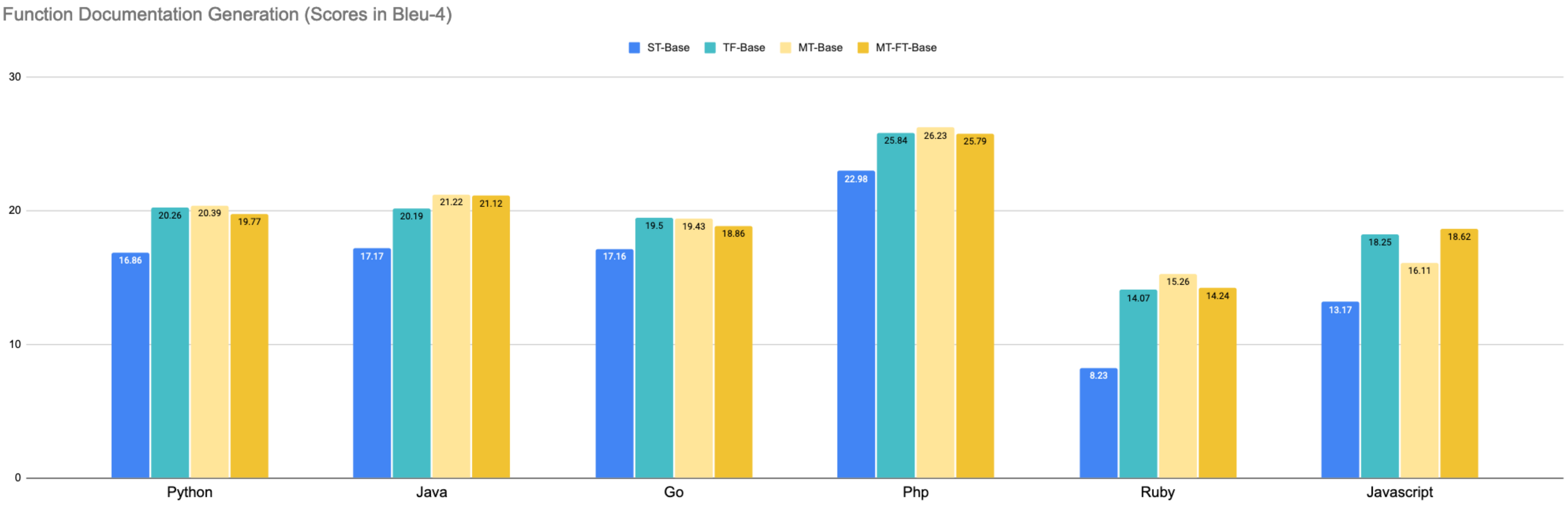
Discussion

- Single Task Learning vs other training strategies



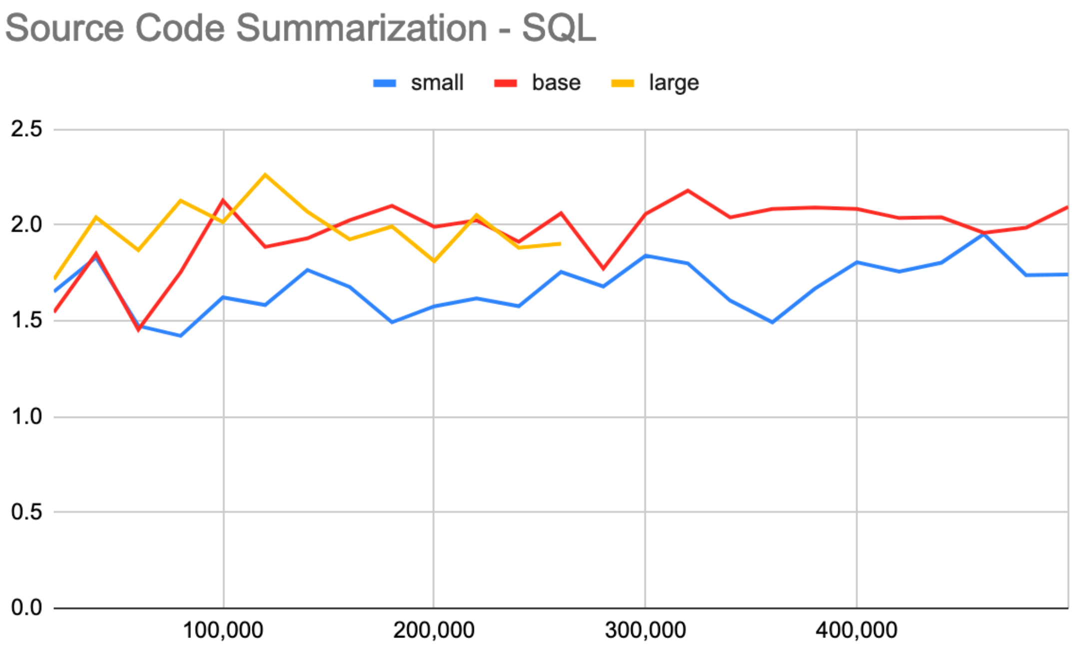
Discussion

- Single Task Learning vs other training strategies

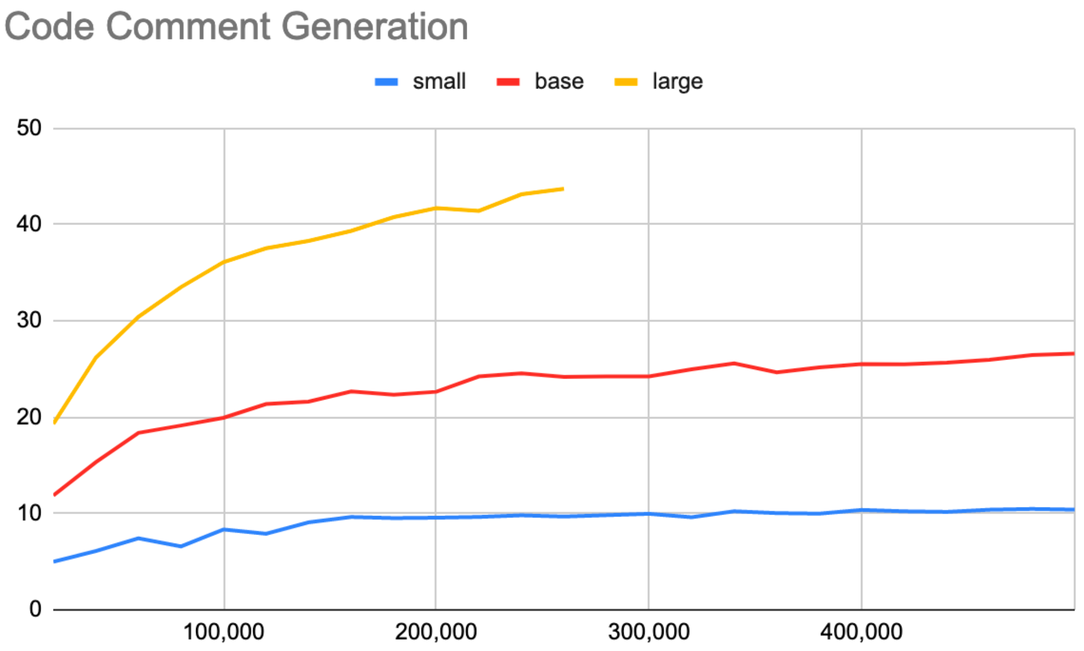


Discussion

- Multi-Task Learning: performance depends on the dataset attributes



Small dataset: 22,492 samples



Large dataset: 470,486 samples

Outline

- Motivation and Introduction
- Research Questions
- Model Architecture
- Tasks and Datasets
- Evaluation Metrics
- Results and Discussion
- **Conclusion and Future Work**
- References

Conclusion

?

What kind of natural language processing models would work best for tasks in the software development domain?

Conclusion

What kind of natural language processing models would work best for tasks in the software development domain?

Conclusion

✓ What kind of natural language processing models would work best for tasks in the software development domain?

? How would transfer learning improve the performance comparing with only training on the labeled data alone?

Conclusion

What kind of natural language processing models would work best for tasks in the software development domain?

How would transfer learning improve the performance comparing with only training on the labeled data alone?

Conclusion

✓ What kind of natural language processing models would work best for tasks in the software development domain?

✓ How would transfer learning improve the performance comparing with only training on the labeled data alone?

? Would transfer learning perform better than multi-task learning for the similar tasks?

Conclusion

- What kind of natural language processing models would work best for tasks in the software development domain?
- How would transfer learning improve the performance comparing with only training on the labeled data alone?
- Would transfer learning perform better than multi-task learning for the similar tasks?

Conclusion

- Uploaded 146 models in the Hugging Face Model Hub

The AI community building the future.

Build, train and deploy state of the art models powered by the reference open source in natural language processing.

Future Work

- Train 250,000 steps more for the large model
- More tasks / languages in the software development domain
- Try other masking techniques

Outline

- Motivation and Introduction
- Research Questions
- Model Architecture
- Tasks and Datasets
- Evaluation Metrics
- Results and Discussion
- Conclusion and Future Work
- **References**

References

- SEVERINI, SILVIA. "Multi-task Deep Learning in the Software Development domain." (2019)
- Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *arXiv preprint arXiv:1910.10683* (2019)
- Feng, Zhangyin, et al. "Codebert: A pre-trained model for programming and natural languages." *arXiv preprint arXiv:2002.08155* (2020)
- Husain, Hamel, et al. "Codesearchnet challenge: Evaluating the state of semantic code search." *arXiv preprint arXiv:1909.09436* (2019)
- Hu, Xing, et al. "Deep code comment generation." *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*. IEEE, 2018
- Jiang, Siyuan, and Collin McMillan. "Towards automatic generation of short summaries of commits." *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*. IEEE, 2017
- Gu, Xiaodong, et al. "Deep API learning." *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. 2016
- Polosukhin, Illia, and Alexander Skidanov. "Neural program search: Solving programming tasks from description and examples." *arXiv preprint arXiv:1802.04335* (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)*. 2016
- http://jalammar.github.io/illustrated-transformer/

Thanks!

Questions?

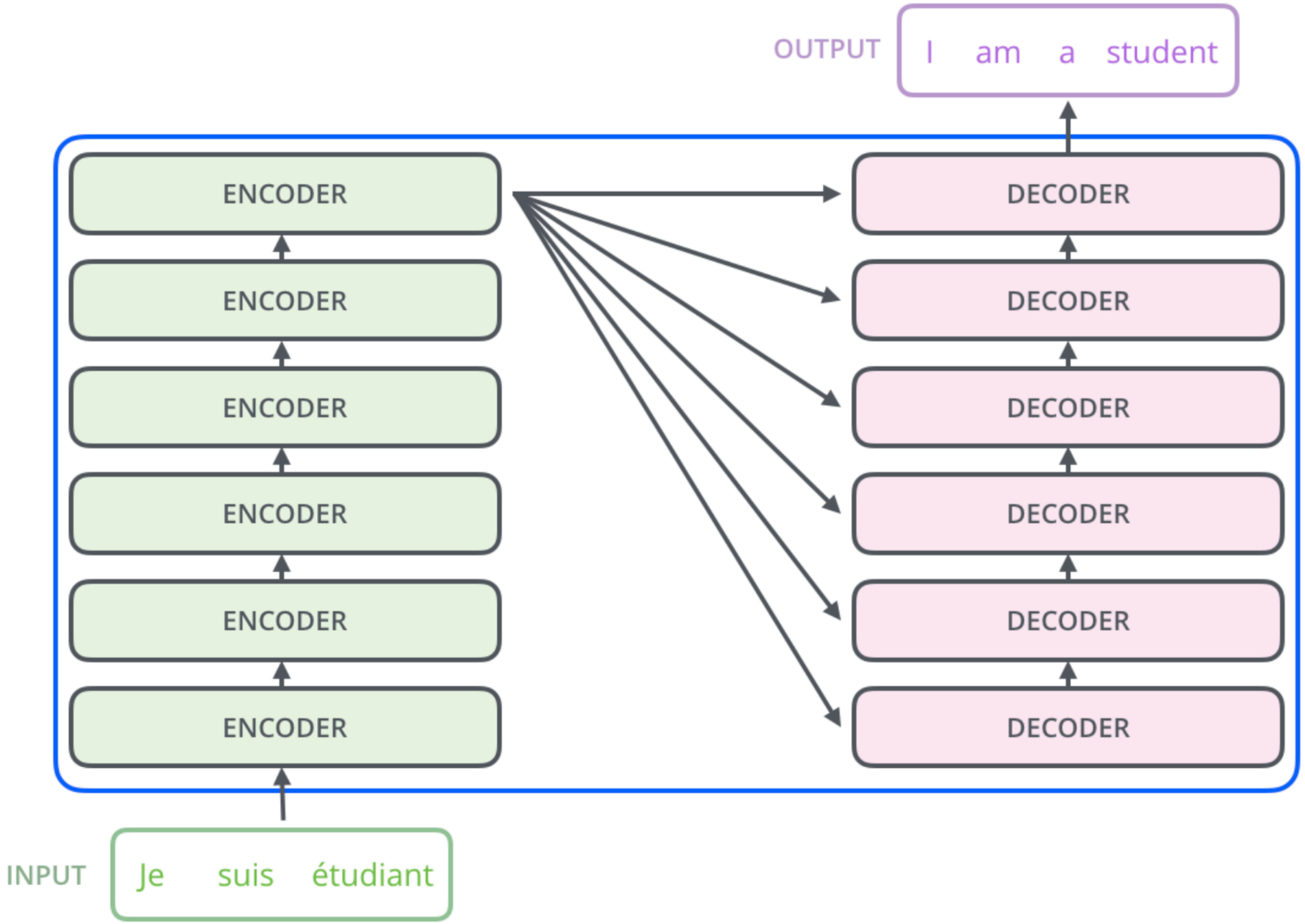
Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

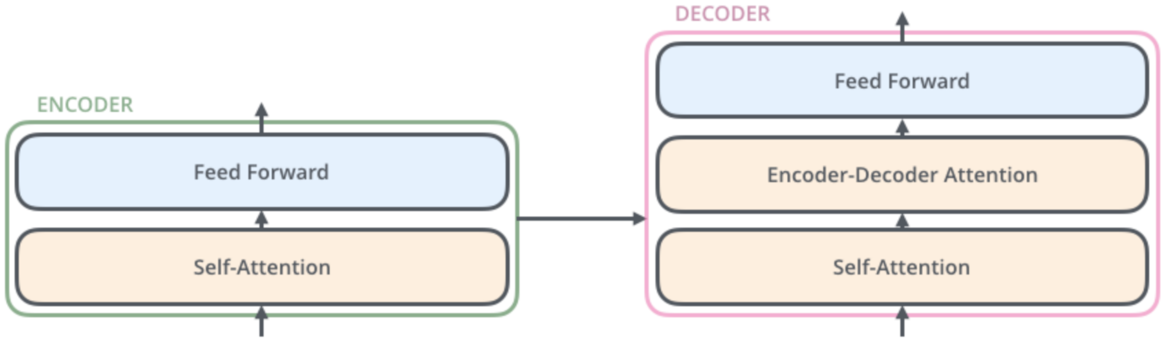
Tel +49.89.289. 17132
Fax +49.89.289.17136

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

Model Architecture



	Small	Base	Large
Number of Blocks Each	6	12	24
Dense Layer Output Dimension	2048	3072	4096
Attention Layer Key Value Dimension	64	64	64
Number of Attention Heads	8	12	16
Sub-layers and Embeddings Dimension	512	768	1024
Total Parameters (in Million)	60	220	770



Discussion

- Model Size
- Single Task Learning vs Transfer Learning vs Multi-Task Learning Fine-tuning
- Multi-Task Learning
- Evaluation Metrics:

Steps	2000	4000	6000	8000	10000	12000	14000	16000	18000	20000
BLEU	11.94	11.96	11.50	11.13	11.78	11.66	**12.36**	11.83	12.07	11.79
ROUGE-1	**39.59**	38.73	37.79	37.55	37.2	36.66	37.21	37.01	36.93	36.77
ROUGE-2	**19.79**	18.73	17.6	17.39	17.09	16.69	17.02	16.79	16.83	16.57
ROUGE-L	**37.39**	36.26	35.2	35.06	34.66	34.11	34.55	34.43	34.26	34.03

(Code Documentation Generation – Java Task on validation set after fine-tuning the multi-task learning base model.)