EMNLP19 NIPS19 Paper Notes

Xiachong Feng

1 EMNLP19

1.1 Mask-Predict: Parallel Decoding of Conditional Masked Language Models

本篇工作是 Facebook AI(西雅图) 的一篇工作，这篇工作的思路第一眼看有点像 [Zhang et al., 2019]，目的也是为了将 BERT 利用到解码端，所以先生成 Summary 的草稿，然后再 Refine。和这篇论文提出的 Mask-predict 核心想法一致。这篇工作属于 Non-Autoregressive 一大类，今年 NIPS19[Gu et al., 2019] 是我看到的非自回归解码中最硬核的一篇，当然 ICLR18[Gu et al., 2017]，在机器翻译上也是 Gu 首次提出了非自回归的方法，牺牲了一定的效果，但是在解码端可以并行加速。我觉得不仅仅是并行加速的问题，这种不需要从左到右的解码方式为很多知识、很多结构的融入，或者说是很多先验的融入提供了机会。这篇工作基于 Encoder-Decoder 框架实现了条件 Mask Language Model(CMLM)，在 Decoder 端可以预测任意位置的词语，Encoder 编码一句话，并且在 Encoder 端拼接特殊 [LENGTH] 来预测目标序列的长度，在 Decoder 端，最开始全部 Mask，然后全部预测，然后选取概率低的 n 个词语，继续进行 Mask 来预测，迭代 T 次，这个 T 可以自己设置，或者是目标序列长度的一个函数。每次选择的 n 个 Mask 的词语是线性衰减的。损失包括了每一个 Mask 位置和 [LENGTH] 的预测。在生成的时候，首先通过 Encoder 端的 [LENGTH] 来预测目标序列长度，选取 Top k 个可能的长度进行生成，最后选择概率最高的一个序列。最后实验发现超过了之前的非自回归的方法。一个实际的例子如1，最开始全部 mask，然后生成了 t=0 句话。选择黄色部分 Mask，进行 Mask-Predict。最后三轮以后发现得到最终效果不错。

src	Der Abzug der franzsischen Kampftruppen wurde am 20. November abgeschlossen .
$t = 0$	The departure of the French combat completed completed on 20 November .
$t = 1$	The departure of French combat troops was completed on 20 November .
$t = 2$	The withdrawal of French combat troops was completed on November 20th .

图 1: Mask Predict

1.2 NCLS: Neural Cross-Lingual Summarization

来自于中科院自动化所模式识别国家重点实验室。针对跨语言摘要（Cross-Lingual Summarization）提出了新的数据集（准确的说是第一份数据集，ACL19[Duan et al., 2019] 利用 Gigaword 只标注了开发集和测试集）。基于该数据集完成端到端的训练，并利用单语言摘要（MS，Monolingual Summarization）和机器翻译（MT，Machine Translation）作为辅助任务进行多任务学习。针对的问题还是由于之前没有直接的数据，所以需要基于”摘要“和”翻译“的 pipeline 完成，这会导致

错误传播。有了新的数据集以后，就可以来完成端到端的训练了。某种意义上类似于机器翻译，但是机器翻译是一一对应，相当于压缩比是 1:1，但是跨语言摘要还需要完成重要内容提取，然后在目标端重新组织语言。

跨语言摘要简单来说就是输入源语言文档，输出目标语言摘要。源语言与目标语言不同。之前的方法是基于 pipeline 的方法。先摘要后翻译：这种方法首先需要在源语言端有摘要的数据来训练一个源语言的摘要模型，对于 Low-resource 来说，就比较麻烦。或者可以使用一些类似 TextRank 之类的无监督方法实现。如果摘要这一步就比较差，再加上翻译效果不好，那最后的结果可想而知。另一种是先翻译再摘要：翻译主要有 domain 的问题，训练翻译模型的 domain 和摘要的 domain 不同，那第一步翻译结果就不行，然后再摘要最后效果也会一般。所以整体来说，pipeline 方法会有严重的错误级联问题，一直被诟病。[Ouyang et al., 2019] 觉得先摘要的方法对于 Low-resource 来说行不通，所以更加 prefer 先翻译的方法。那么解决 pipeline 方法的根本方式就是有一个端到端的数据集，这篇论文就干了这样一件事情。

数据集由英语单语数据集 CNNDM 和多模态摘要数据集 MSMO[Zhu et al., 2018] 和中文单语数据集 LCSTS[Hu et al., 2015] 得到。其中 CNNDM 和 MSMO 统称 ENSUM，属于新闻领域，LCSTS 数据来源于新浪微博。基于 ENSUM 来构建 En2Zh，基于 LCSTS 来构建 Zh2EN。采用的方法叫做 Round-trip translation strategy，round-trip 翻译就是首先把一个句子翻译到另外一种语言（forward translation），然后再翻译回来（back translation）。基本过程如图2。

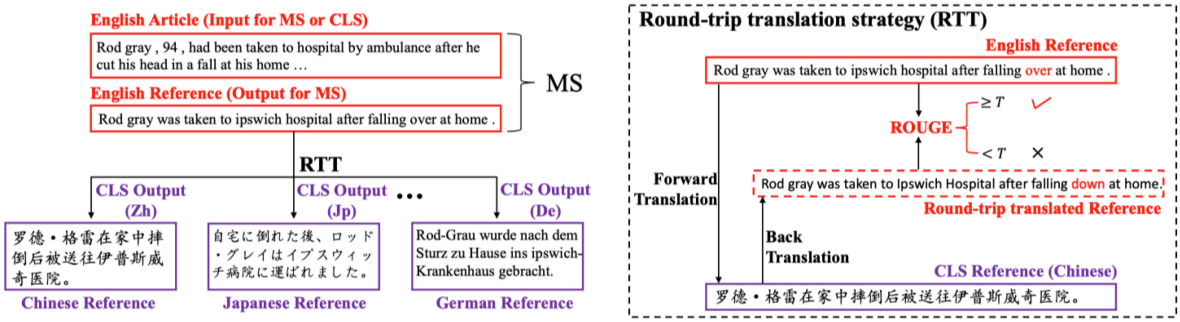

图 2: Round-trip translation

拿英语来说，原有英语摘要对为 (D_{en}, S_{en})，将 S_{en} 中的每一句话进行 round-trip 翻译，如果翻译之后的结果与原来的句子 ROUGE F1 值高于设定的值，那么就选取，否则就剔除。对于整个 S_{en}) 来说，如果三分之二的句子都被选取了，那么这个跨语言摘要对就会被采用。(对于中文来说也一样，计算 ROUGE 用 Char，实际写代码我用 BERT 词表把字符替换成了数字进行计算)。最开始看到这篇论文，我以为找到了某一种资源直接可以得到跨语言的摘要对，但实际还是通过一种自动化的方式，加了一些限制。不过也挺好，有了标准数据集，可以有 baseline 了，之前 Cross-Lingual 的论文都是各搞各的，现在即使做无监督，也可以有个比的。(不过目前见到的 cross-lingual 都是中英。) 最后得到 370759 En2Zh 对，1699713 Zh2En 对。

Baseline 方法包括了 Early Translation (ETran) 和 Late Translation (LTran)。还有基于端到端的 Transformer-based NCLS models (TNCLS)。之后利用 MS 和 MT 来进行多任务学习 CLS+MS，CLS+MT。对于 CLS+MS 来说，输入是一个文档，输出是不同语言的摘要。对于 CLS+MT 来说，输入也不同，一个是摘要数据，一个是翻译数据。如图3。

1. **TETran** 首先利用 LDC 训练基于 Transformer 的机器翻译模型，先翻译，再利用 LexRank 摘要。利用无监督摘要方法的原因是：在翻译以后，缺少这种单语语料的训练数据。

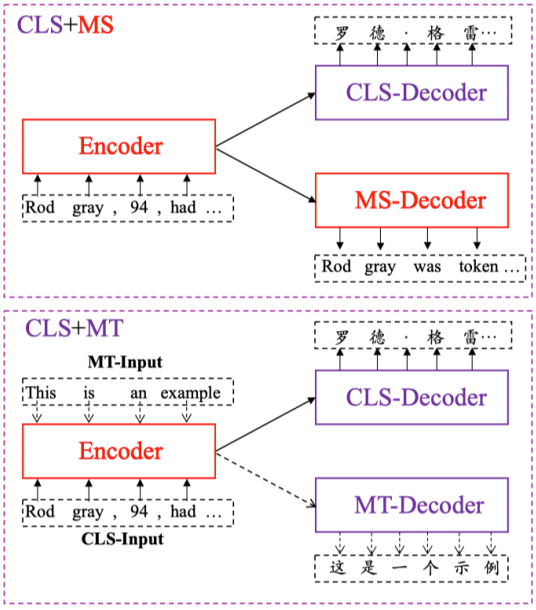

图 3: Multi-task NCLS

2. **TLTran** 利用基于 Transformer 的模型，分别训练翻译模型和单语摘要模型。串联进行。
3. **GETran** 利用 Google Translator。
4. **GLTran** 利用 Google Translator。
5. **TNCLS** 基于 Transformer 的 end2end 模型。
6. **CLS+MS** 结合 MS 多任务。
7. **CLS+MT** 结合 MT 多任务。

下面看一下实验结果4，还是有很多有意义的结果。

Model	Unit	En2ZhSum	En2ZhSum*	Zh2EnSum	Zh2EnSum*
		RG1-RG2-RGL(↑)	RG1-RG2-RGL(↑)	RG1-RG2-RGL(↑)	RG1-RG2-RGL(↑)
TETran	–	**26.12-10.59-23.21**	**26.15-10.60-23.24**	**22.81- 7.17-18.55**	**23.09- 7.33-18.74**
GETran	–	**28.17-11.38-25.75**	**28.19-11.40-25.77**	**24.03- 8.91-19.92**	**24.34- 9.14-20.13**
TLTran	c-c	–	–	32.85-15.34-29.21	33.01-15.43-29.32
	w-w	**30.20-12.20-27.02**	**30.22-12.20-27.04**	31.11-13.23-27.55	31.38-13.42-27.69
	sw-sw	–	–	**33.64-15.58-29.74**	**33.92-15.81-29.86**
GLTran	c-c	–	–	34.44-15.71-30.13	34.58-16.01-30.25
	w-w	**32.15-13.84-29.42**	**32.17-13.85-29.43**	32.42-15.19-28.75	32.52-15.39-28.88
	sw-sw	–	–	**35.28-16.59-31.08**	**35.45-16.86-31.28**
TNCLS	c-w	–	–	36.36-19.74-32.66	35.82-19.04-32.06
	w-c	**36.83-18.76-33.22**	**36.82-18.72-33.20**	–	–
	w-w	33.09-14.85-29.82	33.10-14.83-29.82	38.54-22.34-35.05	37.70-21.15-34.05
	sw-sw	–	–	**39.80-23.15-36.11**	**38.85-21.93-35.05**

图 4: Baseline 实验结果，* 代表了人工校正以后的结果

1. GETran、GLTran 要比 TETran、TLTran 效果好，说明在 Pipeline 方法中，翻译效果比较重要。

2. 只看 TNCLS 的 En2Zh 数据集，发现 w-c 的分割方式要比 w-w 好得多。也就是中文端用 char 靠谱。论文中的解释是：中文端基于 char 分割，可以显著降低词表大小，decode 时候生成更少 unk。

3. 在 Zh2En 上，sw-sw 效果好，论文解释是可以降低词表，减少 UNK，c-w 效果不如 sw-sw，但是论文没有实验 c-sw，不知道效果如何？

4. En2Zh 和 En2Zh* 结果差不多，原因是 En2Zh 是新闻领域的，现有 MT 在新闻领域效果不错。Zh2En 和 Zh2En* 效果相差很多，因为 Zh2En 是微博数据，MT 系统翻译效果差。人工校正就比较多。

5. 基于 Transformer 的 End2End 方法要比 pipeline 方法好。

下面是 Multitask 的结果5。

Model	En2ZhSum	En2ZhSum*	Zh2EnSum	Zh2EnSum*
	RG1-RG2-RGL(↑)	RG1-RG2-RGL(↑)	RG1-RG2-RGL(↑)	RG1-RG2-RGL(↑)
TNCLS	**36.83-18.76-33.22**	**36.82-18.72-33.20**	**39.80-23.15-36.11**	**38.85-21.93-35.05**
CLS+MS	38.23-20.21-34.76	38.25-20.20-34.76	41.08-23.67-37.19	**40.34-22.65-36.39**
CLS+MT	**40.24-22.36-36.61**	**40.23-22.32-36.59**	**41.09-23.70-37.17**	40.25-22.58-36.21

图 5: 多任务学习，En2Zh(w-c)、Zh2En(sw-sw)

1. 多任务学习对于 baseline 都有帮助。

2. 在 En2Zh 上，MT 帮助更显著，原因可能是：（1）MT 数据集大；（2）MT 数据集是新闻领域的。但是在 Zh2En 上，效果基本一致。

另外关于一些实验细节的说明，在 En2Zh 数据集上，En 没有采用 subword 分割，因为这样会导致输入太长，那当然 char 就更不用说了，没有意义。所以在 En2Zh 上输入端英文只用 word 分割。在 En2Zh 数据集上可以有两种方式，w-c、w-w，分别代表了英语输入端用 word，输出端中文用 char；英语输入端用 word，输出端中文用 word。对于 En2Zh 数据集输入截断为 200 个词语，英语摘要截断为 120 个词语，中文截断为 150 个字。

相比于单语言摘要，跨语言摘要的难点在哪里呢？反正都有了端到端的数据集了，直接训练不就好了？所以是不是还是应该从输入输出语言不同下手？能不能同这两个反向的互相帮助？再仔细想想 MT 和 MS 的作用？基于 cross-lingual pretrian 来做呢？

1.3 Text Summarization with Pretrained Encoders

来自于爱丁堡大学 Yang Liu 和 Mirella Lapata，Yang Liu 已经出过一个基于 Pre-train 的 Summarization 论文 [Liu, 2019]，当时在抽取式摘要上达到了 SOTA。这篇论文的抽取式摘要部分看起来就是 BERTSUM。在生成式摘要部分，由于 encoder 部分预训练过，decoder 部分没有预训练，所以有两个优化器来分别优化 encoder 和 decoder。

1.4 Neural Extractive Text Summarization with Syntactic Compression

本篇工作来自于得克萨斯大学奥斯汀分校，基于句子抽取（Extraction）和句子压缩（Compression）来完成单文档摘要。首先从文档中选取一些句子，然后对于选中的每一个句子，从一个压缩规则集合中选择一个规则对该句进行压缩。这些规则是从句法成分分析（syntactic constituency parses）取得的。因为是压缩是显示的规则，所以对于最后的生成结果，可解释性也比较强。

压缩规则：同位名词短语; 关系从句和状语从句; 名词短语中的形容词短语和状语短语 (见图6); 作为名词短语一部分的动名词短语 (参见6); 某些配置中的介词短语，比如周一; 括号和其他括号内的内容; 当然这些规则覆盖面肯定不是非常全，对于特定的领域和任务，可以制定特定的规则。一个具体的压缩示例6。

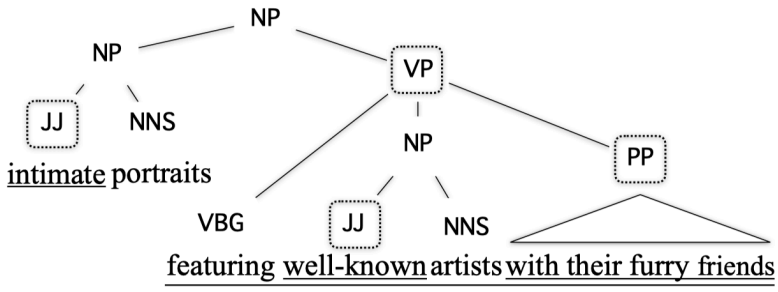

图 6: 文本压缩示例

句子抽取：文档中的句子用 Bi-LSTM 表示，CNN 得到句子整体表示，过 DOC LSTM 得到表示，DOC CNN 得到文档整体，表示然后一个 DOC LSTM decoder 来选择句子。在每一个 deocde step，拿到 doc 表示、上一时刻选取句子表示、上一时刻隐层表示，得到新的隐层表示，利用新的隐层和每一个句子表示得到 logit，选择最大的一个。被选取的句子之后不能再被选取，可以通过强制该句 logit 为 0 来实现。如图7。

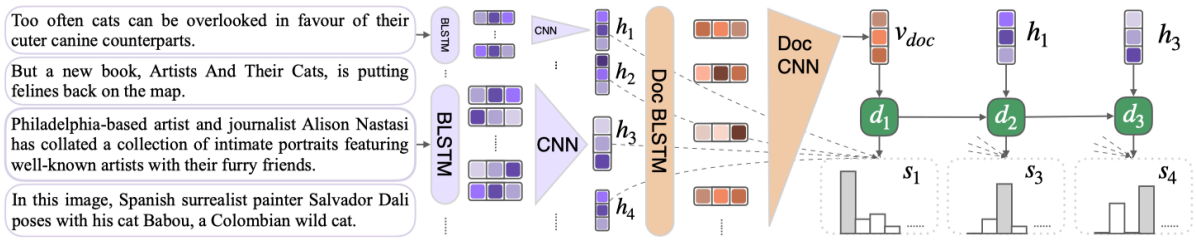

图 7: 句子抽取模型

句子压缩：根据规则可以得到可能压缩的 span，对于每一个可能压缩的词语或者短语，模型判断是否删除。标准的 Label 由规则给出。如图8。

1.5 Countering the Effects of Lead Bias in News Summarization via Multi-Stage Training and Auxiliary Losses

来自于加拿大麦吉尔大学的工作。虽然 lead bias 是一种非常显著的特征，尤其在新闻领域，但是并非所有情况都是这样的，过度的使用这种特征会导致一些重要内容并非集中在开始的样例效果差。如果能平衡 lead bias 和 semantic content selection 能力，就可以很好地应对两种情况。这篇论文提出了两种办法来增强内容选择的能力，一种是两步训练（Multi-Stage Training），首先在打乱句子顺序的数据集上预训练，然后在原始数据集上的训练。先在乱序的数据集上训练，可以使得

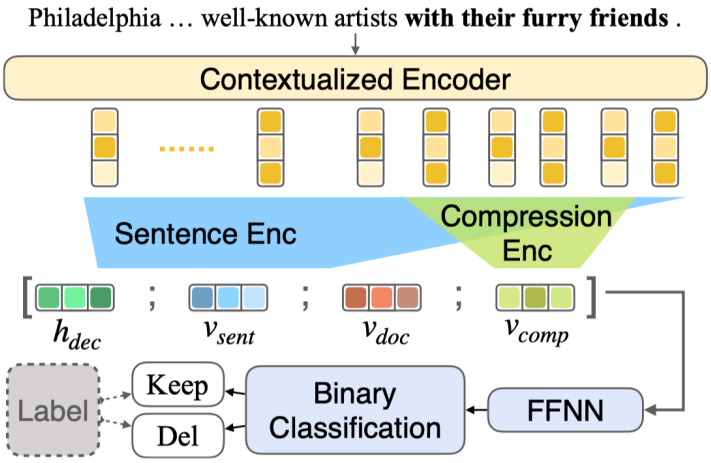

图 8: 句子压缩模型

模型最开始有一定的内容选择能力，而不会过分的依赖于选择 lead 位置。另一种是用一种句子级别的辅助 loss（Auxiliary Losses），在选择句子的时候，考虑预测的得分和真实的得分，会使得模型能够知道，即使在后面，也会有高分句子出现，这个方法有点类似于 [Zhou et al., 2018]，利用文本句子和 golden 来计算一个 rouge 得分，归一化为一个分布，模型会预测一个分布和这个分布算 loss。最后效果发现 pretrain 效果一般，没有太大提升，甚至会查。辅助 loss 效果好。应该认真探索一下 pretrain 的方法，怎么好使。

1.6 Explicit Cross-lingual Pre-training for Unsupervised Machine Translation

北航的工作，基于 XLM 又提出了一种新的训练方式，叫做 Cross-lingual Masked Language Model (CMLM)，核心是 MASK source 语言的一个 n-gram，预测 target 语言的对应翻译。首先这个翻译也是通过无监督的方式得来的，不过之前都是词对齐，这里是 n-gram 对齐。通过这个方式可以拿到一个推理出来的 n-gram 翻译词表。然后在 XLM 的基础上，按照 CMLM 的方法来进行预测，需要注意的是 mask 和预测可能长度不一致，因为是不同语言，所以基于 IBM model 进行了修改。如图9。

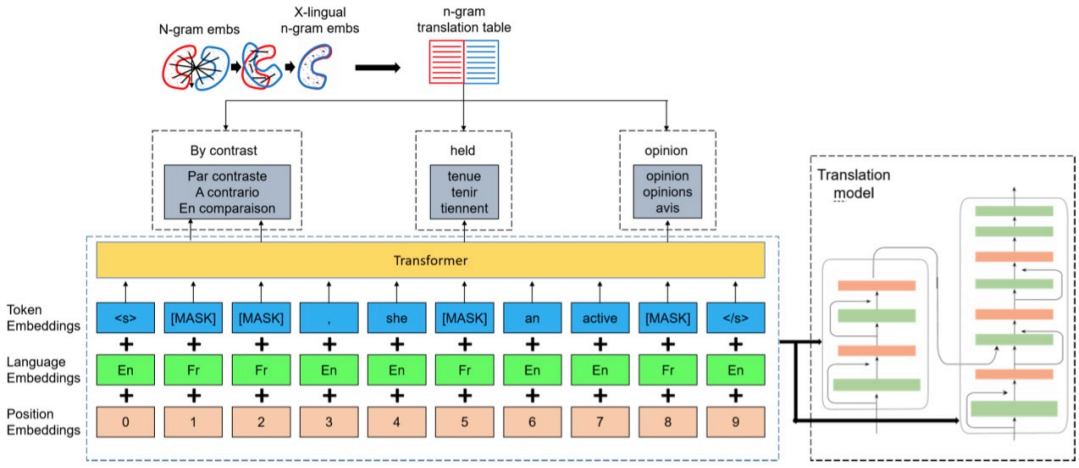

图 9: Cross-lingual Masked Language Model (CMLM)

1.7 Enhancing AMR-to-Text Generation with Dual Graph Representations

来自达姆施塔特工业大学。任务是 AMR-to-Text，放出了基于 openNMT 的代码，感觉以后可以用Graph2seq。还是利用图网络来完成生成任务，只不过利用了不同的图的表示形式，称之为 dual graph10，其实就是 top-to-down、down-to-top 两个方法，相当于两个方向吧，使得图的表示有两个 view 的表示，最后经过 Dual Graph Encoder 来编码11。

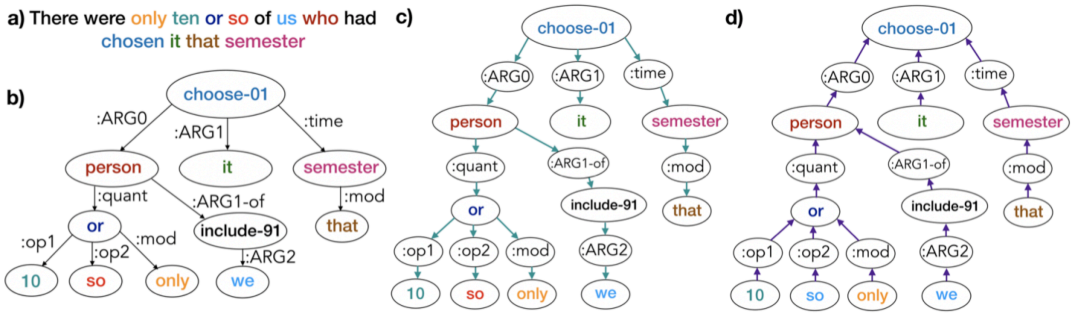

图 10: Dual Graph

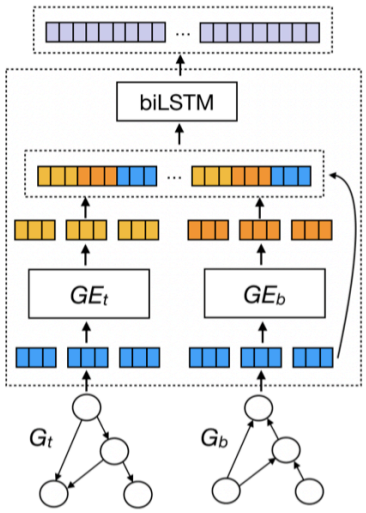

图 11: Dual Graph Encoder)

2 NIPS2019

2.1 Episodic Memory in Lifelong Language Learning

这篇论文来自 Deep Mind，二作是 NLP 界网红大哥 Ruder，cross lingual word embedding 硬核选手。这篇工作主要关注自然语言处理的终身（life long）学习问题，核心还是要解决灾难性遗忘问题，关于 lifelong，台大李宏毅有 7 节课讲解，涵盖了一些核心概念。论文模型比较简单，核心在于提出了一个 Episodic Memory，如图12。用于记录之前看到过的 Example，在训练的时候会进行 sparse experience replay，也就是会重新利用这些数据来计算梯度更新参数，在 inference 的时候，会进行 local adaptation，来在 memory 中查找 k 个最近的 Example 来先更新 model 参数到一个邻近空间下，然后再进行预测。（这个感觉像是 meta learning 的操作，在 test 的时候会先更

新一波参数到 related 空间下）。对于 life long 来说，多任务学习（multitask learning）是其上界，life long 考虑的是在每一个 task 都学完以后，model 需要保留之前的能力，不能新的数据集学完了，旧的之前会做也给忘了。通过实验验证了文本分类任务和 QA 任务效果都优于之前的方法。虽然这篇论文关注于终身学习，但是这个 key-value memory net 其实可以用到很多地方，值得借鉴。

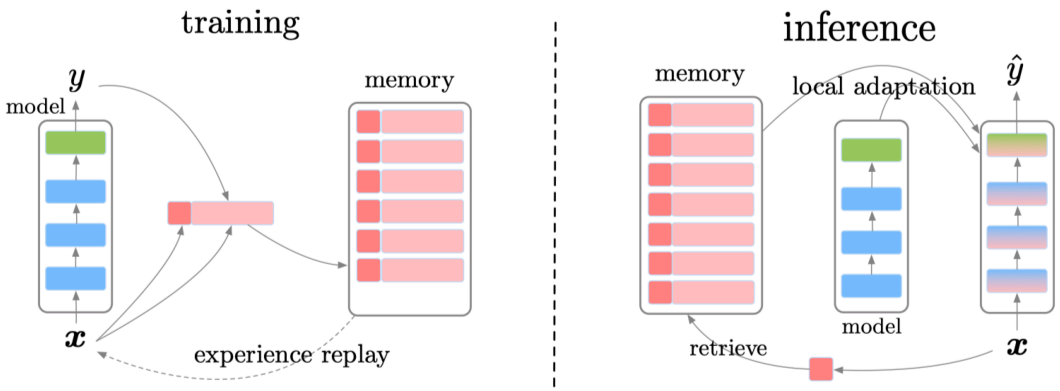

图 12: Episodic memory module

参考文献

Xiangyu Duan, Mingming Yin, Min Zhang, Boxing Chen, and Weihua Luo. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3162–3172, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1305.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Jiatao Gu, Changhan Wang, and Jake Zhao. Levenshtein transformer. *arXiv preprint arXiv:1905.11006*, 2019.

Baotian Hu, Qingcai Chen, and Fangze Zhu. Lcsts: A large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865*, 2015.

Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

Jessica Ouyang, Boya Song, and Kathy McKeown. A robust abstractive system for cross-lingual summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2025–2031, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1204. URL https://www.aclweb.org/anthology/N19-1204.

Haoyu Zhang, Yeyun Gong, Yu Yan, Nan Duan, Jianjun Xu, Ji Wang, Ming Gong, and Ming Zhou. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*, 2019.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305*, 2018.

Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. Msmo: Multimodal summarization with multimodal output. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4154–4164, 2018.