The Alan Turing Institute

WARWICK
THE UNIVERSITY OF WARWICK

How to be organized & productive during your PhD

Nicole Peinelt

29.10.2020

Overview

1. Planning your week
2. Making the most of weekly meetings with your supervisor
3. Literature management
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

- 07:30 Get ready
- Commute
- Get results 4
- Prediction overlap 5
- 11:07 Prediction overlap 5
- 12:00 Skype 3
- 12:00 Lunch
- Email 4
- 13:38 BERT+Topics 5
- 14:45 BERT+Topics 4
- PT
- PT
- PT
- 16:45 prepare topic tu...
- Commute
- 19:00 Dinner
- 19:45

Tue 12

- 07:30 Get ready
- Commute
- Get results 4
- PAWS doc topic baseline
- 10:37 Run topic bert 5
- check bert_topic results
- implement BERT+word...
- 12:22 Lunch
- implement BERT+word...
- 14:00 Reading Group 4
- 15:00 debug BERT+topic architecture 3.5
- 16:30 experiment with less hidden layers 5
- Commute
- 18:28 Dinner
- 19:45

Wed 13

- 07:40 Get results 3.5
- 08:30 Get ready
- 09:57 train
- Cod...
- Anal...
- NiNo
- 12:30 train
- Cod...
- Lunch
- Commute
- Lunch
- 14:19 Try to train model on (shuffled) subset o...
- 15:45 Write experiment for (shuffled) Quora subset 4
- git commit 4
- update START profile 3
- Commute
- 18:46 Dinner
- 20:00

Thu 14

- 07:30 Get ready
- 08:15 Get results from VM 5
- 09:15 Prepare for meeting 5
- 10:15 Skype 4
- topic baseline for Seme...
- run BERT on Semeval A...
- Lunch
- 12:53 fix BERT preprocessing for semeval A and C (different len of sent...
- 14:15 run BERT on Semeval A...
- 15:45 emails 3.5
- Send reading group not...
- Plan schedule 3.5
- 18:00 sideproject 5
- 18:45 Dinner

Fri 15

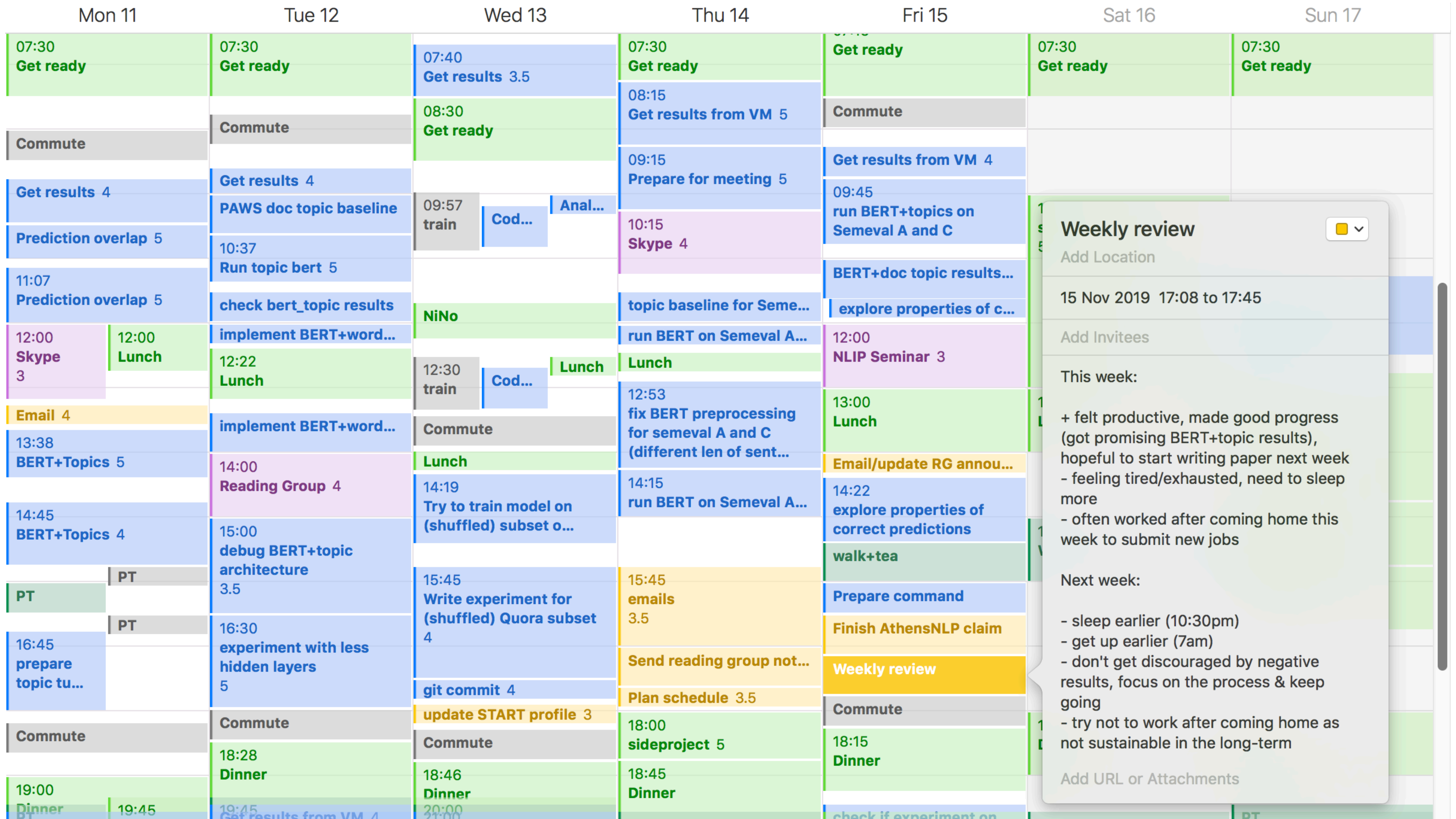
- Get ready
- Commute
- Get results from VM 4
- 09:45 run BERT+topics on Semeval A and C
- BERT+doc topic results...
- explore properties of c...
- 12:00 NLIP Seminar 3
- 13:00 Lunch
- Email/update RG annou...
- 14:22 explore properties of correct predictions
- walk+tea
- Prepare command
- Finish AthensNLP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

Sat 16

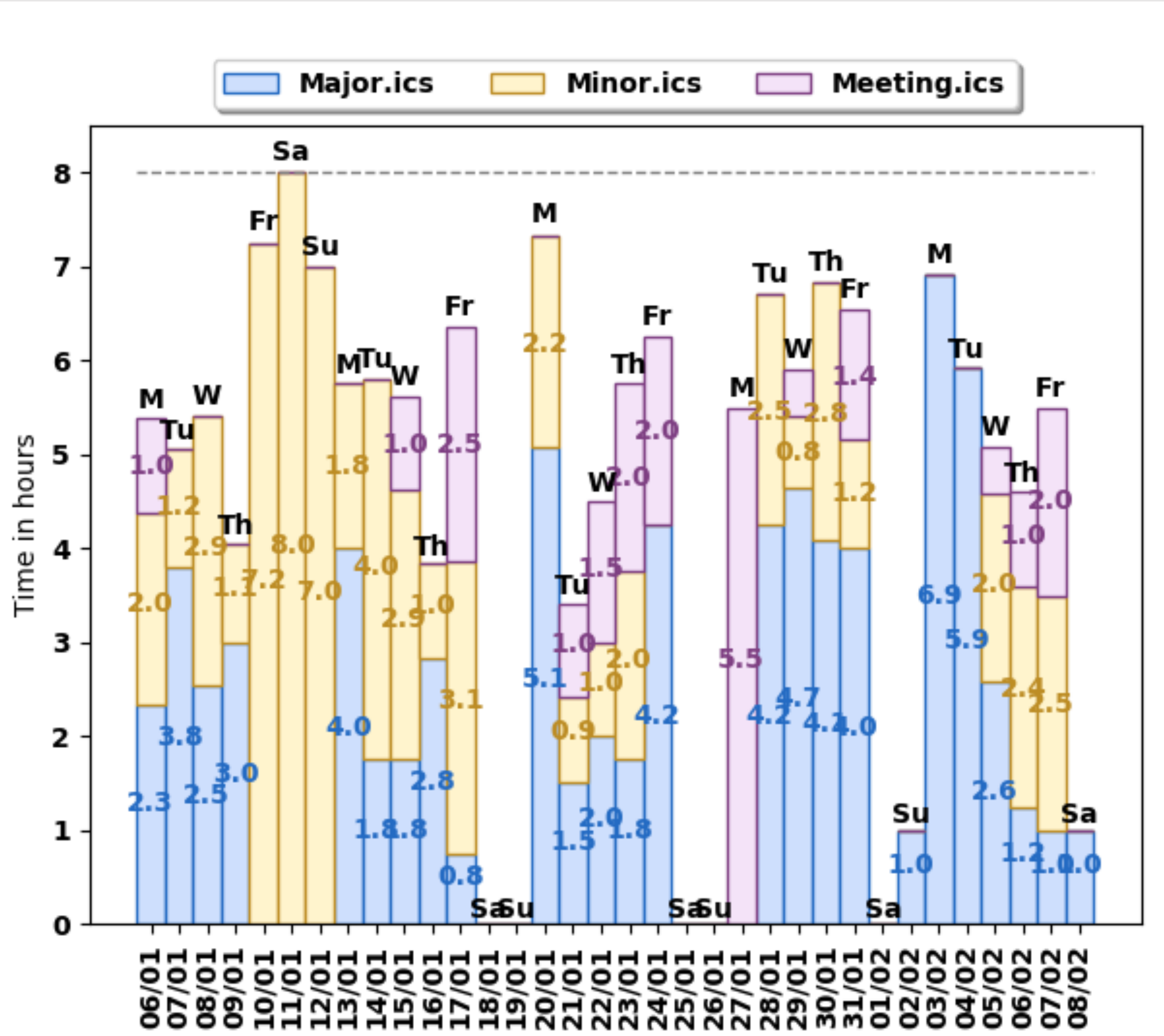
- 07:30 Get ready
- 10:00 sideproject 5
- 13:00 Lunch
- 15:00 Walk
- 18:00 Dinner

Sun 17

- 07:30 Get ready
- 11:15 Get results, balance GPU load
- 12:45 sideproject 4
- 14:45 Lunch
- 15:45 sideproject 3.5
- 18:00 Dinner

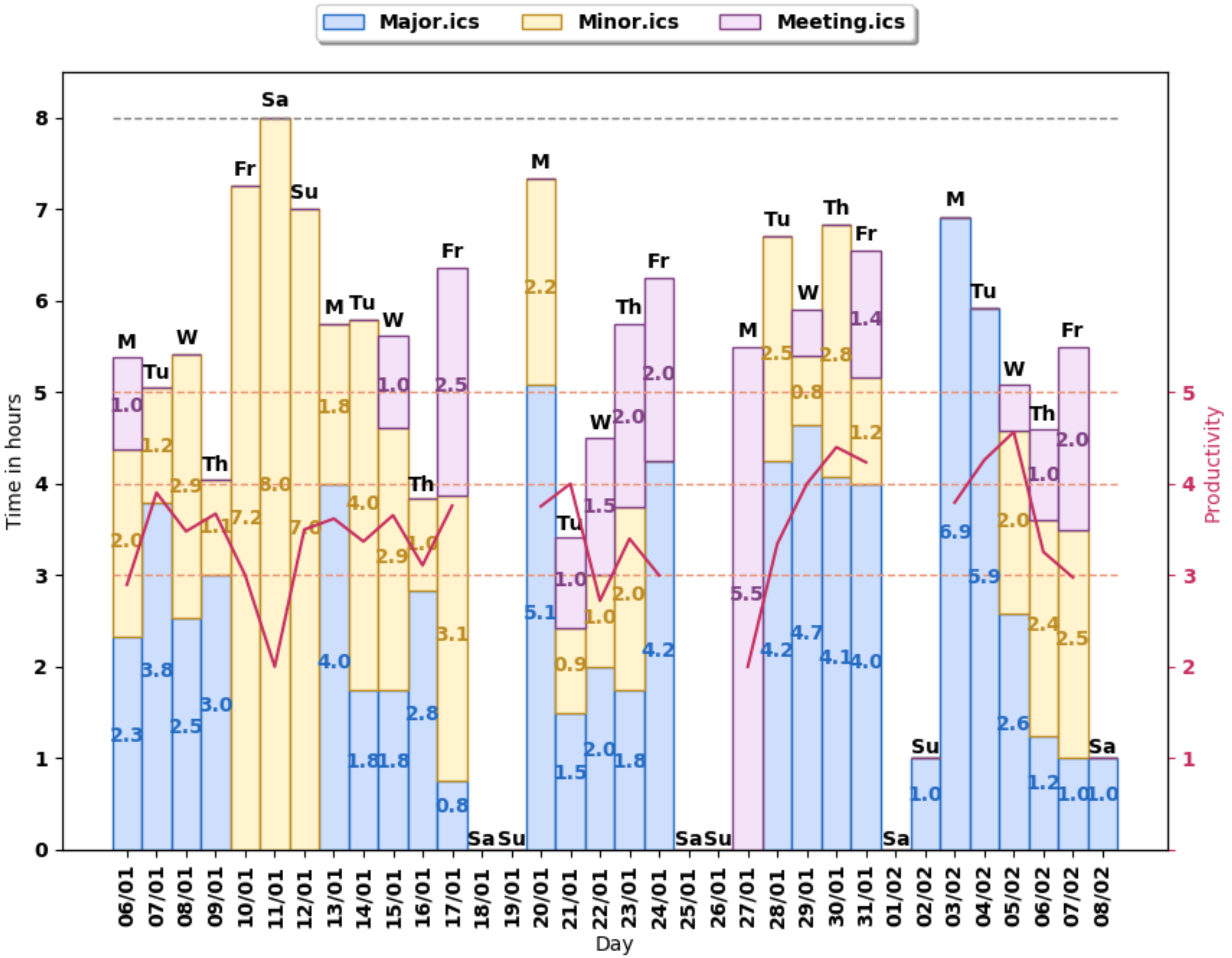

Mon 11
Tue 12
Wed 13
Thu 14
Fri 15
Sat 16
Sun 17
07:30 Get ready
Commute
Get results 4
Prediction overlap 5
11:07 Prediction overlap 5
12:00 Skype 3
12:00 Lunch
Email 4
13:38 BERT+Topics 5
14:45 BERT+Topics 4
PT
PT
PT
16:45 prepare topic tu...
Commute
19:00 Dinner
19:45
07:30 Get ready
Commute
Get results 4
PAWS doc topic baseline
10:37 Run topic bert 5
check bert_topic results
implement BERT+word...
12:22 Lunch
implement BERT+word...
14:00 Reading Group 4
15:00 debug BERT+topic architecture 3.5
16:30 experiment with less hidden layers 5
Commute
18:28 Dinner
19:45 Get results from VM 4
07:40 Get results 3.5
08:30 Get ready
09:57 train
Cod...
Anal...
NiNo
12:30 train
Cod...
Lunch
Commute
Lunch
14:19 Try to train model on (shuffled) subset o...
15:45 Write experiment for (shuffled) Quora subset 4
git commit 4
update START profile 3
Commute
18:46 Dinner
20:00
07:30 Get ready
08:15 Get results from VM 5
09:15 Prepare for meeting 5
10:15 Skype 4
topic baseline for Seme...
run BERT on Semeval A...
Lunch
12:53 fix BERT preprocessing for semeval A and C (different len of sent...
14:15 run BERT on Semeval A...
15:45 emails 3.5
Send reading group not...
Plan schedule 3.5
18:00 sideproject 5
18:45 Dinner
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
07:30 Get ready
07:30 Get ready
PT
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

Quantity-based calendar data summary



Code available at:
https://github.com/wuningxi/Timetrack

Quality-based calendar data summary



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 through a weekly review session and the Timetrack visualisation
- Why:
  - Holding yourself accountable
  - Allocating limited resources (energy & time) based on a task’s importance
  - Illustrating that every day makes a difference
  - Finding out what works for you & adjust

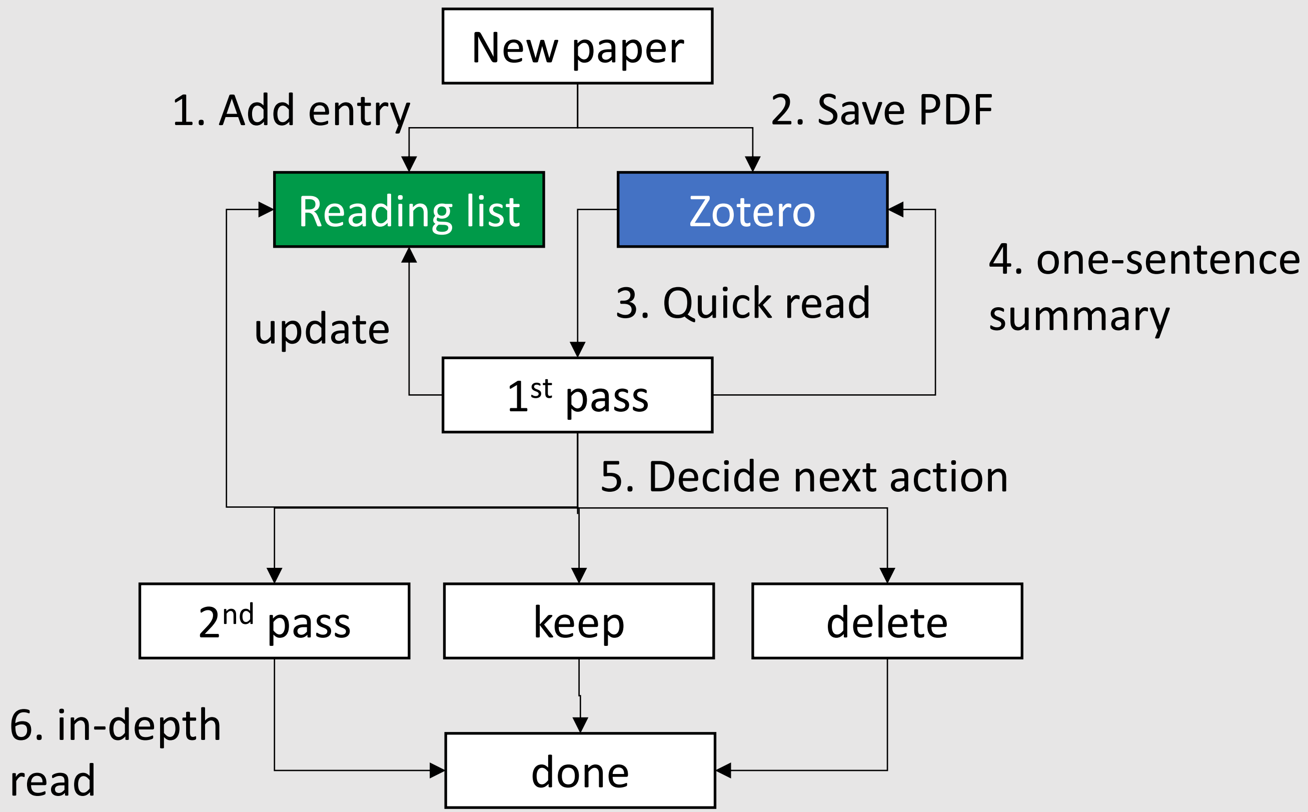
2. Making the most of weekly meetings

- 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

Semantic Simiarity Detection papers

Why read: keep up to date with what is happening in CQA, paraphrase detection and non-factoid question answering

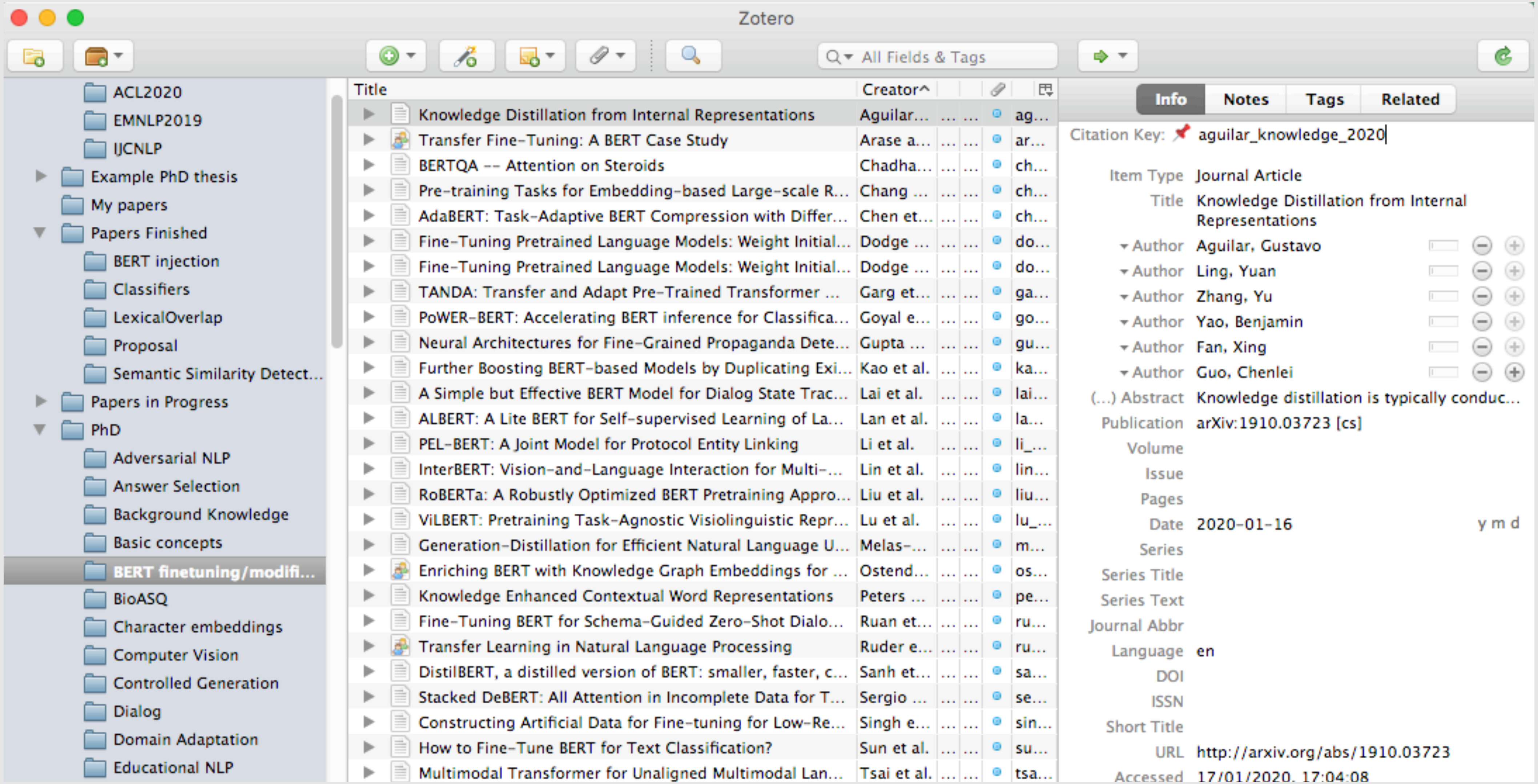
When	Title	Link	topic	first pass	second pass	Zotero
07.01.2020	Can AI Generate Love Advice?: Toward Neural Answer Generation f	https://arxiv.org/abs/1912.10163	CQA/non-factoid QA	n/a		
07.01.2020	Essential Sentences for Navigating Stack Overflow Answers	https://arxiv.org/abs/1912.13455	CQA/non-factoid QA	n/a		
07.01.2020	Knowledge-Enhanced Attentive Learning for Answer Selection in Co	https://arxiv.org/abs/1912.07915	CQA/non-factoid QA	n/a		
07.01.2020	A Multi-cascaded Model with Data Augmentation for Enhanced Par	https://arxiv.org/abs/1912.12068	paraphrase detection	n/a		
07.01.2020	CoSimLex: A Resource for Evaluating Graded Word Similarity in Con	https://arxiv.org/abs/1912.05320	other semantic similarity	n/a		
07.01.2020	Semantic Similarity To Improve Question Understanding in a Virtua	https://arxiv.org/abs/1912.07421	other semantic similarity	n/a		
20.01.2020	Building Interactive Sentence-aware Representation based on Gene	https://www.sciencedirect.com/sci	CQA	done	waiting	wu_building_2020
22.01.2020	Multi-level Head-wise Match and Aggregation in Transformer for T	https://arxiv.org/abs/2001.07234	paraphrase detection	done		wang_multi-level_2020
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
30.01.2020	What's happened in MOOC Posts Analysis, Knowledge Tracing and I	https://arxiv.org/pdf/2001.09830.pc	MOOC	waiting		ravikiran_whats_2020
31.01.2020	TANDA: Transfer and Adapt Pre-Trained Transformer Models for An	https://arxiv.org/abs/1911.04118	answer selection; bert finetunin	done		garg_tanda_2019

BERT modification papers

Why read: Check how papers have enhanced BERT, potentially useful for early topic fusion in BERT

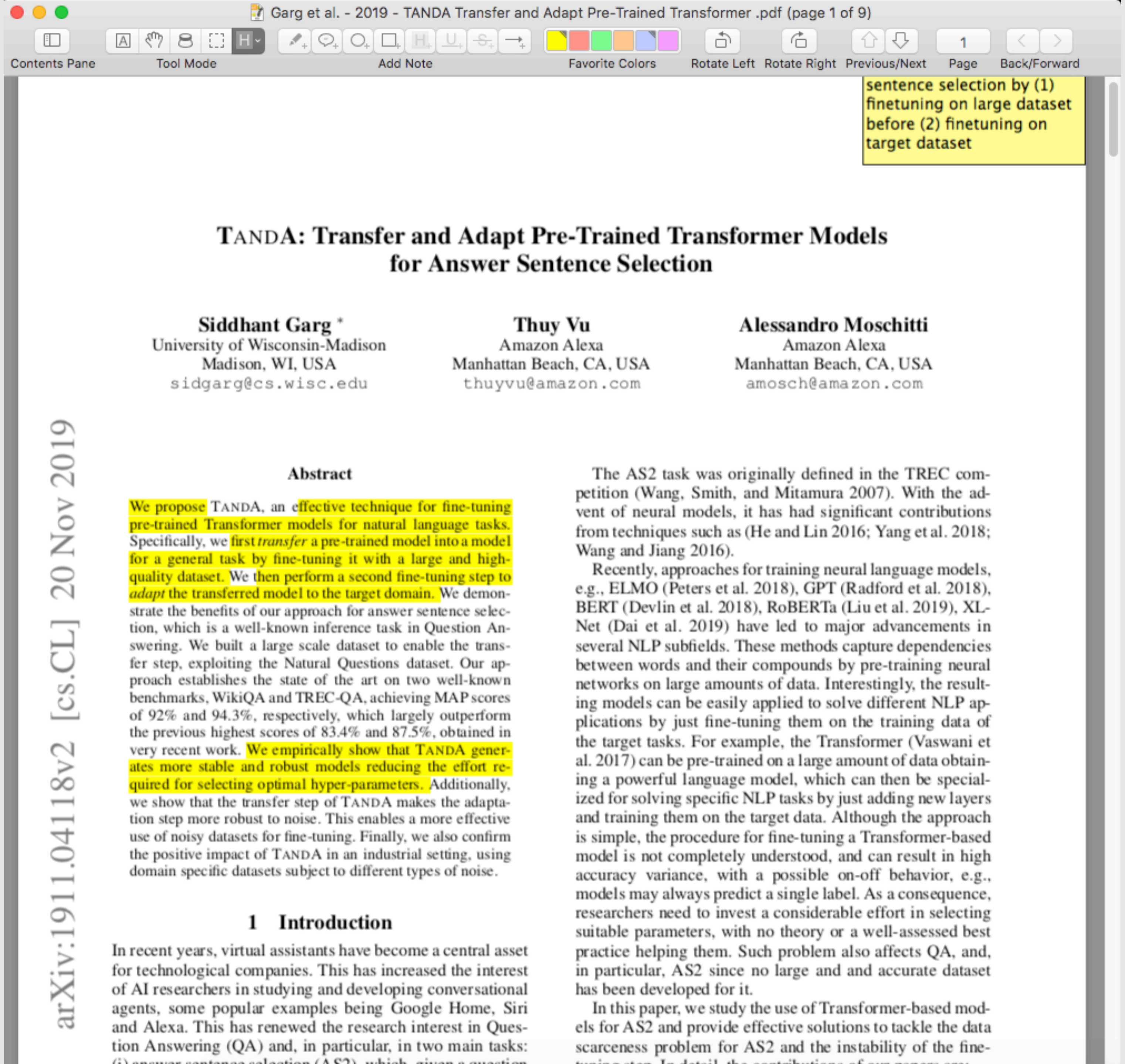
When	Title	Link	Topic	first pass	second pass	Zotero
08.01.2020	Semantics-aware BERT for Language Understanding	https://arxiv.org/abs/1909.02209v2	external BERT modification	done		zhang_semantics-aware_2020
07.01.2020	Stacked DeBERT: All Attention in Incomplete Data for Text Classific	https://arxiv.org/abs/2001.00137	internal BERT modification	waiting		
12.11.2019	**Knowledge Enhanced Contextual Word Representations**	**https://www.aclweb.org/anthology**	**internal BERT modification**	**waiting**		**peters_knowledge_2019**
14.01.2020	Unsupervised Domain Adaptation on Reading Comprehension	https://arxiv.org/abs/1911.06137	domain adaptation	n/a		
07.01.2020	BERTQA -- Attention on Steroids	https://arxiv.org/abs/1912.10435	external BERT modification	n/a		
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
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
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
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
14.01.2020	AdaBERT: Task-Adaptive BERT Compression with Differentiable Ne	https://arxiv.org/abs/2001.04246	lightweight	done		chen_adabert_2020
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
17.01.2019	Knowledge Distillation from Internal Representations	https://arxiv.org/abs/1910.03723	lightweight	done		aguilar_knowledge_2020
17.01.2019	DistilBERT, a distilled version of BERT: smaller, faster, cheaper and	https://arxiv.org/abs/1910.01108	lightweight	done		sanh_distilbert_2019
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)



1st pass

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



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:

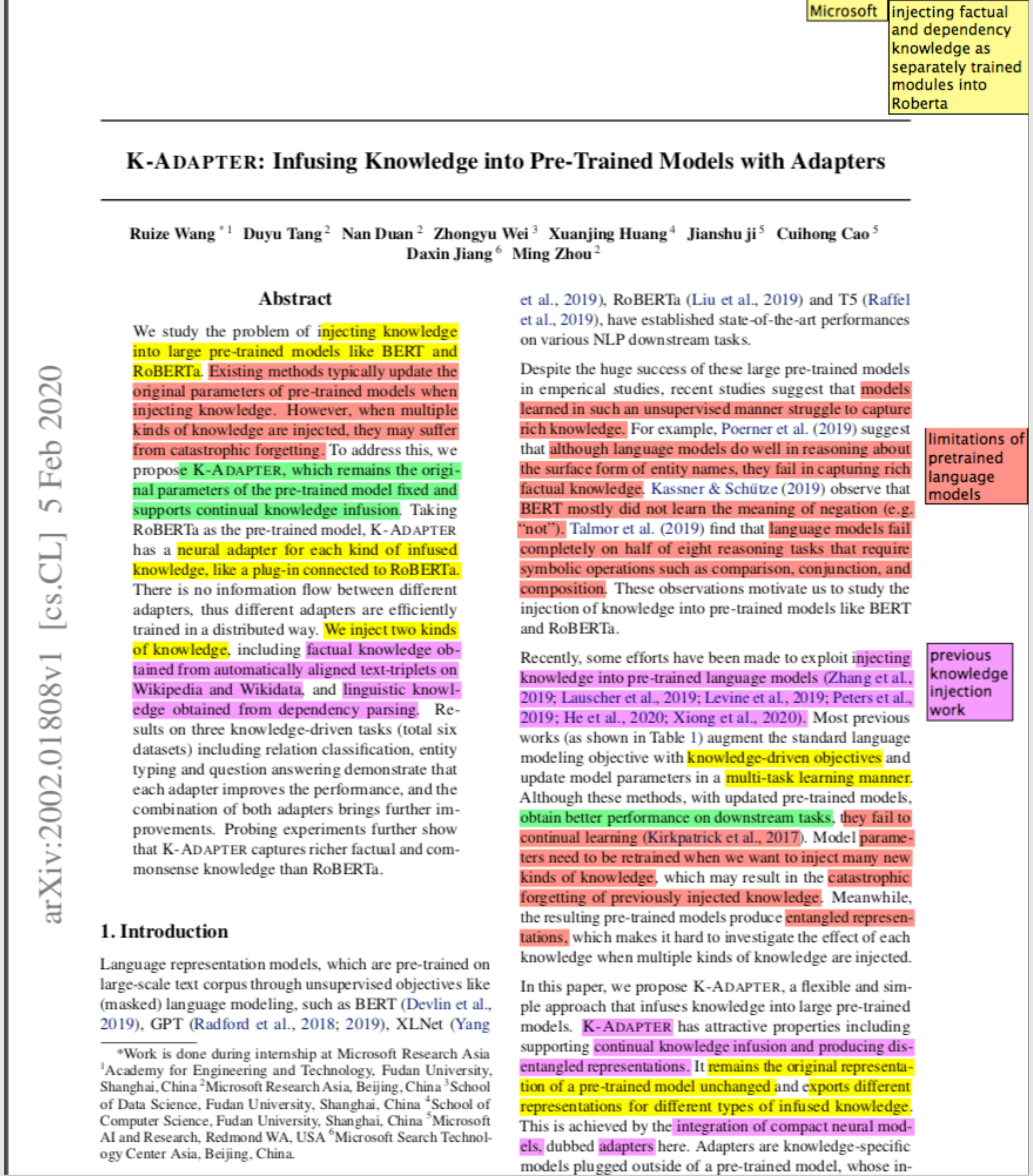
The AS2 task was originally defined in the TREC competition (Wang, Smith, and Mitamura 2007). With the advent of neural models, it has had significant contributions from techniques such as (He and Lin 2016; Yang et al. 2018; Wang and Jiang 2016).

Recently, approaches for training neural language models, e.g., ELMO (Peters et al. 2018), GPT (Radford et al. 2018), BERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019), XL-Net (Dai et al. 2019) have led to major advancements in several NLP subfields. These methods capture dependencies between words and their compounds by pre-training neural networks on large amounts of data. Interestingly, the resulting models can be easily applied to solve different NLP applications by just fine-tuning them on the training data of the target tasks. For example, the Transformer (Vaswani et al. 2017) can be pre-trained on a large amount of data obtaining a powerful language model, which can then be specialized for solving specific NLP tasks by just adding new layers and training them on the target data. Although the approach is simple, the procedure for fine-tuning a Transformer-based model is not completely understood, and can result in high accuracy variance, with a possible on-off behavior, e.g., models may always predict a single label. As a consequence, researchers need to invest a considerable effort in selecting suitable parameters, with no theory or a well-assessed best practice helping them. Such problem also affects QA, and, in particular, AS2 since no large and and accurate dataset has been developed for it.

In this paper, we study the use of Transformer-based models for AS2 and provide effective solutions to tackle the data scarceness problem for AS2 and the instability of the fine-

2nd pass

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



Microsoft

injecting factual and dependency knowledge as separately trained modules into Roberta

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] Daxin 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 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 emperical 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.

limitations of pretrained language models

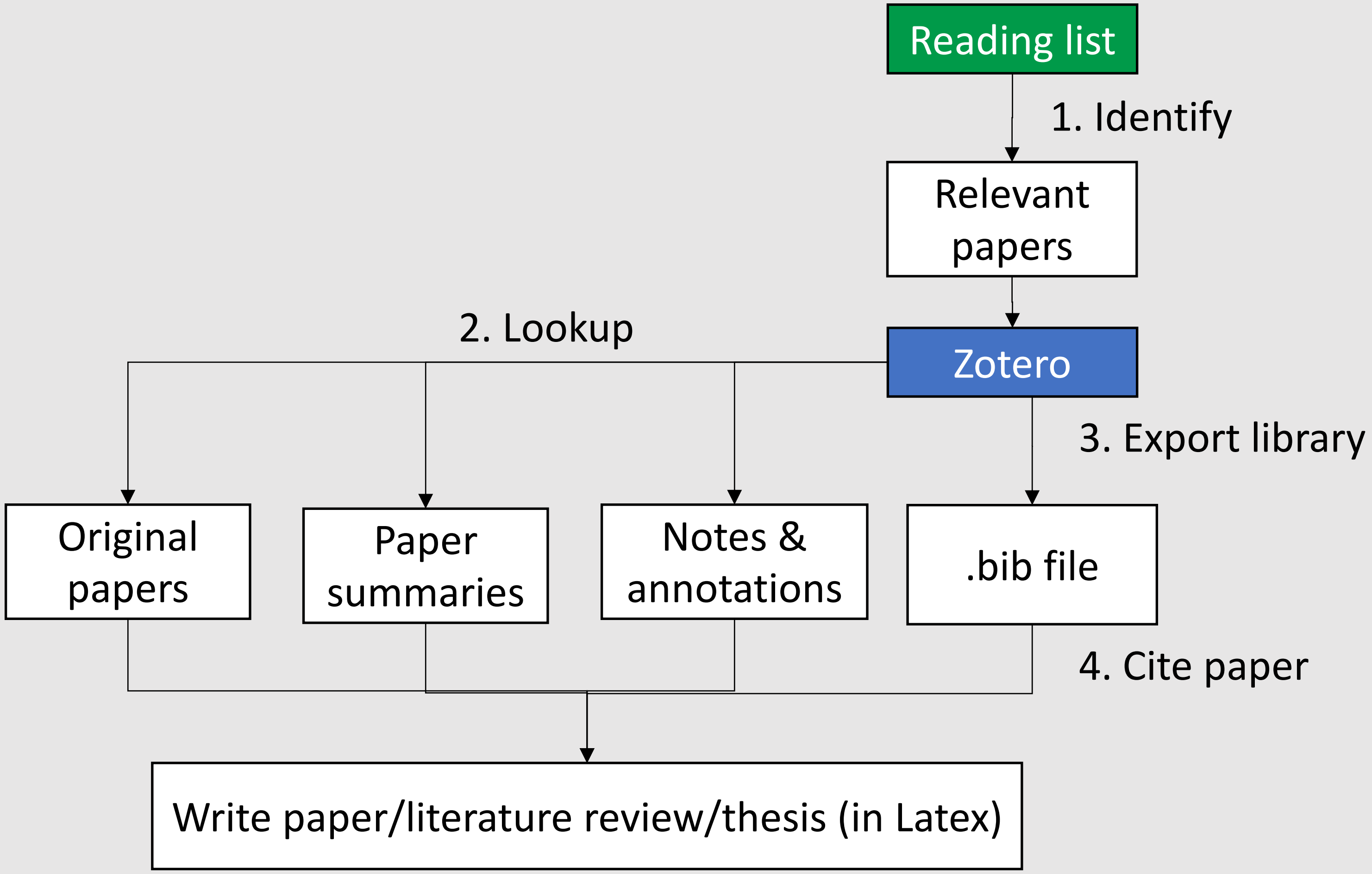
Recently, some efforts have been made to exploit injecting knowledge into pre-trained language models (Zhang et al., 2019; Lauscher et al., 2019; Levine et al., 2019; Peters et al., 2019; He et al., 2020; Xiong et al., 2020). Most previous works (as shown in Table 1) augment the standard language modeling 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.

previous knowledge injection work

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-

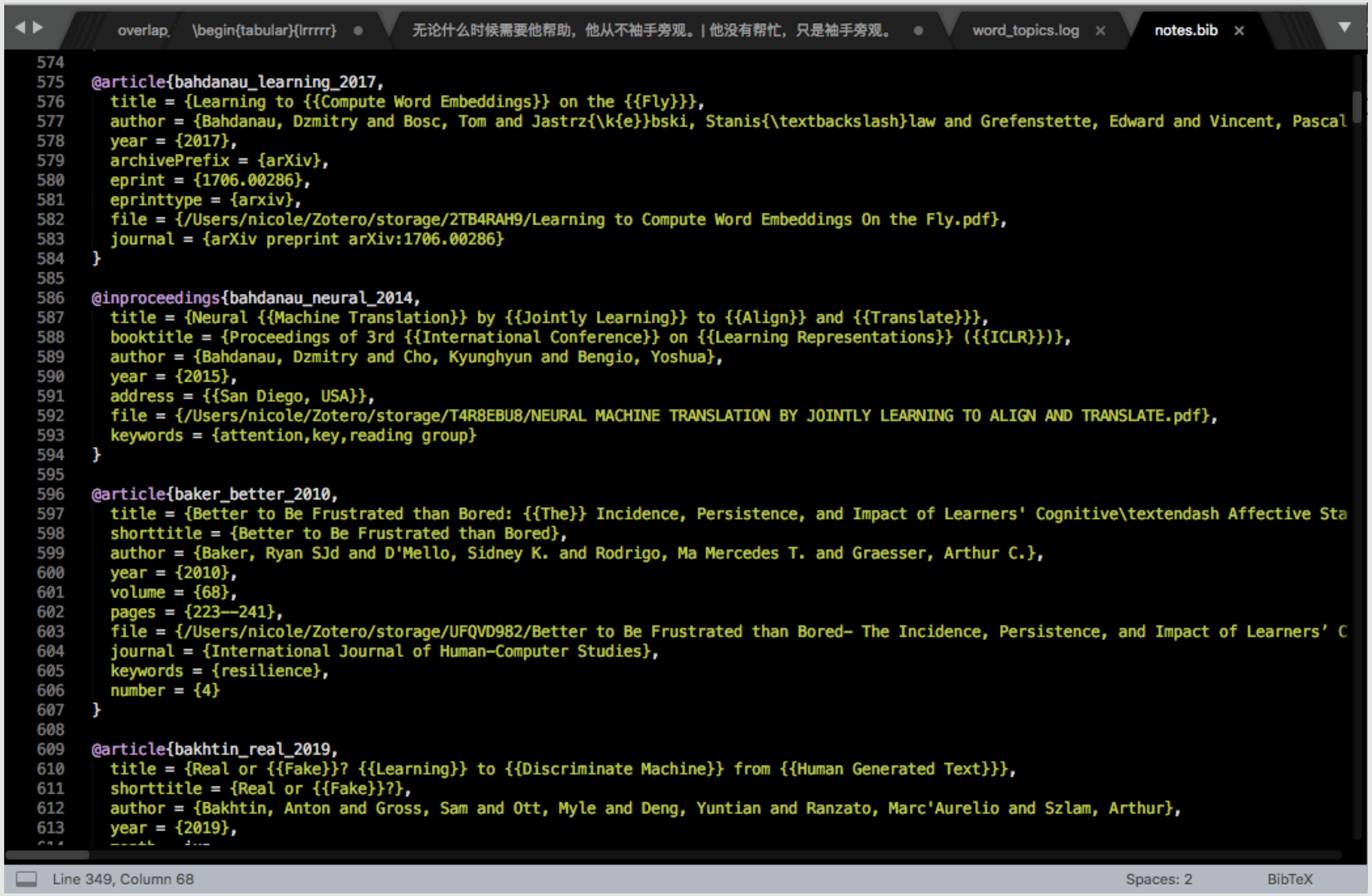
*Work is done during internship at Microsoft Research Asia [1]Academy for Engineering and Technology, Fudan University, Shanghai, China [2]Microsoft Research Asia, Beijing, China [3]School of Data Science, Fudan University, Shanghai, China [4]School of Computer Science, Fudan University, Shanghai, China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

3. Literature management: Writing

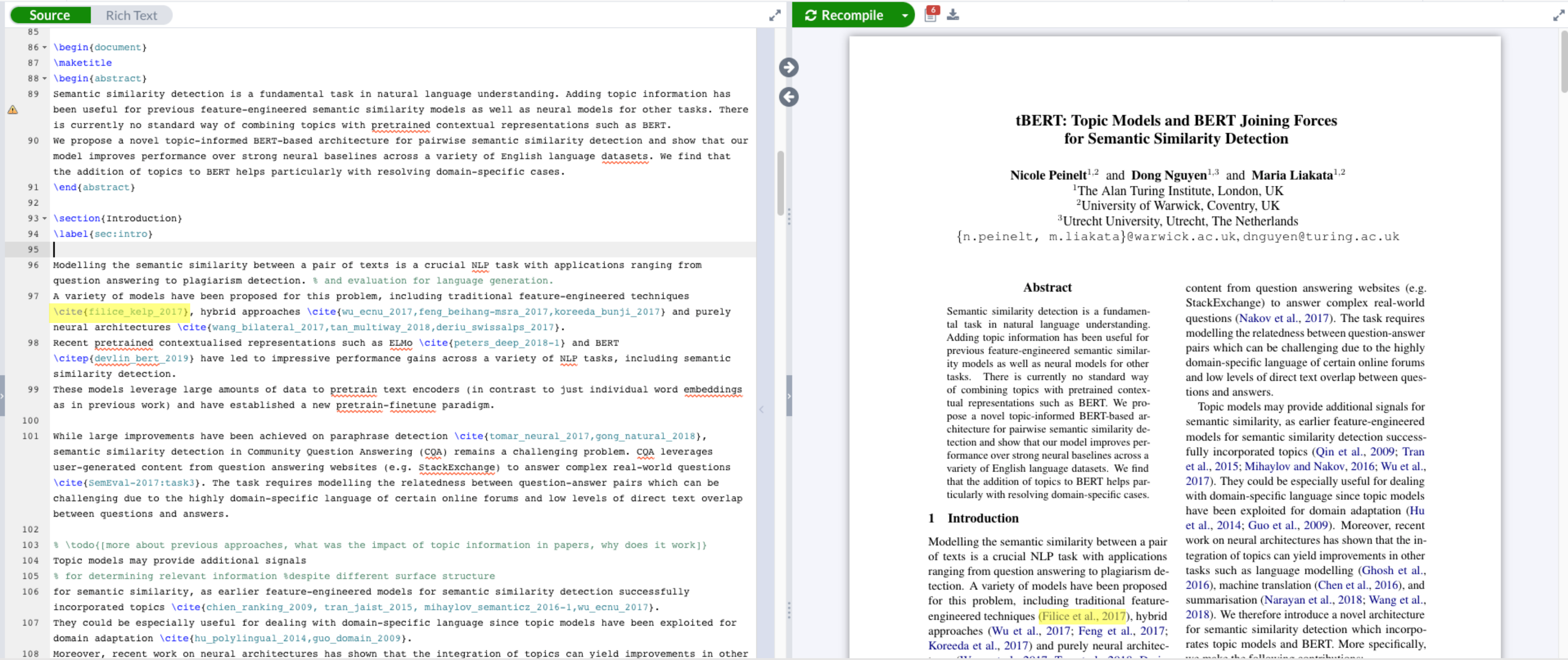


Export BibTex file for all references

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



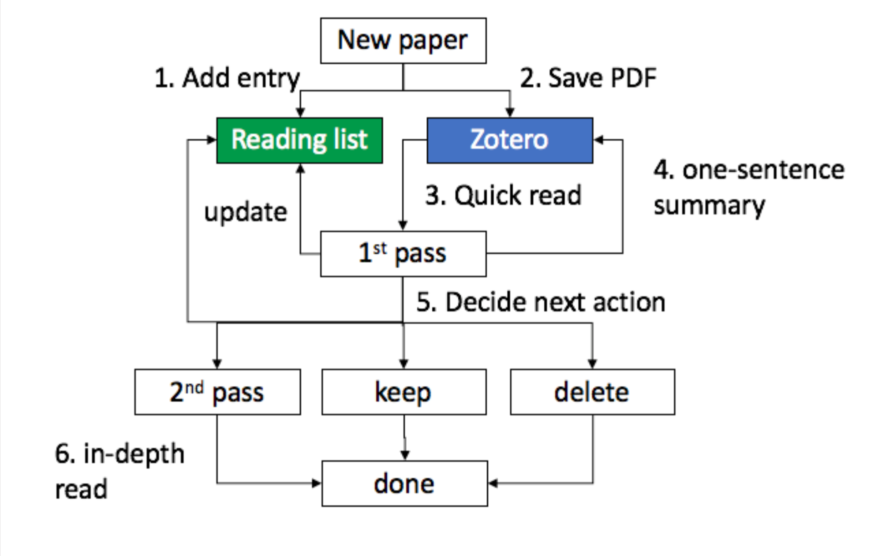
Write paper in Latex → \cite{zotero_key}



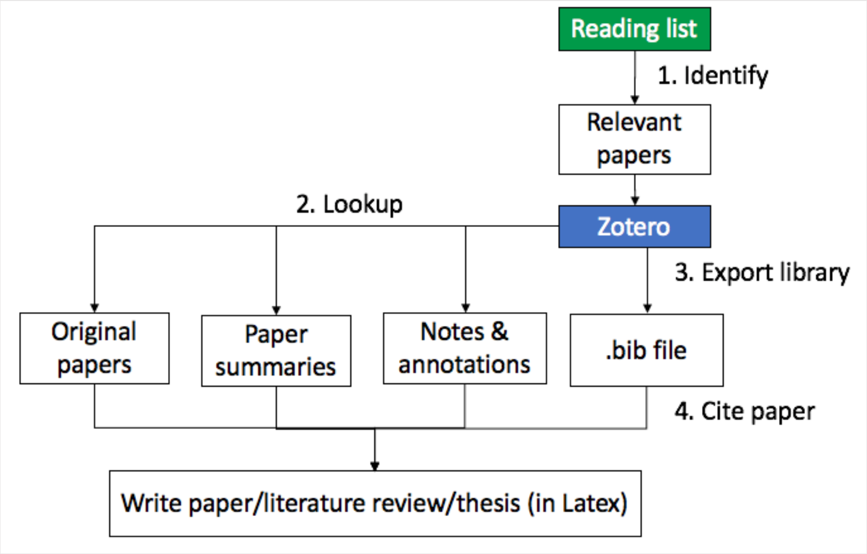
Literature management

- How:

Reading workflow



Writing workflow



- 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