



How to be organized & productive during your PhD

Nicole Peinelt

29.10.2020

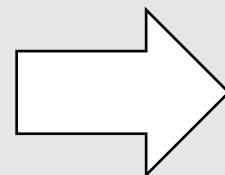
Overview

1. Planning your week
2. Making the most of meetings with your supervisor
3. Literature management for reading and writing
4. Summary

1. Planning your week

- Devi Parikh's calendar-based approach

<https://blog.usejournal.com/calendar-in stead-of-to-do-lists-9ada86a512dd>



Mon 11	Tue 12	Wed 13	Thu 14	Fri 15	Sat 16	Sun 17
07:30 Get ready	07:30 Get ready	07:40 Get results 3.5	07:30 Get ready	Get ready	07:30 Get ready	07:30 Get ready
Commute	Commute	08:30 Get ready	08:15 Get results from VM 5	Commute		
Get results 4	Get results 4	09:57 train	09:15 Prepare for meeting 5	Get results from VM 4		
Prediction overlap 5	PAWS doc topic baseline	Cod...	10:15 Skype 4	run BERT+topics on Semeval A and C	10:00 sideproject 5	
11:07 Prediction overlap 5	10:37 Run topic bert 5	Anal...	topic baseline for Semeval A and C	BERT+doc topic results...	explore properties of c...	
12:00 Skype 3	12:00 Lunch	NiNo	12:53 run BERT on Semeval A...	11:15 Get results, balance GPU load		
Email 4	implement BERT+word...	Commute	13:00 Lunch	12:00 NLIP Seminar 3	13:00 Lunch	
13:38 BERT+Topics 5	14:00 Reading Group 4	Lunch	14:15 run BERT on Semeval A...	Email/update RG annou...	14:22 explore properties of correct predictions	
14:45 BERT+Topics 4	15:00 debug BERT+topic architecture 3.5	Commute	14:19 Try to train model on (shuffled) subset o...	14:22 explore properties of correct predictions	15:00 Walk	14:45 Lunch
PT	PT	16:30 experiment with less hidden layers 5	14:45 Write experiment for (shuffled) Quora subset 4	15:45 emails 3.5	Prepare command	15:45 sideproject 3.5
16:45 prepare topic tu...		git commit 4	14:45 update START profile 3	Send reading group not...	Finish AthensNLP claim	
Commute	Commute	Commute	15:00 Plan schedule 3.5	Weekly review	Commute	
19:00 Dinner	18:28 Dinner	18:46 Dinner	18:00 sideproject 5	18:00 Dinner	18:15 Dinner	18:00 Dinner
19:45	19:45	19:45	19:45	19:45	19:45	19:45

Mon 11	Tue 12	Wed 13	Thu 14	Fri 15	Sat 16	Sun 17
07:30 Get ready	07:30 Get ready	07:40 Get results 3.5	07:30 Get ready	Get ready	07:30 Get ready	07:30 Get ready
Commute	Commute	08:30 Get ready	08:15 Get results from VM 5	Commute		
Get results 4	Get results 4	09:15 Prepare for meeting 5	09:45 run BERT+topics on Semeval A and C	Get results from VM 4		
Prediction overlap 5	PAWS doc topic baseline	10:15 Skype 4	BERT+doc topic results...	10:00 sideproject 5		
11:07 Prediction overlap 5	10:37 Run topic bert 5	NiNo	topic baseline for Seme...	explore properties of c...	11:15 Get results, balance GPU load	
12:00 Skype 3	12:00 Lunch	12:30 train	run BERT on Semeval A...	12:00 NLIP Seminar 3	12:45 sideproject 4	
Email 4	implement BERT+word...	Cod...	Lunch	13:00 Lunch	13:00 Lunch	
13:38 BERT+Topics 5	12:22 Lunch	12:30 train	fix BERT preprocessing for semeval A and C (different len of sent...)	Email/update RG annou...		
14:45 BERT+Topics 4	13:40 Reading Group 4	Commute	14:15 run BERT on Semeval A...	14:22 explore properties of correct predictions	14:45 Lunch	
PT	14:00 Reading Group 4	Lunch	14:19 Try to train model on (shuffled) subset o...	walk+tea	15:00 Walk	15:45 sideproject 3.5
PT	15:00 debug BERT+topic architecture 3.5	15:45 Write experiment for (shuffled) Quora subset 4	15:45 emails 3.5	Prepare command		
16:45 prepare topic tu...	16:30 experiment with less hidden layers 5	git commit 4	Send reading group not...	Finish AthensNLP claim		
Commute	Commute	update START profile 3	Plan schedule 3.5	Weekly review		
19:00 Dinner	18:28 Dinner	Commute	18:00 sideproject 5	Commute	18:00 Dinner	18:00 Dinner
19:45 Dinner	18:46 Dinner	18:46 Dinner	18:45 Dinner	18:15 Dinner		
	19:45 Get results from VM 4	20:00		check if experiment on		PT

Mon 11	Tue 12	Wed 13
07:30 Get ready	07:30 Get ready	07:40 Get results 3.5
Commute	Commute	08:30 Get ready
Get results 4	Get results 4	09:57 train
Prediction overlap 5	PAWS doc topic baseline	Cod...
11:07 Prediction overlap 5	10:37 Run topic bert 5	Anal...
12:00 Skype 3	check bert_topic results	NiNo
Email 4	implement BERT+word...	12:30 train
13:38 BERT+Topics 5	12:22 Lunch	Cod...
14:45 BERT+Topics 4	implement BERT+word...	Lunch
PT	14:00 Reading Group 4	Commute
PT	15:00 debug BERT+topic architecture 3.5	Lunch
16:45 prepare topic tu...	14:19 Try to train model on (shuffled) subset o...	14:19 Try to train model on (shuffled) subset o...
Commute	15:45 Write experiment for (shuffled) Quora subset 4	15:45 Write experiment for (shuffled) Quora subset 4
19:00 Dinner	git commit 4	git commit 4
19:45 Run	update START profile 3	update START profile 3
Commute	Commute	Commute
18:28 Dinner	18:46 Dinner	18:46 Dinner
19:45 Run	19:45 Get results from VM 4	20:00

Productivity-based calendar planning

- Based on Devi Parikh's calendar planning method (schedule each task in your calendar) with the following adjustments:
1. Keep separate calendars (and don't move tasks to the 'done' calendar), e.g.:
 - Blue = major tasks (contribute to main professional goal)
 - Yellow = minor tasks (less important, but need to get done)
 - Pink = meeting
 - Light green = life
 - Dark green = health & wellbeing
 - Grey = travel & commuting
 2. Record your productivity levels
 - 1 = unproductive, barely any progress
 - 2 = got work done, but slow & easily distracted
 - 3 = normal 'baseline' productivity
 - 4 = productive
 - 5 = flow state
 3. Review what is/isn't working
 4. Analyse your calendar and productivity data with Timetrack

Fri 15

Sat 16

Sun 17

Weekly review

- At the end of each week, reflect on your week and ask yourself:
 - 1) What went well?
 - 2) What didn't?
 - 3) What can I improve next week?
- Keep notes to hold yourself accountable and find your ideal working style over time

Schedule

Fri 15	Sat 16	Sun 17
Get ready	07:30 Get ready	07:30 Get ready
Commute		
Get results from VM 4		
09:45 run BERT+topics on Semeval A and C		
BERT+doc topic results...		
explore properties of c...		
12:00 NLIP Seminar 3		
13:00 Lunch		
Email/update RG annou...		
14:22 explore properties of correct predictions		
walk+tea		
Prepare command		
Finish AthensNLP claim		
Weekly review		
Commute		
18:15 Dinner		
check if experiment on		

Weekly review

Add Location

15 Nov 2019 17:08 to 17:45

Add Invitees

This week:

- + felt productive, made good progress (got promising BERT+topic results), hopeful to start writing paper next week
- feeling tired/exhausted, need to sleep more
- often worked after coming home this week to submit new jobs

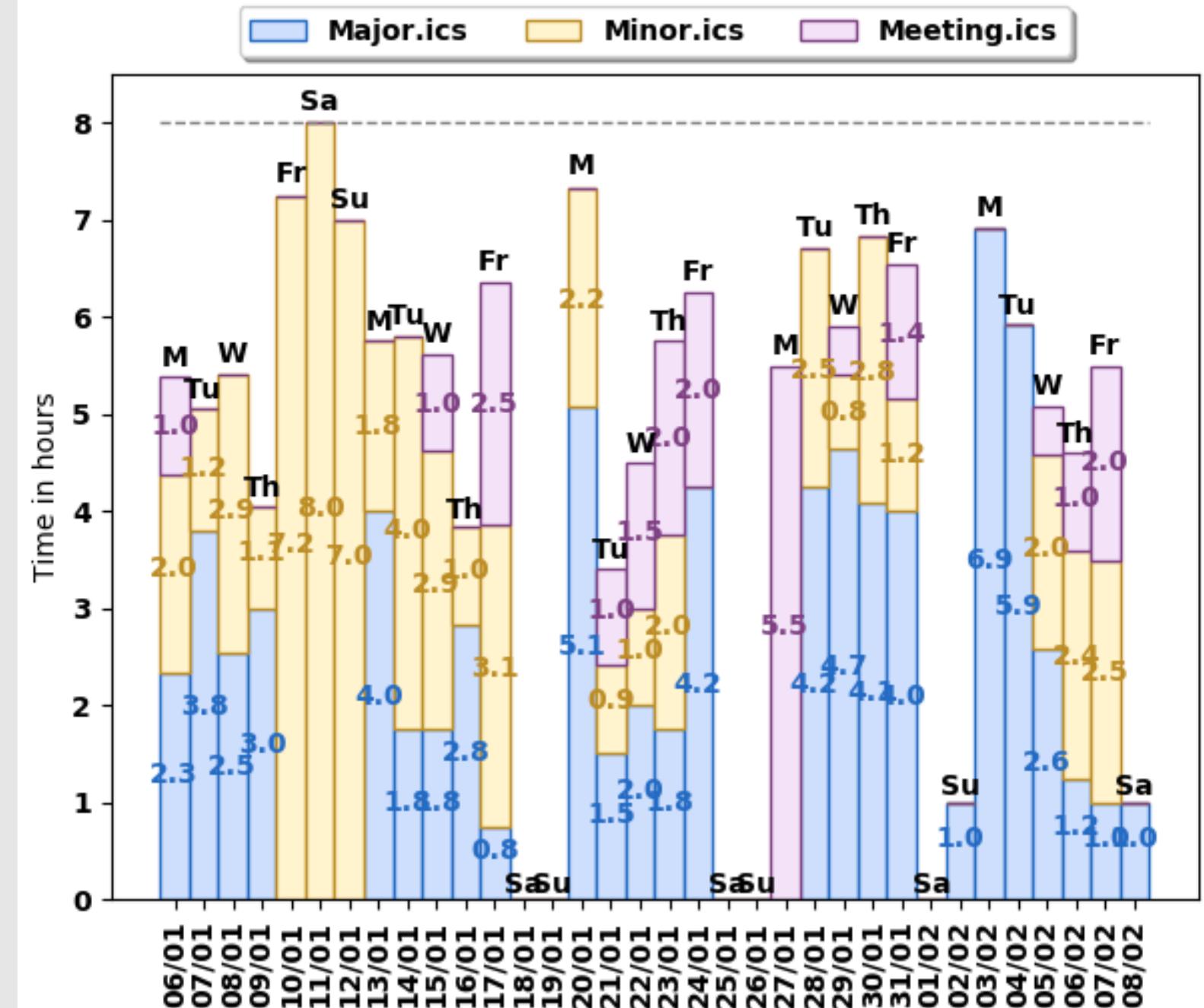
Next week:

- sleep earlier (10:30pm)
- get up earlier (7am)
- don't get discouraged by negative results, focus on the process & keep going
- try not to work after coming home as not sustainable in the long-term

Add URL or Attachments

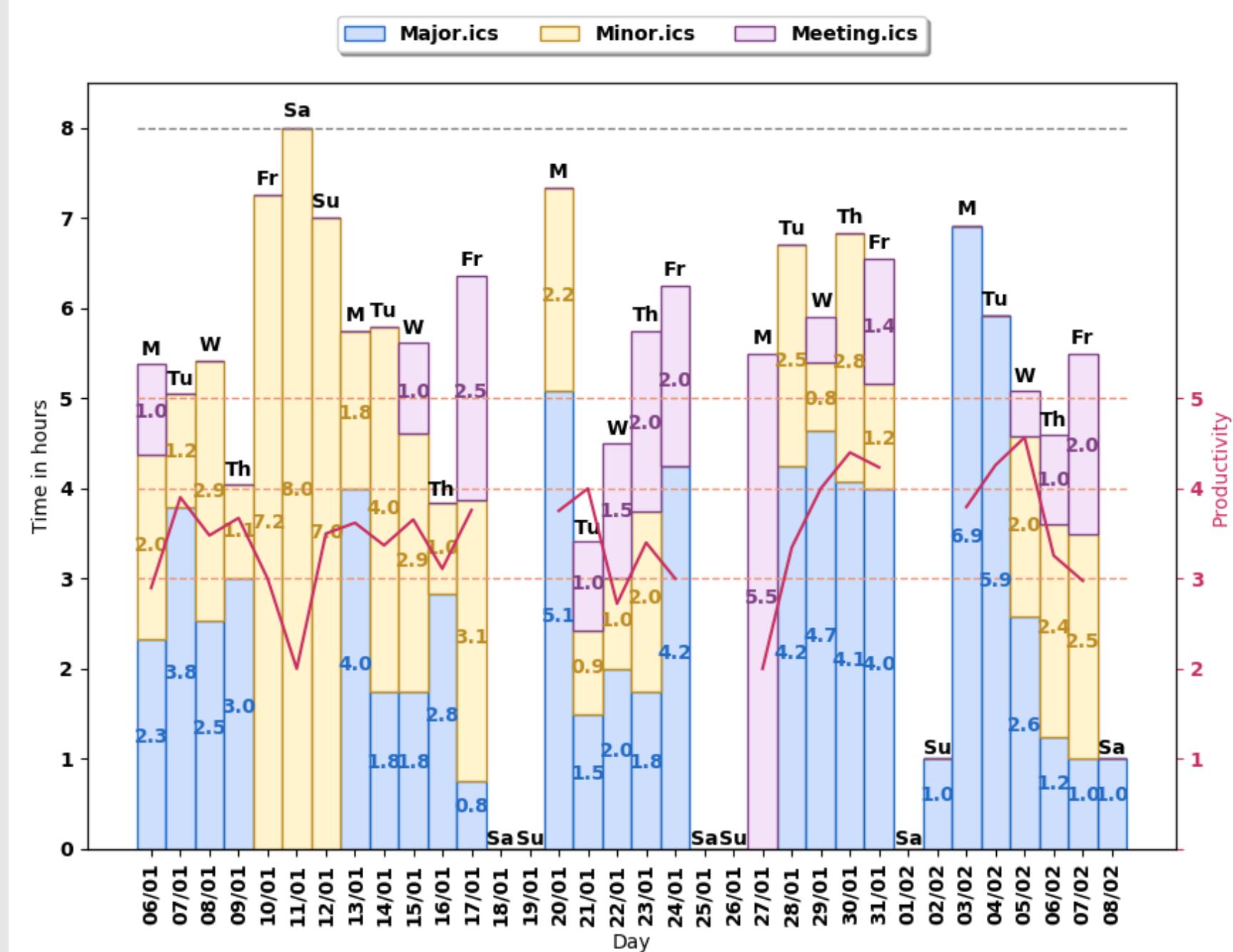
Quantitative summary

- Analyse working hours by category over time with the Timetrack tool
- Code available at:
<https://github.com/wuningxi/Timetrack>



Qualitative summary

- Visualisation of productivity levels
- Focus on working smarter, not more
- Code available at:
<https://github.com/wuningxi/Timetrack>



Calendar-based time management

- How:
 - Decide when to work & when not
 - Schedule each task in your calendar (assign timeslot & importance)
 - Give each task a productivity score after it is completed
 - Reflect on your week & productivity patterns based on a weekly review session and the Timetrack tool
- Why:
 - Hold yourself accountable
 - Allocate limited resources (time & energy) based on a task's importance
 - Visualise that every day counts
 - Find out what works for you & adjust

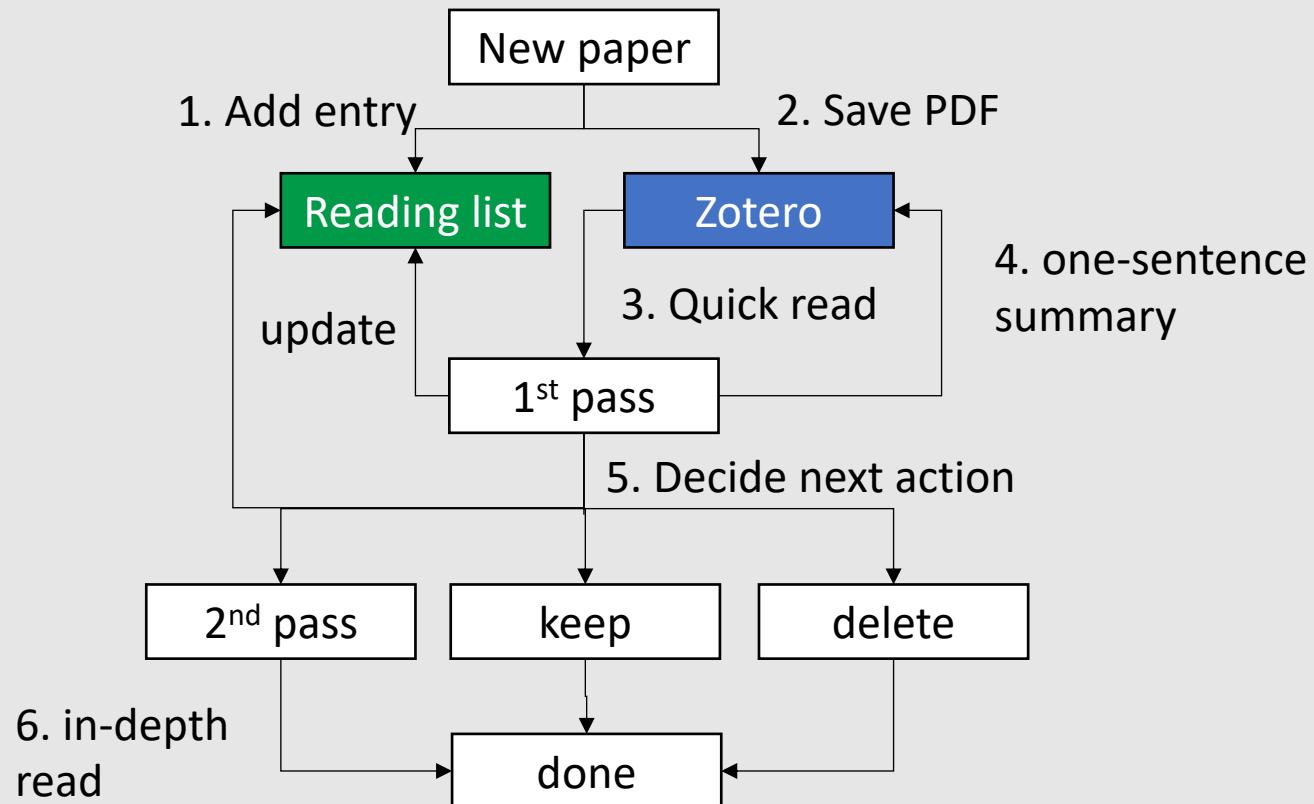
2. Making the most of supervisor meetings

- How:
 - For each meeting, prepare a couple of PowerPoint slides:
 - 1 slide with date of meeting
 - 1 slide with talking points for the meeting (meeting agenda)
 - 3-5 slides with main progress (e.g. result tables, findings, summary of literature, problems...)
 - At the end of the meeting:
 - 1 slide with action points for next week
- Why:
 - Basis for productive discussion
 - Keeps you on track
 - Reminds your supervisor of the current status of your project
 - Documentation of your progress & valuable future reference

3. Literature management basics

- How:
 - Keep everything in a digital format, don't print out your papers
 - Instead read papers with your favourite PDF viewer and make digital annotations (e.g. highlights and notes)
- Why:
 - Annotating a paper takes time, but increases its value (e.g. easier to understand & summarise key points)
 - Printed papers will stack up over time, making it difficult & time consuming to find relevant information
 - Infeasible to carry all your printed papers with you

3. Literature management: Reading



Reading list

1	Semantic Similarity Detection papers							
2								
3	Why read: keep up to date with what is happening in CQA, paraphrase detection and non-factoid question answering							
4								
5	When	Title	Link	topic	first pass	second pass	Zotero	
6	07.01.2020	Can AI Generate Love Advice?: Toward Neural Answer Generation f	https://arxiv.org/abs/1912.10163	CQA/non-factoid QA	n/a			
7	07.01.2020	Essential Sentences for Navigating Stack Overflow Answers	https://arxiv.org/abs/1912.13455	CQA/non-factoid QA	n/a			
8	07.01.2020	Knowledge-Enhanced Attentive Learning for Answer Selection in Co	https://arxiv.org/abs/1912.07915	CQA/non-factoid QA	n/a			
9	07.01.2020	A Multi-cascaded Model with Data Augmentation for Enhanced Par	https://arxiv.org/abs/1912.12068	paraphrase detection	n/a			
10	07.01.2020	CoSimLex: A Resource for Evaluating Graded Word Similarity in Con	https://arxiv.org/abs/1912.05320	other semantic similarity	n/a			
11	07.01.2020	Semantic Similarity To Improve Question Understanding in a Virtua	https://arxiv.org/abs/1912.07421	other semantic similarity	n/a			
12	20.01.2020	Building Interactive Sentence-aware Representation based on Gen	https://www.sciencedirect.com/sci	CQA	done	waiting	wu_building_2020	
13	22.01.2020	Multi-level Head-wise Match and Aggregation in Transformer for To	https://arxiv.org/abs/2001.07234	paraphrase detection	done		wang_multi-level_2020	
14	28.01.2020	Is BERT Really Robust? A Strong Baseline for Natural Language Att	https://arxiv.org/abs/1907.11932	adversarial	done		jin_is_2020	
15	30.01.2020	What's happened in MOOC Posts Analysis, Knowledge Tracing and I	https://arxiv.org/pdf/2001.09830.pdf	MOOC	waiting		ravikiran_whats_2020	
16	31.01.2020	TANDA: Transfer and Adapt Pre-Trained Transformer Models for An	https://arxiv.org/abs/1911.04118	answer selection; bert finetuning	done		garg_tanda_2019	
17	30.10.2020	Gated Semantic Difference Based Sentence Semantic Equivalence I	https://ieeexplore.ieee.org/abstract	paraphrase detection	waiting		liu_gated_2020	
18								
19	BERT modification papers							
20								
21	Why read: Check how papers have enhanced BERT, potentially useful for early topic fusion in BERT							
22								
23	When	Title	Link	Topic	first pass	second pass	Zotero	
24	08.01.2020	Semantics-aware BERT for Language Understanding	https://arxiv.org/abs/1909.02209v2	external BERT modification	done		zhang_semantics-aware_2020	
25	07.01.2020	Stacked DeBERT: All Attention in Incomplete Data for Text Classific	https://arxiv.org/abs/2001.00137	internal BERT modification	waiting			
26	12.11.2019	Knowledge Enhanced Contextual Word Representations	https://www.aclweb.org/anthology	internal BERT modification	waiting		peters_knowledge_2019	
27	14.01.2020	Unsupervised Domain Adaptation on Reading Comprehension	https://arxiv.org/abs/1911.06137	domain adaptation	n/a			
28	07.01.2020	BERTQA -- Attention on Steroids	https://arxiv.org/abs/1912.10435	external BERT modification	n/a			
29	16.01.2020	FGN: Fusion Glyph Network for Chinese Named Entity Recognition	https://arxiv.org/abs/2001.05272	external BERT modification	done	possibly check 3 cited papers	xuan_fgn_2020	
30	13.01.2020	Enriching Pre-trained Language Model with Entity Information for R	https://arxiv.org/abs/1905.08284	external BERT modification	done		wu_enriching_2019	
31	16.01.2020	A BERT based Sentiment Analysis and Key Entity Detection Approac	https://arxiv.org/abs/2001.05326	input modification	done		zhao_bert_2020	
32	16.01.2020	Constructing Artificial Data for Fine-tuning for Low-Resource Biome	https://arxiv.org/abs/1910.09255	input modification	done		singh_constructing_2020	
33	14.01.2020	AdaBERT: Task-Adaptive BERT Compression with Differentiable Ne	https://arxiv.org/abs/2001.04246	lightweight	done		chen_adabert_2020	
34	14.01.2020	ALBERT: A Lite BERT for Self-supervised Learning of Language Repr	https://arxiv.org/abs/1909.11942	lightweight	done		lan_albert_2020	
35	17.01.2019	Knowledge Distillation from Internal Representations	https://arxiv.org/abs/1910.03723	lightweight	done		aguilar_knowledge_2020	
36	17.01.2019	DistilBERT, a distilled version of BERT: smaller, faster, cheaper and	https://arxiv.org/abs/1910.01108	lightweight	done		sanh_distilbert_2019	
37	07.01.2020	GlossBERT: BERT for Word Sense Disambiguation with Gloss Know	https://arxiv.org/abs/1908.07245		delete			

Managing PDFs with Zotero (with better Bibtex addon)

The screenshot shows the Zotero application interface. On the left is a sidebar with a tree view of library collections and sub-collections. The 'Papers Finished' collection is expanded, showing sub-folders like 'BERT injection', 'Classifiers', 'LexicalOverlap', 'Proposal', and 'Semantic Similarity Detect...'. A folder named 'BERT finetuning/modifi...' is highlighted. The main pane displays a list of research papers with columns for Title, Creator, and a preview icon. The right pane provides a detailed view of a selected paper, with tabs for Info, Notes, Tags, and Related. The 'Info' tab is active, showing fields such as Citation Key (set to 'aguilar_knowledge_2020'), Item Type (Journal Article), Title (Knowledge Distillation from Internal Representations), and various author details. The 'Notes' tab contains a note about knowledge distillation being conducted. The 'Tags' and 'Related' tabs are also visible.

Zotero key

1st pass

- Read abstract, introduction, figures, tables and conclusion
- Understand main idea
- Write one sentence summary
- Decide next action: discard/keep/2nd pass

Garg et al. - 2019 - TANDA Transfer and Adapt Pre-Trained Transformer.pdf (page 1 of 9)

Contents Pane Tool Mode Add Note Favorite Colors Rotate Left Rotate Right Previous/Next Page Back/Forward

sentence selection by (1) finetuning on large dataset before (2) finetuning on target dataset

TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection

Siddhant Garg *
University of Wisconsin-Madison
Madison, WI, USA
sidgarg@cs.wisc.edu

Thuy Vu
Amazon Alexa
Manhattan Beach, CA, USA
thuyvu@amazon.com

Alessandro Moschitti
Amazon Alexa
Manhattan Beach, CA, USA
amosch@amazon.com

Abstract

We propose TANDA, an effective technique for fine-tuning pre-trained Transformer models for natural language tasks. Specifically, we first transfer a pre-trained model into a model for a general task by fine-tuning it with a large and high-quality dataset. We then perform a second fine-tuning step to adapt the transferred model to the target domain. We demonstrate the benefits of our approach for answer sentence selection, which is a well-known inference task in Question Answering. We built a large scale dataset to enable the transfer step, exploiting the Natural Questions dataset. Our approach establishes the state of the art on two well-known benchmarks, WikiQA and TREC-QA, achieving MAP scores of 92% and 94.3%, respectively, which largely outperform the previous highest scores of 83.4% and 87.5%, obtained in very recent work. We empirically show that TANDA generates more stable and robust models reducing the effort required for selecting optimal hyper-parameters. Additionally, we show that the transfer step of TANDA makes the adaptation step more robust to noise. This enables a more effective use of noisy datasets for fine-tuning. Finally, we also confirm the positive impact of TANDA in an industrial setting, using domain specific datasets subject to different types of noise.

1 Introduction

In recent years, virtual assistants have become a central asset for technological companies. This has increased the interest of AI researchers in studying and developing conversational agents, some popular examples being Google Home, Siri and Alexa. This has renewed the research interest in Question Answering (QA) and, in particular, in two main tasks: (i) answer sentence selection (AS2), which, given a question

arXiv:1911.04118v2 [cs.CL] 20 Nov 2019

2nd pass

- Understand details, make detailed annotation (using colour-coded notes)

arXiv:2002.01808v1 [cs.CL] 5 Feb 2020

K-ADAPTER: Infusing Knowledge into Pre-Trained Models with Adapters

Ruize Wang ^{*1} Duyu Tang ² Nan Duan ² Zhongyu Wei ³ Xuanjing Huang ⁴ Jianshu ji ⁵ Cuihong Cao ⁵
Dixin Jiang ⁶ Ming Zhou ²

Abstract

We study the problem of injecting knowledge into large pre-trained models like BERT and RoBERTa. Existing methods typically update the original parameters of pre-trained models when injecting knowledge. However, when multiple kinds of knowledge are injected, they may suffer from catastrophic forgetting. To address this, we propose K-ADAPTER, which remains the original parameters of the pre-trained model fixed and supports continual knowledge infusion. Taking RoBERTa as the pre-trained model, K-ADAPTER has a neural adapter for each kind of infused knowledge, like a plug-in connected to RoBERTa. There is no information flow between different adapters, thus different adapters are efficiently trained in a distributed way. We inject two kinds of knowledge, including factual knowledge obtained from automatically aligned text-triplets on Wikipedia and Wikidata, and linguistic knowledge obtained from dependency parsing. Results on three knowledge-driven tasks (total six datasets) including relation classification, entity typing and question answering demonstrate that each adapter improves the performance, and the combination of both adapters brings further improvements. Probing experiments further show that K-ADAPTER captures richer factual and commonsense knowledge than RoBERTa.

1. Introduction

Language representation models, which are pre-trained on large-scale text corpus through unsupervised objectives like (masked) language modeling, such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018; 2019), XLNet (Yang

^{*}Work is done during internship at Microsoft Research Asia
¹Academy for Engineering and Technology, Fudan University, Shanghai, China ²Microsoft Research Asia, Beijing, China ³School of Data Science, Fudan University, Shanghai, China ⁴School of Computer Science, Fudan University, Shanghai, China ⁵Microsoft AI and Research, Redmond WA, USA ⁶Microsoft Search Technology Center Asia, Beijing, China.

et al., 2019), RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2019), have established state-of-the-art performances on various NLP downstream tasks.

Despite the huge success of these large pre-trained models in empirical studies, recent studies suggest that models learned in such an unsupervised manner struggle to capture rich knowledge. For example, Poerner et al. (2019) suggest that although language models do well in reasoning about the surface form of entity names, they fail in capturing rich factual knowledge. Kassner & Schütze (2019) observe that BERT mostly did not learn the meaning of negation (e.g. "not"). Talmor et al. (2019) find that language models fail completely on half of eight reasoning tasks that require symbolic operations such as comparison, conjunction, and composition. These observations motivate us to study the injection of knowledge into pre-trained models like BERT and RoBERTa.

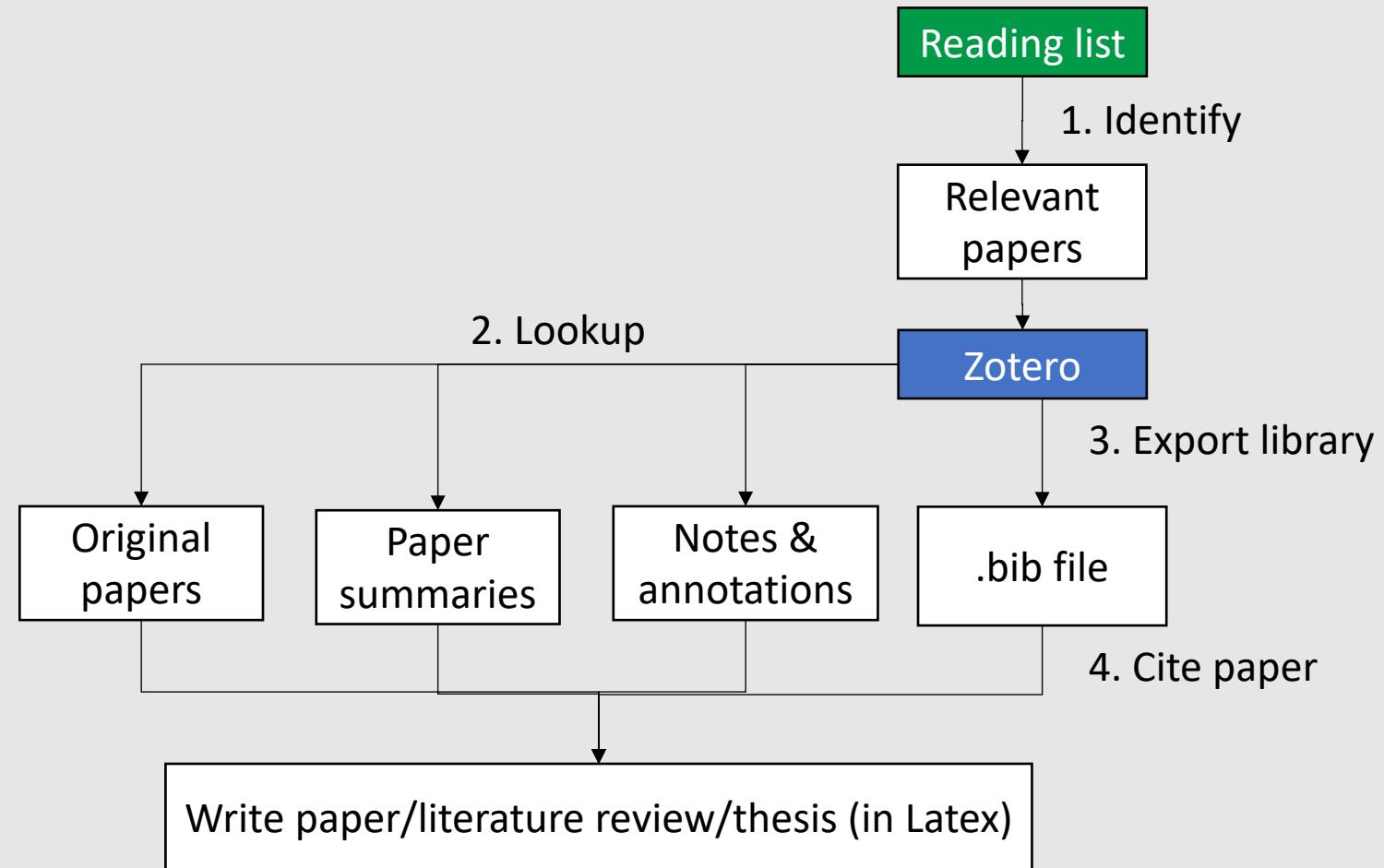
Recently, some efforts have been made to exploit injecting knowledge into pre-trained language models (Zhang et al., 2019; Lauscher et al., 2019; Levine et al., 2019; Peters et al., 2019; He et al., 2020; Xiong et al., 2020). Most previous works (as shown in Table 1) augment the standard language modeling objective with knowledge-driven objectives and update model parameters in a multi-task learning manner. Although these methods, with updated pre-trained models, obtain better performance on downstream tasks, they fail to continual learning (Kirkpatrick et al., 2017). Model parameters need to be retrained when we want to inject many new kinds of knowledge, which may result in the catastrophic forgetting of previously injected knowledge. Meanwhile, the resulting pre-trained models produce entangled representations, which makes it hard to investigate the effect of each knowledge when multiple kinds of knowledge are injected.

In this paper, we propose K-ADAPTER, a flexible and simple approach that infuses knowledge into large pre-trained models. K-ADAPTER has attractive properties including supporting continual knowledge infusion and producing disentangled representations. It remains the original representation of a pre-trained model unchanged and exports different representations for different types of infused knowledge. This is achieved by the integration of compact neural models, dubbed adapters here. Adapters are knowledge-specific models plugged outside of a pre-trained model, whose in-

limitations of pretrained language models

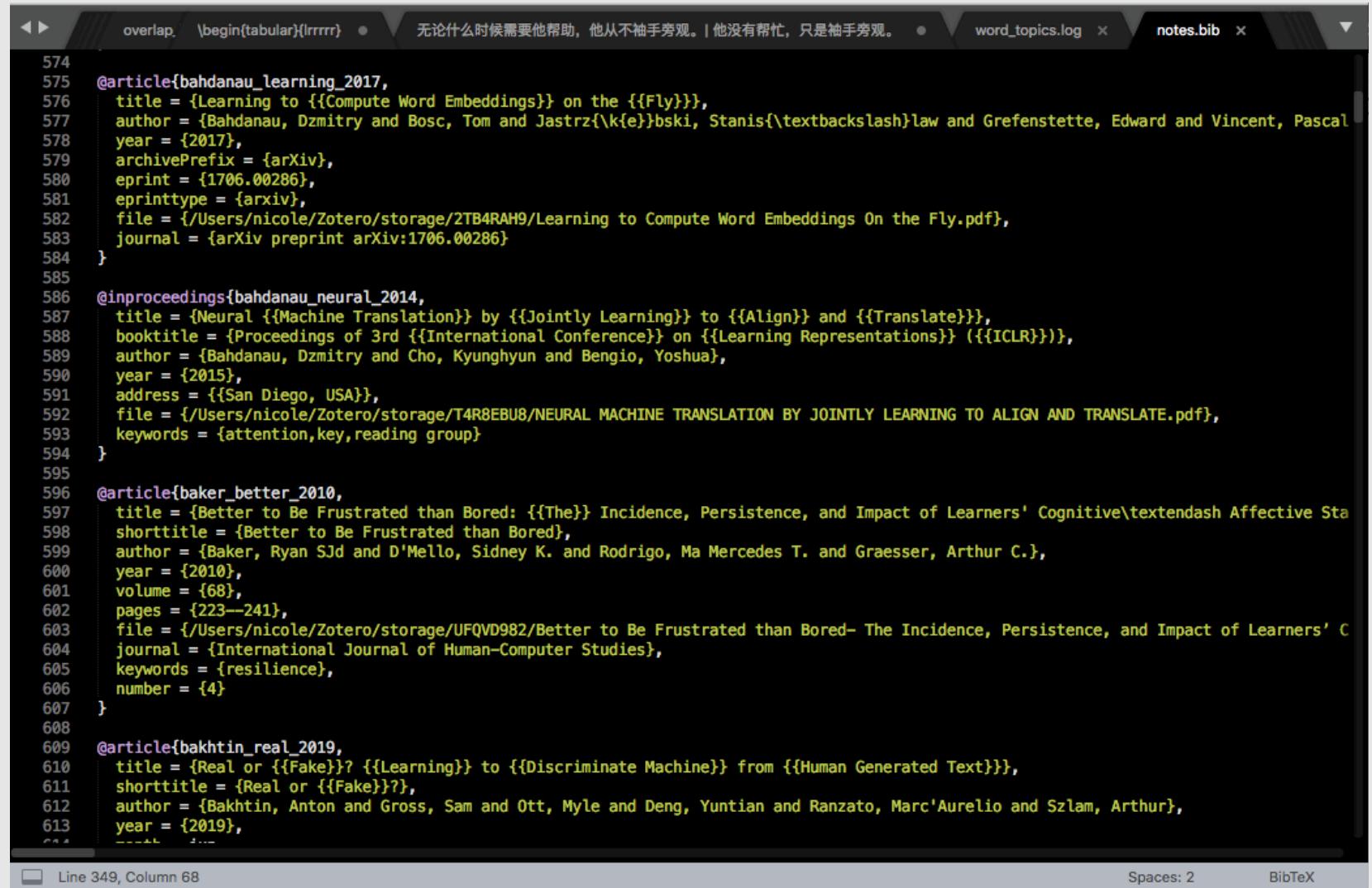
previous knowledge injection work

3. Literature management: Writing



Export BibTex file for all references

- Export Zotero library as .bib file to Latex project directory or use Zotero-Overleaf integration



The screenshot shows a LaTeX editor interface with multiple tabs at the top: 'overlap', '\begin{tabular}{|rrrrr|}', '无论什么时候需要他帮助, 他从不袖手旁观。 | 他没有帮忙, 只是袖手旁观。', 'word_topics.log', and 'notes.bib'. The main window displays a BibTeX file with the following entries:

```
overlap. \begin{tabular}{|rrrrr|} 无论什么时候需要他帮助, 他从不袖手旁观。 | 他没有帮忙, 只是袖手旁观。 word_topics.log notes.bib

574 @article{bahdanau_learning_2017,
575   title = {Learning to {{Compute Word Embeddings}} on the {{Fly}}},
576   author = {Bahdanau, Dzmitry and Bosc, Tom and Jastrz{\k{e}}bski, Stanis{\textbackslash}aw and Grefenstette, Edward and Vincent, Pascal},
577   year = {2017},
578   archivePrefix = {arXiv},
579   eprint = {1706.00286},
580   printtype = {arxiv},
581   file = {/Users/nicole/Zotero/storage/2TB4RAH9/Learning to Compute Word Embeddings On the Fly.pdf},
582   journal = {arXiv preprint arXiv:1706.00286}
583 }
584
585 @inproceedings{bahdanau_neural_2014,
586   title = {Neural {{Machine Translation}} by {{Jointly Learning}} to {{Align}} and {{Translate}}},
587   booktitle = {Proceedings of 3rd {{International Conference}} on {{Learning Representations}} ({{ICLR}})},
588   author = {Bahdanau, Dzmitry and Cho, Kyunghyun and Bengio, Yoshua},
589   year = {2015},
590   address = {{San Diego, USA}},
591   file = {/Users/nicole/Zotero/storage/T4R8EBU8/NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE.pdf},
592   keywords = {attention, key, reading group}
593 }
594
595 @article{baker_better_2010,
596   title = {Better to Be Frustrated than Bored: {{The}} Incidence, Persistence, and Impact of Learners' Cognitive\textendash Affective Sta
597   shorttitle = {Better to Be Frustrated than Bored},
598   author = {Baker, Ryan SJ and D'Mello, Sidney K. and Rodrigo, Ma Mercedes T. and Graesser, Arthur C.},
599   year = {2010},
600   volume = {68},
601   pages = {223–241},
602   file = {/Users/nicole/Zotero/storage/UFQVD982/Better to Be Frustrated than Bored- The Incidence, Persistence, and Impact of Learners' C
603   journal = {International Journal of Human-Computer Studies},
604   keywords = {resilience},
605   number = {4}
606 }
607
608 @article{bakhtin_real_2019,
609   title = {Real or {{Fake}}? {{Learning}} to {{Discriminate Machine}} from {{Human Generated Text}}},
610   shorttitle = {Real or {{Fake}}?},
611   author = {Bakhtin, Anton and Gross, Sam and Ott, Myle and Deng, Yuntian and Ranzato, Marc'Aurelio and Szlam, Arthur},
612   year = {2019},
613   ----
614 }
```

Line 349, Column 68 Spaces: 2 BibTeX

Write paper in Latex → \cite{zotero_key}

The screenshot shows the Overleaf LaTeX editor interface. On the left, the 'Source' tab is active, displaying the LaTeX code for a research paper. The code includes sections like 'Introduction' and 'Abstract', and discusses semantic similarity detection. It uses the `\cite` command to cite references from Zotero. On the right, the 'Rich Text' tab is active, showing the generated PDF output. The PDF title is 'tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection'. It features a header with authors Nicole Peinelt, Dong Nguyen, and Maria Liakata, and their institutional affiliations. The abstract section describes the work on semantic similarity detection, mentioning the integration of topic models and BERT. The Overleaf interface includes standard document navigation tools like 'Recompile', 'Preview', and 'File'.

```
85
86 \begin{document}
87 \maketitle
88 \begin{abstract}
89 Semantic similarity detection is a fundamental task in natural language understanding. Adding topic information has
90 been useful for previous feature-engineered semantic similarity models as well as neural models for other tasks. There
91 is currently no standard way of combining topics with pretrained contextual representations such as BERT.
92 We propose a novel topic-informed BERT-based architecture for pairwise semantic similarity detection and show that our
93 model improves performance over strong neural baselines across a variety of English language datasets. We find that
94 the addition of topics to BERT helps particularly with resolving domain-specific cases.
95 \end{abstract}
96 Modelling the semantic similarity between a pair of texts is a crucial NLP task with applications ranging from
97 question answering to plagiarism detection. % and evaluation for language generation.
98 A variety of models have been proposed for this problem, including traditional feature-engineered techniques
99 \cite{filice_kelp_2017}, hybrid approaches \cite{wu_ecnu_2017,feng_beihang-msra_2017,koreeda_bunji_2017} and purely
100 neural architectures \cite{wang_bilateral_2017,tan_multiway_2018,deriu_swissalps_2017}.
101 Recent pretrained contextualised representations such as ELMo \cite{peters_deep_2018-1} and BERT
102 \cite{devlin_bert_2019} have led to impressive performance gains across a variety of NLP tasks, including semantic
103 These models leverage large amounts of data to pretrain text encoders (in contrast to just individual word embeddings
104 as in previous work) and have established a new pretrain-finetune paradigm.
105 While large improvements have been achieved on paraphrase detection \cite{tomar_neural_2017,gong_natural_2018},
106 semantic similarity detection in Community Question Answering (CQA) remains a challenging problem. CQA leverages
107 user-generated content from question answering websites (e.g. StackExchange) to answer complex real-world questions
108 \cite{SemEval-2017:task3}. The task requires modelling the relatedness between question-answer pairs which can be
109 challenging due to the highly domain-specific language of certain online forums and low levels of direct text overlap
110 between questions and answers.
111 % \todo{[more about previous approaches, what was the impact of topic information in papers, why does it work]}
112 Topic models may provide additional signals
113 % for determining relevant information %despite different surface structure
114 for semantic similarity, as earlier feature-engineered models for semantic similarity detection successfully
115 incorporated topics \cite{chien_ranking_2009,tran_jaist_2015,mihaylov_semanticz_2016-1,wu_ecnu_2017}.
116 They could be especially useful for dealing with domain-specific language since topic models have been exploited for
```

tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection

Nicole Peinelt^{1,2} and Dong Nguyen^{1,3} and Maria Liakata^{1,2}

¹The Alan Turing Institute, London, UK
²University of Warwick, Coventry, UK
³Utrecht University, Utrecht, The Netherlands
n.peinelt, m.liakata}@warwick.ac.uk, dnguyen@turing.ac.uk

Abstract

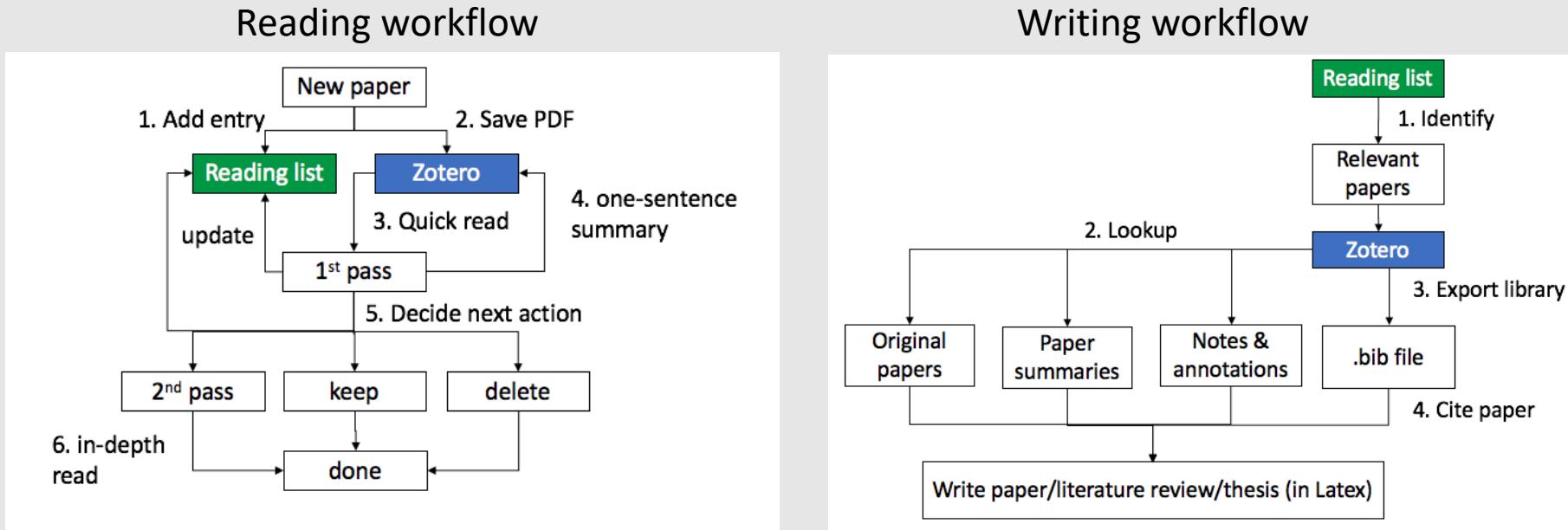
Semantic similarity detection is a fundamental task in natural language understanding. Adding topic information has been useful for previous feature-engineered semantic similarity models as well as neural models for other tasks. There is currently no standard way of combining topics with pretrained contextual representations such as BERT. We propose a novel topic-informed BERT-based architecture for pairwise semantic similarity detection and show that our model improves performance over strong neural baselines across a variety of English language datasets. We find that the addition of topics to BERT helps particularly with resolving domain-specific cases.

1 Introduction

Modelling the semantic similarity between a pair of texts is a crucial NLP task with applications ranging from question answering to plagiarism detection. A variety of models have been proposed for this problem, including traditional feature-engineered techniques [Filice et al., 2017], hybrid approaches [Wu et al., 2017; Feng et al., 2017]

Literature management

- How:



- Why:

- Reading process builds up valuable & instantly accessible resources for future writing
- All tools connected by unique Zotero paper key
 - Easy to find and cite previously read papers
 - Saves a lot of time!

Being organized & productive during your PhD

- Plan your week
- Prepare for weekly meetings
- Manage your literature
- Choose & adapt methods which work for you
- Focus on the process, not the outcome
- Find a healthy work-life balance

