How to read NLP papers

CQU CS 1701

NLP Group

- Search papers and group them
- Select the better paper
- The reading order you should follow
- Write down the notes
- Make a presentation

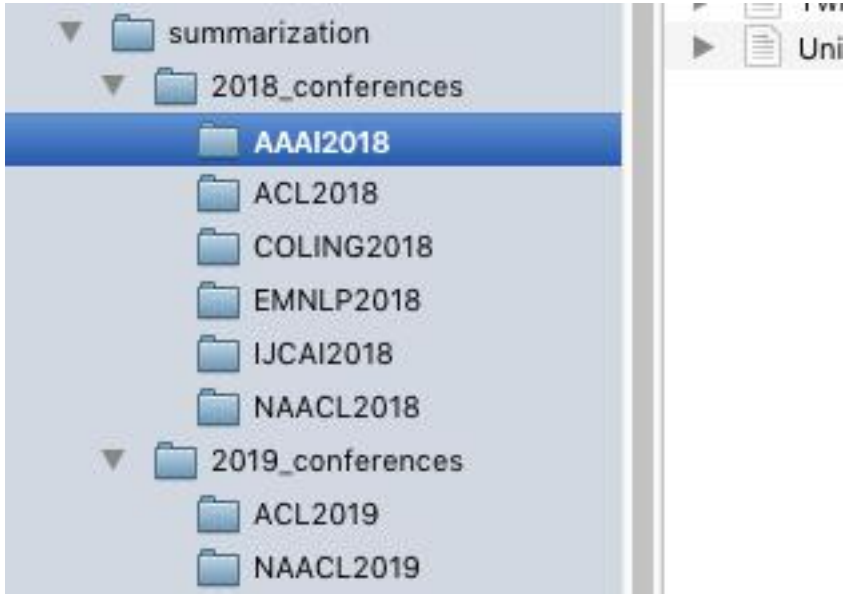
Search papers and group them

- By conferences
- By preprint or not
- By problems
- By methods(models)
- By dataset(text type)
- By optimize methods(depends on your own idea)

By conferences

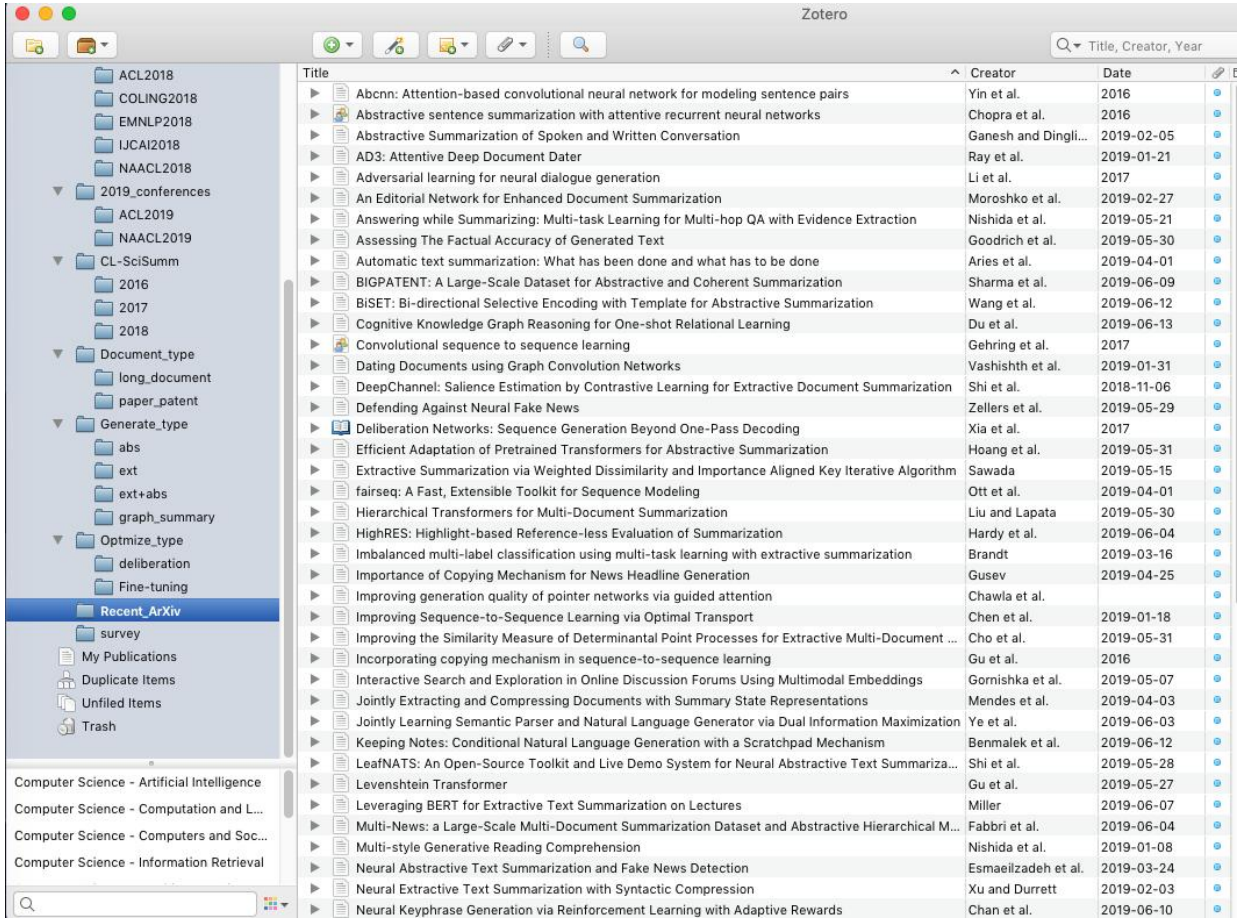
ACL Events

Venue	Present – 2010									
ACL	19	18	17	16	15	14	13	12	11	10
ANLP										
CL	19	18	17	16	15	14	13	12	11	10
CoNLL		18	17	16	15	14	13	12	11	10
EACL			17			14		12		
EMNLP		18	17	16	15	14	13	12	11	10
NAACL	19	18		16	15		13	12		10
*SEMEVAL	19	18	17	16	15	14	13	12		10



https://www.aclweb.org/anthology/

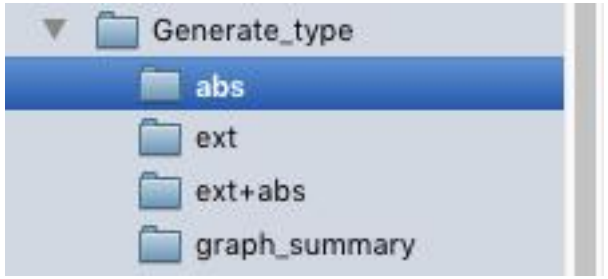
By preprint or not



https://arxiv.org/

By problems

- Summarization as example:
  - Abstractive
  - Extractive
  - Unsupervised
  - Graph based



By methods(models)

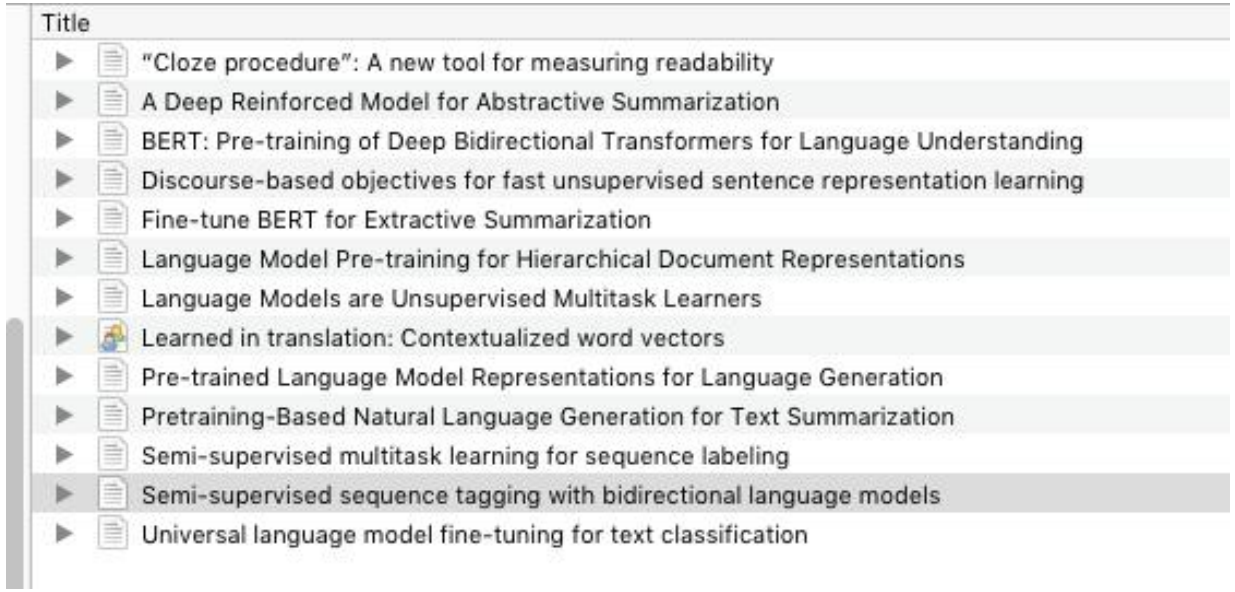
- CNN
- RNN
- GNN
- Transformer
- Attention
- Reinforcement

By dataset(text type)

- DUC
- LCSTS
- CNN/Daily Mail
- News
- Science /Academic papers
- Patent
- Health record

By optimize methods(depends on your own idea)

- Fine-tuning
- Deliberation
- Dual-learning



Select the better paper

How could you treat a paper as a good paper?

- Conferences
- Relation
- Citation
- Influence
- Code

Conferences

- ACL
- COLING
- EMNLP
- NAACL

NLP domain

- AAAI
- IJCAI
- NIPS
- ICLR2019

General AI

- NLPCC
- CCKS
- CCIR

Chinese NLP

Relation

- The most closely related paper

Example： UGC & PGC fusion

- Text Generation
  - Q&A
  - Essay writing
  - NMT
  - Summarization

Citation

Fast abstractive summarization with reinforce-selected sentence rewriting

YC Chen, M Bansal - arXiv preprint arXiv:1805.11080, 2018 - arxiv.org

Inspired by how humans summarize long documents, we propose an accurate and fast summarization model that first selects salient sentences and then rewrites them abstractively (ie, compresses and paraphrases) to generate a concise overall summary. We use a novel sentence-level policy gradient method to bridge the non-differentiable computation between these two neural networks in a hierarchical way, while maintaining language fluency. Empirically, we achieve the new state-of-the-art on all metrics (including human evaluation) …

☆ 99 被引用次数：49 相关文章 所有 3 个版本 »

Extractive Summarization with SWAP-NET: Sentences and Words from Alternating Pointer Networks

A Jadhav, V Rajan - Proceedings of the 56th Annual Meeting of the …, 2018 - aclweb.org

We present a new neural sequence-to-sequence model for extractive summarization called SWAP-NET (Sentences and Words from Alternating Pointer Networks). Extractive summaries comprising a salient subset of input sentences, often also contain important key words. Guided by this principle, we design SWAP-NET that models the interaction of key words and salient sentences using a new two-level pointer network based architecture. SWAP-NET identifies both salient sentences and key words in an input document, and then …

☆ 99 被引用次数：4 相关文章 »

Influence

ACL2019 Summarization ACs:

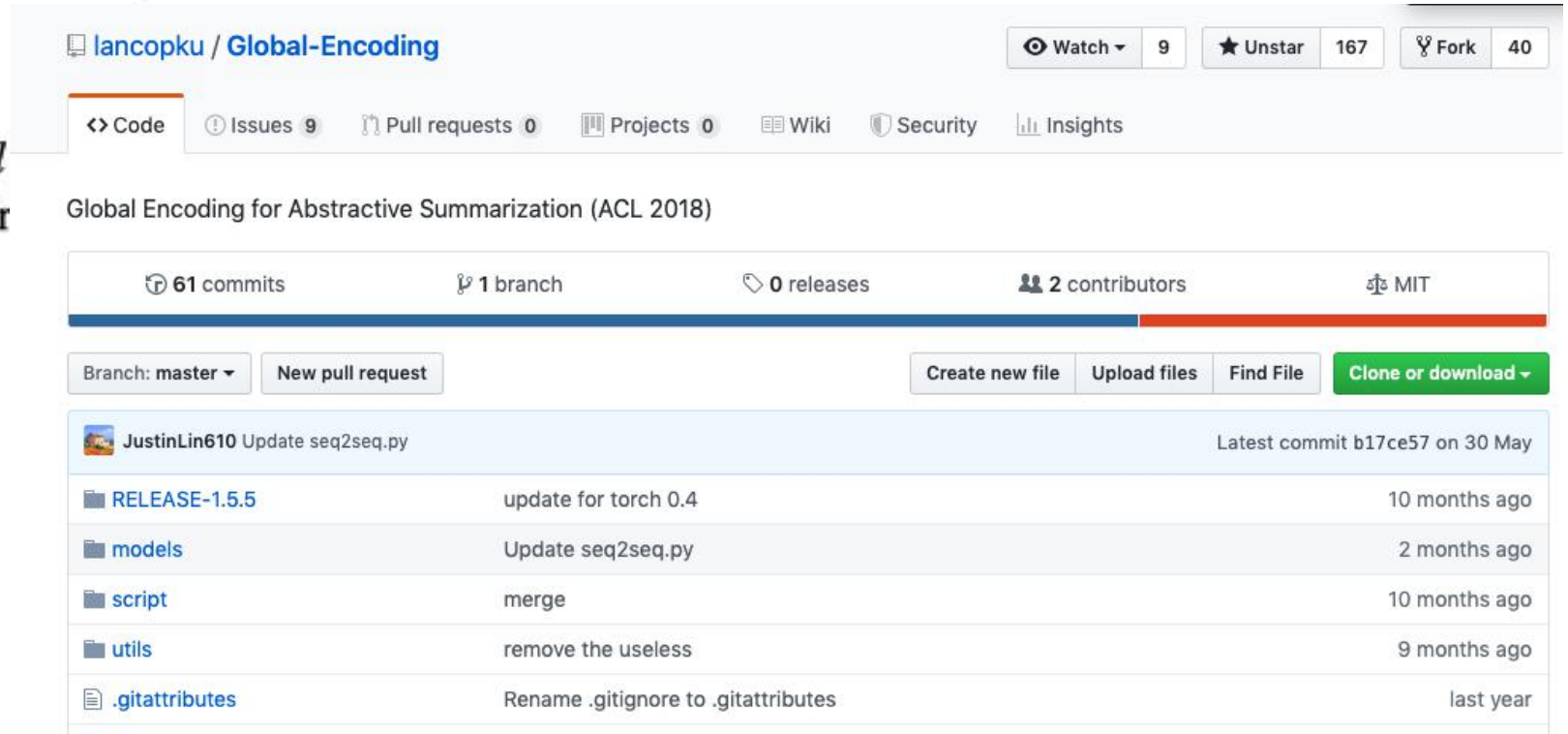
Position	Name	Website	Department
SAC	Mirella Lapata	http://homepages.inf.ed.ac.uk/mlap/index.php?page=index	Institute for Language, Cognition and Computation School of Informatics, University of Edinburgh
SAC	Chin-Yew Lin	https://www.microsoft.com/en-us/research/people/cyl/	research manager of the Knowledge Computing group at Microsoft Research Asia
AC	Wenjie Li	http://www4.comp.polyu.edu.hk/~cswjli/	Associate Professor, Department of Computing The Hong Kong Polytechnic University
AC	Xiaojun Wan	https://wanxiaojun.github.io/	北京大学计算机科学技术研究所语言计算与互联网挖掘研究室
AC	Jackie Chi Kit Cheung	https://www.cs.mcgill.ca/~jcheung/index.html	Reasoning and Learning Lab School of Computer Science, McGill University
AC	Shashi Narayan	http://homepages.inf.ed.ac.uk/snaraya2/index.html	School of Informatics The University of Edinburgh
AC	Xiaodan Zhu	http://www.xiaodanzhu.com/about.html	Department of Electrical and Computer Engineering at Queen's University
AC	Fei Liu	http://www.cs.ucf.edu/~feiliu/	UCF Natural Language Processing Group

Code

methods.

[1]The code is available at https://www.github.com/lancopku/Global-Encoding

Proceedings of the 56th Annual
Melbourne, Austr



The reading order you should follow

- Abstract
- Abstract + Introduction(the second half)
- Experiment + Conclusion
- Proposal

- Do not just watch the title!

Abstract

- Subarea
- Problems
- Proposal(mothods)
- Datasets
- Performance

Abstract

In neural abstractive summarization, the conventional sequence-to-sequence (seq2seq) model often suffers from repetition and semantic irrelevance. To tackle the problem, we propose a global encoding framework, which controls the information flow from the encoder to the decoder based on the global information of the source context. It consists of a convolutional gated unit to perform global encoding to improve the representations of the source-side information. Evaluations on the LCSTS and the English Gigaword both demonstrate that our model outperforms the baseline models, and the analysis shows that our model is capable of generating summary of higher quality and reducing repetition[1].

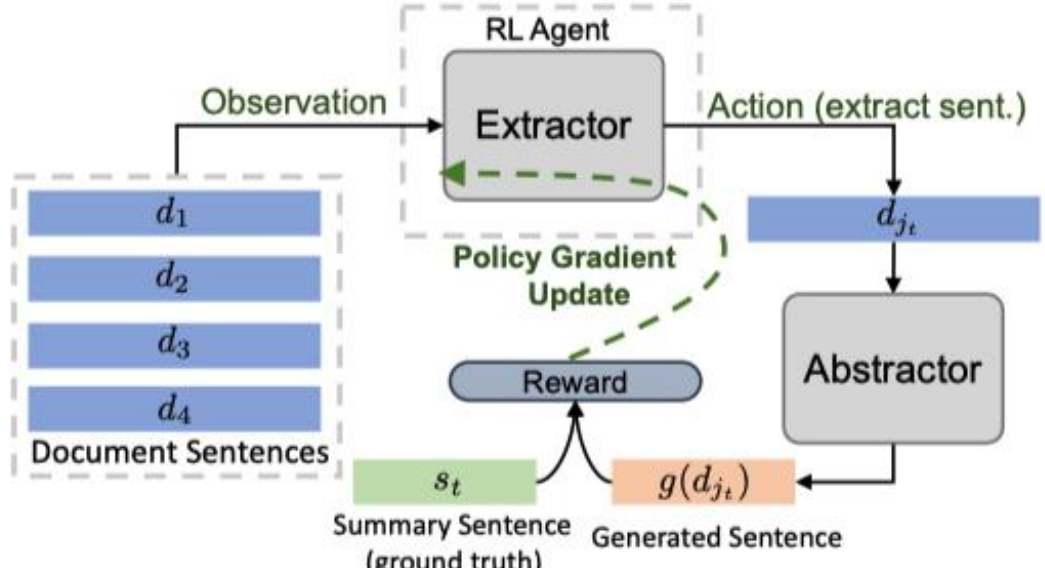
Abstract + Introduction(the second half)

- What if the abstract can not give enough information

Abstract

Inspired by how humans summarize long documents, we propose an accurate and fast summarization model that first selects salient sentences and then rewrites them abstractively (i.e., compresses and paraphrases) to generate a concise overall summary. We use a novel sentence-level policy gradient method to bridge the non-differentiable computation between these two neural networks in a hierarchical way, while maintaining language fluency. Empirically, we achieve the new state-of-the-art on all metrics (including human evaluation) on the CNN/Daily Mail dataset, as well as significantly higher abstractiveness scores. Moreover, by first operating at the sentence-level and then the word-level, we enable *parallel decoding* of our neural generative model that results in substantially faster (10-20x) inference speed as well as 4x faster training convergence than previous long-paragraph encoder-decoder models. We also demonstrate the generalization of our model on the test-only DUC-2002 dataset, where we achieve higher scores than a state-of-the-art model.

Thus, our method incorporates the abstractive paradigm's advantages of concisely rewriting sentences and generating novel words from the full vocabulary, yet it adopts intermediate extractive behavior to improve the overall model's quality, speed, and stability. Instead of encoding and attending to every word in the long input document sequentially, our model adopts a human-inspired *coarse-to-fine* approach that first extracts all the salient sentences and then decodes (rewrites) them (*in parallel*). This also avoids almost all redundancy issues because the model has already chosen non-redundant salient sentences to abstractively summarize (but adding an optional final reranker component does give additional gains by removing the fewer across-sentence repetitions).



Experiment + Conclusion

- Take care of the result

Model	R-1	R-2	R-L
RNN	21.5	8.9	18.6
RNN-context	29.9	17.4	27.2
CopyNet	34.4	21.6	31.3
SRB	33.3	20.0	30.1
DRGD	37.0	24.2	34.2
seq2seq (Our impl.)	33.8	23.1	32.5
+CGU	**39.4**	**26.9**	**36.5**

Table 2: **F-Score of ROUGE on LCSTS.**

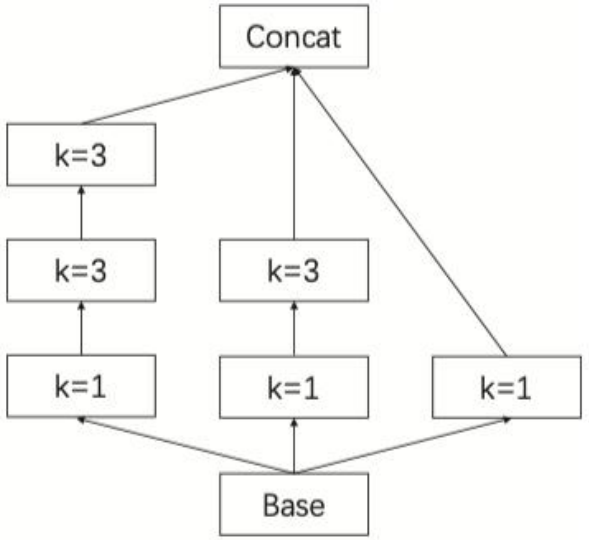
Model	R-1	R-2	R-L
ABS	29.6	11.3	26.4
ABS+	29.8	11.9	27.0
Feats	32.7	15.6	30.6
RAS-LSTM	32.6	14.7	30.0
RAS-Elman	33.8	16.0	31.2
SEASS	36.2	17.5	33.6
DRGD	**36.3**	17.6	33.6
seq2seq (Our impl.)	33.6	16.3	31.3
+CGU	**36.3**	**18.0**	**33.8**

Table 3: **F-Score of ROUGE on Gigaword.**

Proposal

- Intensive reading
- Focus on the parts below:
  - The most creative part
  - Figures(overview and detail)
  - Equations(How and Why)

2.2 Convolutional Gated Unit



Do not just watch the title!

- Bottom-Up Abstractive Summarization(Two-stage)
- Attention Is All You Need(Transformer)

Write down the notes

- From(Conferences/Journals /ArXiv)
- Institution
- Paper
- Topic
- Aim
- Problem to solve
- Solutions
- Strengths
- Limitations
- Datasets
- Evaluation scores
- Code

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	From	institution	Paper	topic	aim	problem to solve	solutions	strengths	limitations	Datasets	ROUGE-1	ROUGE-2	ROUGE-L
2	arXiv:1903.10318 [cs]	Institute for Language, Cognition and Computation, School of Informatics, University of Edinburgh	Fine-tune BERT for Extractive Summarization	Extractive Summarisation	All for the same purpose , improve the ROUGE score		a flat architecture with inter-sentence Transformer layers	transform the segmentation embedding to extractive summary choice	to be found	CNN/Dailymail	43.25	20.24	
3	arXiv:1903.09722 [cs]	Facebook AI Research	Pre-trained Language Model Representations for Language Generation	Abstractive summarisation		Previous work on integrating language models with sequence to sequence models focused on the decoder network and added language model representations right before the output of the decoder	integrate pre-trained representations as input to the encoder network	focus encoder step	to be found	CNN-DailyMail	41.56	18.94	
4	arXiv:1902.09243 [cs]	College of Computer, National University of Defense Technology Microsoft Research Asia	Pretraining-Based Natural Language Generation for Text Summarization	more like Abstractive summarisation		the decoder cannot utilize BERT' s ability to generate high quality context vectors	propose a new word-level refine decoder	two step generation, the main idea like Deliberation network	to be found	CNN/Daily Mail NYT50	41.71 45.33	19.49 26.53	
						1.initialize the							

Make a presentation

Grasp every opportunity you can seize !!!

- Group meetings
- Class presentations
- Other conferences of projects
- Submit papers

How to run opensource code

- The most significant thing——codes
- README.MD
- Issues
- Author's email

The most significant thing——codes

- Depends on the quality of paper and the integrity of authors.
- Github is the most popular way to open source their work(some times they only release them on their web portal)
- Know how to fork repositories

READ.MD

- Markdown file type
- Requirements
- Preprocessing
- Training
- Evaluation

- Different work environment(linux/windows)

Markdown file type

Requirements

Requirements

- Ubuntu 16.0.4
- Python 3.5
- Pytorch 0.4.1 (updated)
- pyrouge

In order to use pyrouge, set rouge path with the line below:

```
pyrouge_set_rouge_path RELEASE-1.5.5/
```

It seems that some user have met problems with pyrouge, so I have updated the script, and users can put the directory "RELEASE-1.5.5" in your home directory and set rouge path to it (or run the command "chmod 777 RELEASE-1.5.5" for the permission).

Preprocessing

Preprocessing

```
python3 preprocess.py -load_data path_to_data -save_data path_to_store_data
```

Remember to put the data into a folder and name them *train.src*, *train.tgt*, *valid.src*, *valid.tgt*, *test.src* and *test.tgt*, and make a new folder inside called *data*

Training

Training

```
python3 train.py -log log_name -config config_yaml -gpus id
```

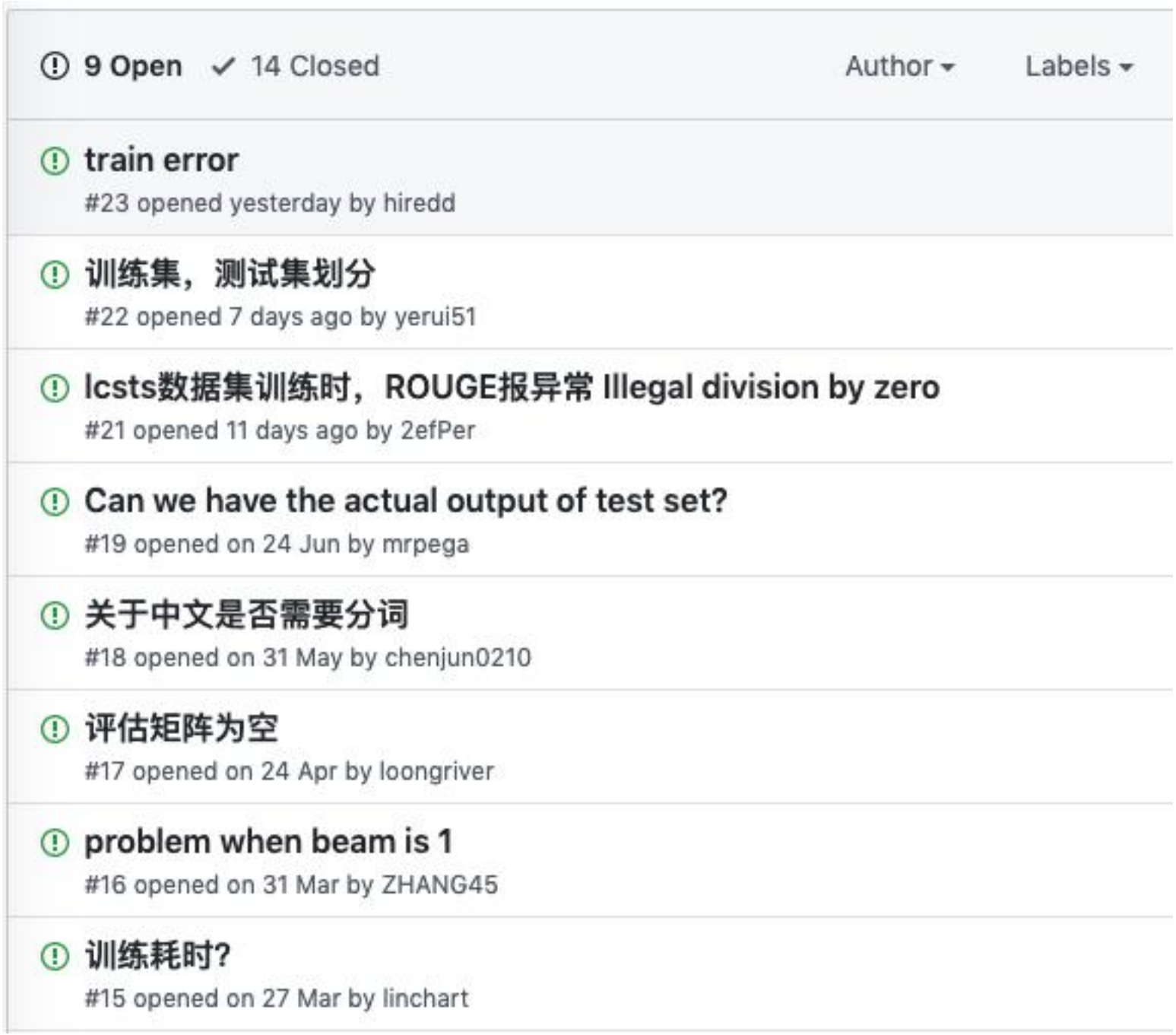
Create your own yaml file for hyperparameter setting.

Evaluation

Evaluation

```
python3 train.py -log log_name -config config_yaml -gpus id -restore checkpoint -mode eval
```

Issues


9 Open
14 Closed
Author
Labels
train error
#23 opened yesterday by hiredd
训练集，测试集划分
#22 opened 7 days ago by yerui51
lcsts数据集训练时，ROUGE报异常 Illegal division by zero
#21 opened 11 days ago by 2efPer
Can we have the actual output of test set?
#19 opened on 24 Jun by mrpega
关于中文是否需要分词
#18 opened on 31 May by chenjun0210
评估矩阵为空
#17 opened on 24 Apr by loongriver
problem when beam is 1
#16 opened on 31 Mar by ZHANG45
训练耗时?
#15 opened on 27 Mar by linchart

Author's email

- Paper mentioned
- Github shown
- Web portal shown
- Friends

Junyang Lin, Xu Sun, Shuming Ma, Qi Su
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
School of Foreign Languages, Peking University
{linjunyang, xusun, shumingma, sukia}@pku.edu.cn

Different work environment(linux/windows)

- Environment variables
- Shell/BAT scripts

```
export CORENLP_HOME=path/to/stanford_jars/stanford-corenlp-full-2018-10-05
```

To run the model, use the command,

```
./run_inference_on_doc.sh <lang> <infile> <outfile>
```

https://github.com/shyamupa/xling-el

Hard way

- Open .sh file and find the python calling command

```
python -m readers.xel_annotator \
       --kb_file data/mykbs/biggest.kb \
       --vocabpkl ${VOCABPKL} \
       --vecpkl ${VECPKL} \
       --ncands 20 \
       --usecoh \
       --cohstr ${COHPATH} \
       --test_doc ${infile} \
       --out_doc ${outfile} \
       --restore ${restore_path} \
       --lang ${lang}
```