内蒙古大学 CCMT2022 蒙汉翻译评测技术报告

杨宗恒，侯宏旭*，孙硕，乌尼尔，菅伟辰，王翌松，王鹏聪

（1. 内蒙古大学计算机学院， 内蒙古 呼和浩特 010020；
2. 内蒙古自治区蒙古文信息处理技术重点实验室， 内蒙古 呼和浩特 010020）

摘要：本文主要介绍内蒙古大学计算机学院蒙古文信息处理技术重点实验室在第十八届全国机器翻译大会(CCMT2022)机器翻译评测项目中参赛的基本情况。在本次机器翻译评测中，我们参加了蒙汉综合领域双语翻译评测项目的在线评测和离线评测。本文采用基于自注意力网络的 Transformer 架构训练了一个基线蒙汉翻译模型，并使用非参数方法使用训练数据构建了一个外部记忆模块，在模型解码时通过检索与匹配从外部知识中获得指导以提升模型能力。本文主要介绍了该模型所采用的具体方法和实验细节，并通过实验表明了该模型在蒙汉双语翻译任务上性能的提升，并对此进行了深入的分析和研究。

关键词：机器翻译；Transformer；非参数方法

中图分类号：TP391　　**文献标志码：** A

1 引言

本文详细介绍了本单位参加第十八届全国机器翻译大会（CCMT2022）蒙汉综合领域双语翻译评测项目的情况。相较于往年的双语翻译评测，今年最大的改变就是蒙汉和藏汉翻译任务各新增了 100 万平行语料，使蒙汉翻译从稀缺资源语言的机器翻译任务变成了中等规模资源的翻译任务。因此以往积累的蒙汉翻译经验需要重新探索和验证，增加了模型的训练难度。

近年来，随着深度学习的发展，神经机器翻译已经取得长足的进步，为了进一步提升译文准确率，越来越多的研究[1-5]开始将训练数据作为某种外部知识，而不是作为模型参数来表达，称为“非参数”方法。由于该方法需要通过搜索来得到外部知识，因此也被称为“基于搜索”的模型。在实验中，我们采用 Transforme[6]神经网络架构作为基线翻译模型。在模型的增强方法上我们选择了 K-Nearest-Neighbor machine translation（KNN-MT）[1]的非参数方法来增强基线模型的能力。KNN-MT 使用 token 级的 k 近邻搜索，从外部记忆模块中获得知识指导，然后与翻译模型的预测结果相结合，极大提升了机器翻译的准确率。此外，我们使用带“耐心因子”控制的束搜索来提高模型的解码效果，进而对译文质量进行了进一步的优化。

2 数据

2.1 数据集统计及构建

我们参加的蒙汉双语评测训练语料今年提升到了百万级别，由于早期的验证集已被包含进了训练集，所以我们利用以往的数据自己构建了验证集。我们的验证集由 3,000 条 CCMT2022 的训练数据和 1,000 条 CCMT2019 的验证集组成。训练集由 CCMT2019 的训练集和 CCMT2022 的训练集组成。在线测试的测试集为 CCMT2021 的离线测试集，离线测试的测试集为 CCMT2022 的测试集。我们的数据统计如表 1 所示：

基金项目：内蒙古自治区科技成果转化专项“蒙古文机器翻译与辅助翻译平台建设与推广”（2019CG028）

* **通信作者：**E-mail: cshhx@imu.edu.cn

表 1 蒙-汉语料的统计信息

Tab. 1 Statistics of dataset in Mongolian-Chinese

	训练集	验证集	在线测试集	离线测试集
蒙-汉	1,244,829	4,000	5,970	10,000

2.2 数据预处理

我们对蒙古语进行 BPE（Byte Pair Encoding）[7]处理以获取一种介于词素义原和单词之间的子词粒度单元，这种粒度较小且具有一定语义信息的 token 单元能够极大地缓解由于语料稀疏而导致的低频词和资源浪费等问题，对于汉语语料由于其没有天然的单词边界，所以我们对其进行分词加 BPE 切分。汉语存在很多出色的中文分词工具，如 jieba、LTP、THULAC 等。本文使用了 THULAC[8]分词工具对汉语进行分词，分词模型由 THULAC 提供，同时支持分词和词性标注功能。该模型由人民日报分词和词性标注语料库训练得到。然后我们使用 subword-nmt[1]对语料进行 BPE 切分，在词表大小上我们选择了以最小词频设置词表大小，蒙古语最小词频为 7，汉语为 8。最终词表大小为蒙古语 57,960，汉语 49,440。由于我们的实验依赖于 Fairseq[9]，最后需要对语料进行二值化处理。测评中的训练集、验证集、测试集都做了同样的处理。

3 方法

由于我们选择了 KNN-MT 作为模型的增强手段，因此需要预训练一个表征能力强的翻译模型。我们训练了一个基于 Transformer 的基线翻译模型。然后使用 KNN-MT 对该模型在解码阶段利用外部知识进行增强，具体流程如图 1 所示：

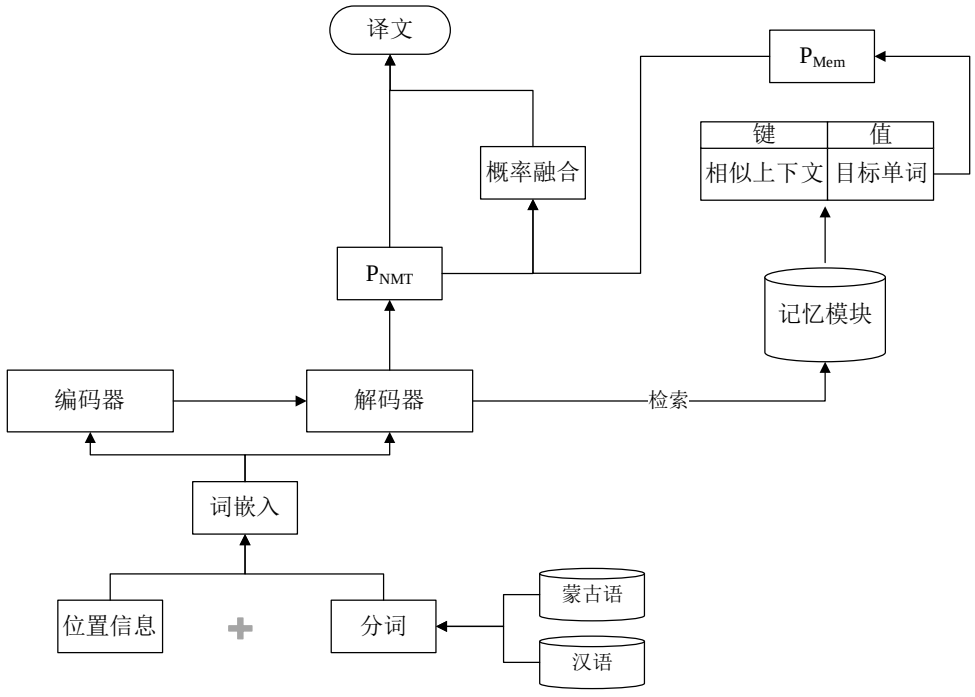


图 1 本文使用方法的流程图

Fig. 1 Flow chart of the method used in this paper

[1] https://github.com/rsennrich/subword-nmt

3.1 KNN-MT

KNN-MT 的方法主要包括记忆模块的构建和外部知识与模型预测结果的融合。在记忆模块构建方面，本文选取离线方式构建 token 级记忆库，其优点是检索效果更佳且匹配度较高，但需要预先训练一个知识表征能力较强的模型。记忆模块的形式为上下文向量和目标 token 的键值对，构建方法为将训练数据输入模型进行一次前向传递而得到。在上下文向量的选择上，根据实验证明选择解码器最后一层的前馈层的输入作为上下文向量效果最好。在给定双语语料$(x, y) \in (\mathcal{X}, \mathcal{Y})$，解码器根据源语言$x$和已经生成的单词$y_{<t}$来解码$y_t$。假设预训练模型的隐藏层状态为$f(x, y_{<t})$，则记忆库的键为$f(x, y_{<t})$值为$y_t$，构建过程如公式 1 所示：

$$(\mathcal{K}, \mathcal{V}) = \{(f(x, y_{<t}), y_t), \forall y_t \in y | (x, y) \in (\mathcal{X}, \mathcal{Y})\} \tag{1}$$

记忆模块构建完成之后就可以在解码阶段通过检索得到近似句子，通过近似句子对应的 token 可以得到一个检索概率，即记忆库通过历史数据给出的经验概率P_{Mem}。为了防止过度拟合到最相似的检索，我们使用了带有“温度”参数 T 的 softmax 函数。最终经验概率通过公式 2 得到：

$$P_{Mem}(y_i | x, \hat{y}_{1:i-1}) \propto \sum_{(k_i, v_i) \in \mathcal{N}} \mathbf{1}_{y_i = v_i} \exp\left(-\frac{-d\left(k_j, f(x, \hat{y}_{1:i-1})\right)}{T}\right) \tag{2}$$

经验概率代表了外部知识指导，KNN-MT 通过简单的线性插值将外部知识与模型知识进行融合，得到最终的概率分布，如公式 3 所示。

$$p(y_t | y_{<t}, x) = \lambda p_{NMT}(y_t | y_{<t}, x) + (1 - \lambda)\, p_{Mem}(y_t | y_{<t}) \tag{3}$$

3.2 带有“耐心因子”的束搜索

在机器翻译推断中，何时终止搜索是一个非常基础的问题。研究者一方面希望尽可能遍历更大的搜索空间，找到更好的结果，另一方面也希望在尽可能短的时间内得到结果。以往的研究[10-12]表明，“产生精确搜索的停止条件会带来性能上的提升。”这时搜索的终止条件就是一个非常关键的指标。针对这些问题，研究人员设计了很多新的方法。比如，可以在束搜索中使用启发性信息让搜索尽可能早地停止，同时保证搜索结果是“最优的”[10]。很多开源机器翻译系统也都使用了简单有效的终止条件。比如，在 OpenNMT 系统中当搜索束中当前最好的假设生成了完整的译文搜索就会停止，在 RNNSearch 系统中当找到预设数量的译文时搜索就会停止，同时在这个过程中会不断减小搜索束的大小。

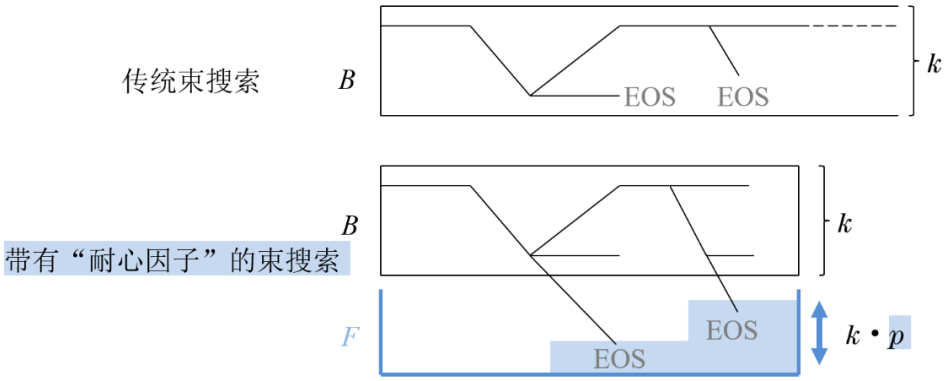


图 2 传统束搜索与带有“耐心因子”的束搜索的对比

Fig. 2 Comparison between vanilla beam search and beam decoding with controlled patience

带有“耐心因子”的束搜索[13]通过给束搜索添加一个“耐心因子”来控制束搜索的深度，来获得更

多的候选译文。传统束搜索通过增加 beam-size 来控制搜索的广度，而“耐心因子”是在不改变广度的情况下控制搜索的深度。对比示意图如图 2 所示。传统束搜索的停止条件为生成 beam-size 个<EOS>，而加入“耐心因子”之后，已经生成的句子被保存下来，模型可以在此基础上继续解码，停止条件为生成 k • p 个句子。因此加入“耐心因子”之后的译文长度会有一定的增加。在产生同样数量的候选译文时带有“耐心因子”的束搜索不会陷入到“波束诅咒”[14]，即当 beam-size > 5 时会生成不好的译文影响译文质量。我们使用带有“耐心因子”的束搜索代替常规的束搜索解码，使模型生成更准确的译文。并且该方法在 Fairseq 中仅需要修改一行代码即可实现。

4 实验

4.1 实验环境

表 2 实验环境配置

Tab. 2 Configuration of the experimental environment

CPU	Intel Core i7-11700F
GPU	NVIDIA GeForce 1660 Ti
内存	16GB
操作系统	Ubuntu 20.04.3
深度学习框架	Pytorch1.10.0
机器翻译框架	fairseq 0.10.1
向量检索工具	faiss 1.5.3

4.2 本文实验设置

主要参数设置如下，每个模型使用 1 块 GPU 进行训练，每个 batch 大小为 2048，参数更新频率设置为 2，学习率为 5e-4，学习率衰减策略为 inverse-sqrt，优化器为 Adam，warmup 步数为 1000。max-epoch 设置为 80，设置有提前停止。编码器的层数为 6 层，词向量维度为 512，隐层状态维度为 2048，解码器为 6 层，多头自注意力机制使用 8 个头。本次评测采用了 dropout 机制，dropout 设为 0.1。每两轮保存一个 checkpoint，并在验证集上测试。在推理阶段，本评测采用 fairseq 进行解码，beam-size 设置为 5，patience pactor 设置为 2。

外部记忆模块包含训练集中每个目标语言的 token。为了搜索这个大型数据存储，我们使用 FAISS[15] 对高维空间中的向量进行快速最近邻检索。FAISS 通过对关键点进行聚类并根据聚类质心查找“邻居”来加快搜索速度，同时通过存储矢量的压缩版本来减少内存使用。然后使用 1M 个随机采样的关键点创建 FAISS 索引，以学习 4096 个聚类中心。在推理过程中，我们检索 k=8 或 12 个邻居，索引在搜索最近邻居的同时查找 32 个聚类中心以获得最佳匹配。

4.3 实验结果与分析

在线测评：

表 3 为在线评测的主要结果，其中验证集的评价指标为 BLEU4[16]，以单个汉字或符号作为评估的基本单位。测试集的在线评价指标为 BLEU_SBP[17]，其中 KNN-MT 的超参设置为：验证集 k=8、lam=0.5、temperature=100；测试集为 k=12、lam=0.4、temperature=100。

根据实验结果分析发现 KNN-MT 对 Transformer 的提升极为明显，将训练数据作为模型的显式记忆的优势就是增强模型的学习与记忆能力，使模型的记忆能力突破了参数的限制。模型通过使用稠密向

量的检索来获得外部知识的指导并与模型自身知识相融合，增强模型的能力。而带有“耐心因子”的束搜索对结果的提升没有那么明显。但是经过我们的实验证明与传统束搜索相比可以稳定的获得 0.4-0.9 个 BLEU 的提升，可以完美替换传统束搜索的解码效果。

表 3 蒙-汉在线评测在验证集和测试集的实验结果

Tab. 3 Experimental results of Mongolian-Chinese online evaluation on verification set and test set

模型	验证集	测试集
Transformer	35.26	0.3491
+KNN-MT	39.61	0.3942
+ Beam with patience	39.96	0.3997

离线测评：

表 4 为离线评测的主要结果，其中验证集的设置与在线评测一致。为了使我们能够选取最优超参数，我们使用奥云翻译[2]对测试集的 10,000 条蒙古语句子进行翻译，使用奥云翻译的译文作为伪真实译文，然后构建了伪离线测试集。测试集的评价指标为 BLEU4，其中 KNN-MT 的超参设置为：k=8、lam=0.5、temperature=100。由表中可以看到，KNN-MT 对基线模型的提升存在着边际效应，当基线模型的翻译能力较强时，外部知识对模型的影响就会相对变弱。

表 4 蒙-汉离线评测在验证集和测试集的实验结果

Tab. 4 Number of <unk> in translation of validation set and test set

模型	验证集	测试集
Transformer	35.26	73.65
+KNN-MT	39.61	74.93
+ Beam with patience	39.96	75.22

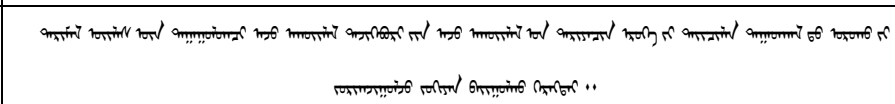
并且通过对译文的分析发现我们的方法能一定程度的缓解<unk>问题。我们的方法生成的译文中<unk>的数量明显少于基线模型，表明当基线模型无法解码生成未登录词时，模型可以通过外部知识的指导从历史翻译中获取信息生成更流畅的译文。

译文分析：

表 6 展示了测试集中基线模型和我们的方法生成译文的例子，其中参考译文来自奥云翻译。对比基线模型，我们的方法在 BLEU 值提高的同时，生成的译文流利度也有所提升。在外部知识的指导下能生成更加忠实与源句的译文，且在低频词的表现上更加出色。其中“鼓励”和“组织”在基线模型中分别出现了漏译和语序错误的问题，我们的方法通过检索与融合最终能得到较为准确的译文。

表 5 蒙-汉翻译实例

Tab. 5 Examples of Mongolian-chinese translation

	蒙-汉
源句	
参考译文	要鼓励、支持、组织种植业龙头企业、养殖企业带动农户投保。
基线模型	种植业龙头企业、养殖企业带动农户投保，宜组织。
本文方法	要鼓励和组织种植业龙头企业、养殖企业带动农户投保。

[2] http://fy.mglip.com/pc

5 总结

本文主要介绍了内蒙古大学计算机学院参加第十八届全国机器翻译大会（CCMT2022）评测的情况。我们在百万规模数据的蒙汉双语评测中，训练了一个基于 Transformer 的蒙汉翻译模型，并使用非参数方法 KNN-MT 对模型进行增强，通过给模型添加一个外部记忆模块使模型的学习与记忆能力大大增强，以生成更加准确的译文。并且我们还探索了带有“耐心因子”的束搜索的应用前景，发现其可以作为传统束搜索的一种有效替代手段，也是一种稳定提升译文质量的小技巧。在未来，我们会持续探索优化蒙汉翻译任务下的非参数方法的应用。

参考文献：

[1] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer and Michael Lewis. “Nearest Neighbor Machine Translation” International Conference on Learning Representations (2021).

[2] Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., & Lewis, M. (2019). Generalization through memorization: Nearest neighbor language models. arXiv preprint arXiv:1911.00172.

[3] Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive Nearest Neighbor Machine Translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 368–374, Online. Association for Computational Linguistics.

[4] Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang and Jiwei Li. “Fast Nearest Neighbor Machine Translation..” arXiv: Computation and Language (2021): n. pag.

[5] Akiko Eriguchi, Spencer Rarrick and Hitokazu Matsushita. “Combining Translation Memory with Neural Machine Translation..” Empirical Methods in Natural Language Processing (2019).

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. “Attention is All you Need” Neural Information Processing Systems (2017).

[7] Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

[8] Sun, M., Chen, X., Zhang, K., Guo, Z., & Liu, Z. (2016). Thulac: An efficient lexical analyzer for chinese. 2017-01-17)[2019-04-02]. https://github. com/thunlp/THULACPython.

[9] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., ... & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. arXiv preprint arXiv:1904.01038.

[10] Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to Finish? Optimal Beam Search for Neural Text Generation (modulo beam size). In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2134–2139, Copenhagen, Denmark. Association for Computational Linguistics.

[11] Yang, Y., Huang, L., & Ma, M. (2018). Breaking the beam search curse: A study of (re-) scoring methods and stopping criteria for neural machine translation. arXiv preprint arXiv:1808.09582.

[12] Ma, M., Zheng, R., & Huang, L. (2019). Learning to stop in structured prediction for neural machine translation. arXiv preprint arXiv:1904.01032.

[13] Kasai, J., Sakaguchi, K., Bras, R. L., Radev, D., Choi, Y., & Smith, N. A. (2022). Beam Decoding with Controlled Patience. arXiv preprint arXiv:2204.05424.

[14] Yang, Y., Huang, L., & Ma, M. (2018). Breaking the beam search curse: A study of (re-) scoring methods and stopping criteria for neural machine translation. arXiv preprint arXiv:1808.09582.

[15] Jeff Johnson, Matthijs Douze and Hervé Jégou. “Billion-Scale Similarity Search with GPUs” IEEE Transactions on Big Data 7 (2021): 535-547.

[16] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).

[17] Chiang, D., DeNeefe, S., Chan, Y. S., & Ng, H. T. (2008, October). Decomposability of translation metrics for improved evaluation and efficient algorithms. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (pp. 610-619).

IMU CCMT2022 Mongolian-Chinese Translation Evaluation Technical Report

ZongHeng Yang, Hongxu Hou*, Shuo Sun, Nier Wu, Weichen Jian, Yisong Wang,

Pengcong Wang

(1.College of Computer Science, Inner Mongolia University, Inner Mongolia, Hohhot 010020; 2. Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Inner Mongolia, Hohhot 010020)

Abstract: This paper mainly introduces the basic information of the participation of the Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, College of Computer Science, Inner Mongolia University in the machine translation evaluation project of the 18th National Machine Translation Conference(CCMT2022). In this machine translation evaluation, we participated in the online evaluation and offline evaluation of the bilingual translation evaluation project in the integrated field of Mongolian and Chinese. In this paper, a baseline Mongolian-Chinese translation model is trained using the Transformer architecture based on self-attention networks, and an external memory module is constructed using the training data in a non-parametric method to obtain guidance from external knowledge during model decoding by retrieval and matching to enhance the model capability. In this paper, we introduce the specific methods and experimental details adopted by the model, and show the performance improvement of the model on the Mongolian-Chinese bilingual translation task through experiments, which are analyzed and studied in depth.

Keywords: Machine Translation; transformer; non-parametric method