A Document-Level Neural Machine Translation Model with Dynamic Caching Guided by Theme-Rheme Information

Yiqi Tong, Jiangbin Zheng, Hongkang Zhu, Yidong Chen[*], Xiaodong Shi
Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen, China
{yqtong, jiangbinzheng, hkzhu}@stu.xmu.edu.cn
{ydchen, mandel}@xmu.edu.cn

Abstract

Research on document-level Neural Machine Translation (NMT) models has attracted increasing attention in recent years. Although the proposed works have proved that the inter-sentence information is helpful for improving the performance of the NMT models, what information should be regarded as context remains ambiguous. To solve this problem, we proposed a novel cache-based document-level NMT model which conducts dynamic caching guided by theme-rheme information. The experiments on NIST evaluation sets demonstrate that our proposed model achieves substantial improvements over the state-of-the-art baseline NMT models. As far as we know, we are the first to introduce theme-rheme theory into the field of machine translation.

1 Introduction

Most state-of-the-art Neural Machine Translation (NMT) models (Bahdanau et al., 2014; Sutskever et al., 2014; Vaswani et al., 2017) regard independent sentence pairs as their training and decoding units without considering the document-level context. Due to the ignorance of the discourse connections between sentences and other valuable contextual information like coreference, translations produced by such NMT systems tend to be problematic in coherence and cohesion, e.g. inconsistent translations of the same words, under-translation or mistranslation of the topic words, etc. (Hardmeier, 2012; Meyer and Webber, 2013; Smith, 2017).

In the past few years, research has been conducted to incorporate inter-sentence information into NMT and the previous works on context-aware NMT models have shown improvements on the document-level translation (Maruf and Haffari, 2017; Jean et al., 2017; Kuang and Xiong, 2018; Miculicich et al., 2018; Su et al., 2018; Tu et al., 2018; Voita et al., 2018; Mansimov et al., 2020; Zheng et al., 2020). Recent studies (Weston et al., 2014; Maruf and Haffari, 2017; Su et al., 2018) introduce an external architecture to produce contextual representation during the translation of a sentence. However, they do not exploit the representations already learned by the NMT encoder and the external structure is inflexible. More recently, Tu et al. (2018) and Kuang and Xiong (2018) propose cache-based NMT models to capture document-level information. In these models, one can define flexible caching rules so that the stored information may be more interpretable. However, these methods usually focus on the target-side context and may suffer from the problem of error propagation, since the target-side context that is used as cache often contains translation errors. Besides, these two methods require high-quality and large-scale parallel corpus with document boundaries, which are seldom available.

Previous works proved that modeling the source-side context is an effective way to improve the performance of a document-level NMT model (Yang et al., 2016; Miculicich et al., 2018). These methods just need small-scale of high-quality corpus with document boundaries. But the current strategies only make use of limited source-side context and are too simple. They usually ignore long-term context outside the k sentences or n words windows and neglect the rest of the discourse. So, we need an extra model to store long-term context while distinguishing important context from noises.

[*] Corresponding author.

In this paper, we proposed an improved document-level NMT model that combines a cache model with source-side context modeling. To model the inter-sentence information, we set a dynamic cache to store the history of encoder states. Meanwhile, motivated by the theory of systematic functional linguistics (Halliday et al., 2014), the mechanism for updating the context information is guided by theme-rheme information. To integrate cached context into the NMT model, the classical cache-based NMT models usually rely on a query-and-read mechanism (Gong et al., 2011; Tu et al., 2018). Instead of using the query-and-read mechanism, our model uses multi-head attention (Vaswani et al., 2017) structure and context gate (Tu et al., 2017) to get corresponding contextual representation.

Our key contributions are as follows:

- We propose a cache-based NMT model with a dynamic cache that capturing the source-side inter-sentence information. And, the experimental results that show our approach has substantial improvements in translation quality as measured via the BLEU score.
- We propose a novel way to automatically guide the dynamic caching with source-side inter-sentence context information, i.e. theme-rheme information. To the best of our knowledge, this is the first work of introducing the theme-rheme theory into the field of NMT.
- We conduct qualitative analysis to validate the effectiveness of our model and the results confirm that our model can generate coherent discourse translations.

2 Related Work

Most MT models are built on strong independent assumptions whether it is based on locality assumptions within a sentence as done by phrase-based models or that outside the sentence as done by even the most advanced NMT models today (Maruf et al., 2019). Text, on the contrary, doesn't consist of isolated, unrelated elements, but of collocated and structured group of sentences bound together by complex linguistic elements. Ignoring these linguistic elements, like theme-rheme information (Xi and Zhou, 2017; Kang et al., 2019), results in translations which may be perfect at the sentence-level but disappointing at document-level. Among many of the document-level translation works, our inspiration comes from the cache-based methods and the source-side context modeling methods.

In researches related to cache-based methods, the concept of cache was introduced by Kuhn and De Mori (1990). Gong et al. (2011) proposed cache-based SMT model and introduced three cache models: dynamic cache, static cache and topic cache. In recent researches, Kuang and Xiong (2018) integrated dynamic cache and topic cache into the NMT model. Tu et al. (2018) only used a dynamic cache to capture target-side context information but setting their cache model to key-value structure like Miller et al. (2016), the final context vector from the cache was then combined with the decoder hidden state via a gating mechanism, and the cache had a fixed length and was updated after generating a completed translation sentence. In our model, we set a similar key-value dynamic cache but using very different mechanism.

In researches related to source-side context modeling methods, many RNN-based methods have been proposed (Lin et al., 2015; Wang and Cho, 2015). The main idea is using other sentences in the document as context while translating current sentences. In recent researches, Zhang et al. (2018) used an extra Transformer encoder to model the context directly. Kuang and Xiong (2018) used inter-sentence gate to control the scale of information flowing from context and current sentence. Miculicich et al. (2018) put forward an attention-based method, using hierarchical attention network to model the context. In our model, we draw lessons from the attention-based method, using a simpler multi-headed attention structure to capture the inter-dependencies between current decoder state and cached context.

3 Our Model

In this section, we will describe our model in detail. First, the overall document-level NMT model framework will be proposed in Subsection 3.1. Then, the internal mechanism including the cache model, the contextual attention structure and the context gating will be presented in Subsection 3.2, 3.3 and 3.4, respectively.

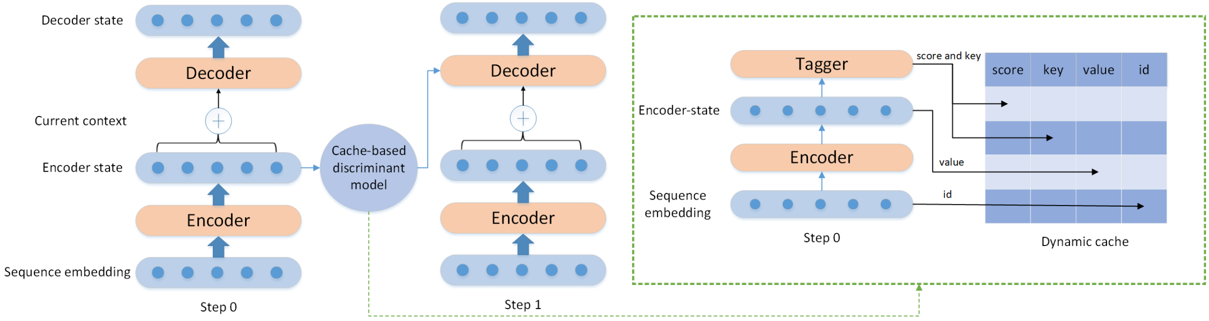

Figure 1: Illustration of the overall architecture of our model. First, we pre-training a Transformer encoder with bilingual parallel sentence pairs, then use the encoder to train a theme-rheme labeling model on annotated discourse corpus as a discriminator. In step 0, the discriminator selects the theme-rheme information of current sentence into the cache, and by step1, the cache will participate in the decoding process of the next sentence.

3.1 The Overall Framework

The aim of an NMT model is to maximize the probability of the target language sequences $Y = (y_1, ..., y_j)$ given the source language sequences $X = (x_1, ..., x_i)$, which is calculated as follows:

$$argmax \frac{1}{N} \sum_{n=1}^{N} log(P_\theta(Y^n|X^n)) \tag{1}$$

where n is the decoding step and N is the length of current sentence. Moreover, when translating documents, we should consider the context information D^n at the sentence decoding step n, the conditional probability could be modified as follows:

$$argmax \frac{1}{N} \sum_{n=1}^{N} log(P_\theta(Y^n|X^n, D^n)) \tag{2}$$

where D is a fixed-size dynamic cache consisted of the sorted history encoder state. Figure 1 illustrates the overall architecture of our proposed NMT model. Concretely, during the n-th sentence translating step, the encoder is responsible for generating the source-side hidden states $(h_1, \ldots, h_t)$ like a normal encoder. Then different weight scores will be assigned to each hidden state h_i by discriminant model, and some hidden states that obtained high score will be selected into D^n. When the dynamic cache is full, those states with a low scores will be forgotten.

While translating sentence $t(t > 0)$, the context information D^n_t will be integrated into current decoder states. For the first sentence in the document, the decoder will just consider current sentence because D^n_0 is null. Moreover, if the current sentence at the end of a document, the encoder states will not be stored and D will be reinitialized.

Clearly, our improved framework makes it more flexible to maintain the context information of previous sentences and allows the long-term context information to propagate through the translating process.

3.2 Dynamic Cache

In this research, we propose a dynamic cache mechanism, which use the theme-rheme information of sentences as guiding signals. Here, the concepts of theme and rheme come from the theme-rheme theory (Halliday et al., 2014), which has been widely accepted and applied in the field of discourse analysis like Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) and PDTB-based theory (Miltsakaki et al., 2004; Prasad et al., 2008).

In systemic functional grammar, theme is the elements which serves as the point of departure of the message and rheme is the remainder of the message. As shown in Table 1, the theme-rheme related words were marked as bold font. We can find that themes, especially the topical themes, usually contain topic

Theme(Topic)	**Rheme(Normal)**
Discounted stamp and **sheetlet**	how to keep going
Discounted stamp	means a set of **discounted stamp** and **sheetlet**
This	is the biggest part in **stamp** field

Table 1: Example of theme-rheme structure of three consecutive sentences.

information, which are relevant to other sentences. And rheme usually contains descriptive information about the corresponding themes. To integrate theme-rheme information into the cache, it is essential to label the theme-rheme information for given sentences. To this end, we introduce a Bi-LSTM based theme-rheme labeling model into the cache model.

In the caching process, we take current encoder states as input. The theme-rheme labeling model will give each encoder state a labeling state o_t:

$$o_t = Concat(f(h_t, s_t), f'(h_{n-t}, s_t)) \tag{3}$$

where t is the current encoding step, n is the length of the input sequence, f is the forward LSTM and f' is the backward LSTM. The new state represents the document structural element. Meanwhile, to measure the importance of o_t, we use a logistic linear model to represent o_t as a score s_t between zero and one:

$$s_t = sigmoid(W_l o_t) \tag{4}$$

Our dynamic cache was designed to be a key-value structure (Miller et al., 2016). Therefore, while storing encoder state h_t, we store a quadruple (s_t, o_t, h_t, id) by setting o_t as key and h_t as value. In addition, the original vocabulary index id is stored together for stop words filtering.

The final caching mechanism works as follow:

- If current id is in stop word list, ignore current state.
- If dynamic cache is full, compare current s_t with the minimum score in the cache. If s_t is smaller, ignore current state. Else seeking whether current id has already in the cache. If so, replace the quadruple which has identical id with new quadruple. Else replace the quadruple which has minimum score.

The theme-rheme labeling model was trained separately on a labeled corpus. By setting the logistic linear optimization target of designated label as one, we can make the dynamic cache focus on specified information.

$$loss_1 = H(p_l) + D_{KL}(p_l || p_{label}) \tag{5}$$

$$loss_2 = \frac{1}{n}\sum_{i=1}^{n}(y_{label} - y_l)^2 \tag{6}$$

$$loss_{final} = loss_1 + loss_2 \tag{7}$$

During the training process, as shown in Figure 2(a), we joint training the Bi-LSTM theme-rheme labeling network and the scoring logistic linear model. For theme-rheme labeling predicting, cross-entry loss l_1 on predicting distribution p_l was used. And for logistic linear predicting, an MSE loss l_2 on predicting value y_l was used, the final loss was the sum of l_1 and l_2.

3.3 Contextual Attention

Most previous cache-based NMT models usually apply the inter-sentence information by simply querying a similar state in the cache. To better model the history states, we integrate a multi-headed attention structure. Compared with the query-and-read mechanism, an attention-based structure is trainable and

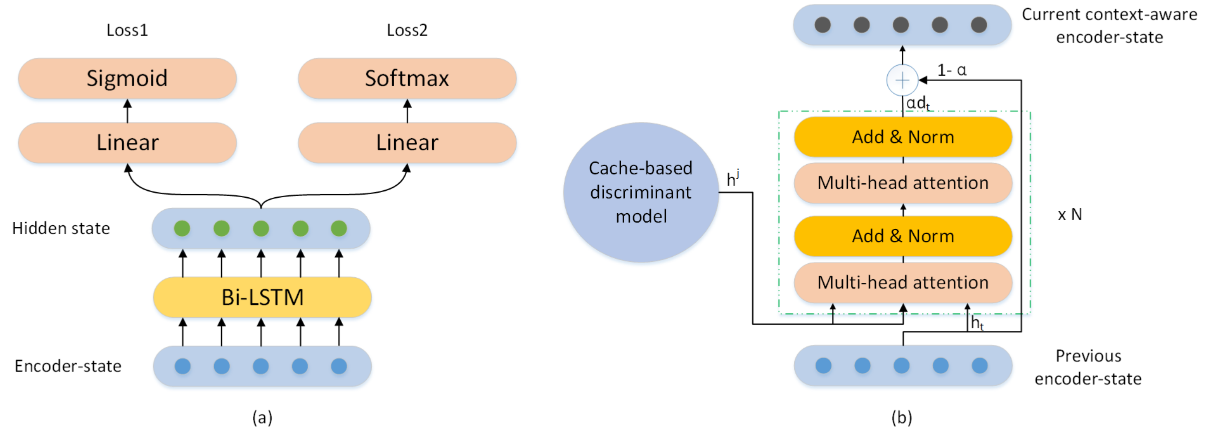

Figure 2: Illustration of the architecture of our dynamic cache. In (a), to get the cache-based discriminant model, we joint training the theme-rheme labeling model and logistic linear model. In (b), to get the context-aware encoder-state for current sentence, we compute context representation d_t with history encoder states h^j and current encoder state h_t. Finally, we use context gating to combine d_t with h_t.

more flexible. Concretely, we take current encoder state h_t as query. Normally, key and value are both history encoder state h_j. Thus, t-th final context representation d_t is calculated as follow:

$$d_t = FNN(MultiHead(h_t, h^j, h^j)) \tag{8}$$

As Figure 2(b) shows, FNN is a position-wise feed-forward layer and each layer is followed by a normalization layer (Ba et al., 2016). Please note that, we don't use the theme-rheme labeling states in contextual attention process because a different parameter distribution will lower the effect of attention model.

3.4 Context Gating

We use a context gate (Tu et al., 2017; Tu et al., 2018) to combine context representation d_t with original encoder state h_t. Because each word requires different level of context information. The function of context gate is assigning weight α for d_t and h_t as follows, W and U are linear transformer matrix of h_t and d_t, while c_t is the context-ware encoder state after introducing d_t to h_t.

$$\alpha = sigmoid(Wh_t + Ud_t) \tag{9}$$

$$c_t = \alpha d_t + (1 - \alpha)h_t \tag{10}$$

4 Experiments

We carried out a series of Chinese-to-English translation experiments to evaluate the effectiveness of our proposed document-level NMT model and conducted in-depth analyses on experiment results and translations.

4.1 Datasets

Our datasets consist of the following three parts:

- LDC: Contains 1.25M parallel sentence pairs selected from the origin LDC corpus.

- FBIS: Contains 10,355 documents with 230K parallel sentence pairs that have document boundary information.

Model	**Accuracy**
char-CNN+Bi-LSTM+CRF	83.56%
NMT-encoder+Bi-LSTM	86.45%

Table 2: Comparison of the prediction accuracy.

- CTRD[1]: Contains 525 documents with 45K Chinese monolingual sentences that have theme-rheme label.

We selected origin LDC corpus as our bilingual training data for pre-training the base NMT model. We also used FBIS corpus and CTRD to train the final document-level NMT model, where document boundaries are kept. It should be noted that, since there is no open test dataset for the automatic theme-rheme labeling task, we respectively extracted 2K sentences from the labeled CTRD as validation set and test set, and took the rest part as training data.

Finally, for evaluating the NMT systems, we chose NIST04 dataset as validation set, and NIST02, NIST05, NIST06, NIST08 as test sets.

4.2 Initialization

We adopted a Transformer from OpenNMT (Klein et al., 2017) as baseline model to carry out our experiment. The encoder and decoder were both set to 6 layers. The dimension of each hidden layers was 512, with 0.1 dropout and 8 heads for multi-head attention. For the automatic theme-rheme labeling model, we keep the dimension as 512, but only 2 layers. For the contextual attention layer, we use the same setting with baseline model hidden layer. All parameters were initialized with a uniform (0.01) distribution.

The vocabulary sizes were set both 50K for Chinese and English. And we replaced the rate words with ”<UNK>”, which was also in the dynamic cache stop words list. The dynamic cache size is 25. We wanted to keep the original form of the source-side text, so we didn’t use Byte-Pair-Encoding (BPE) (Sennrich et al., 2015).

4.3 Training

In the NMT model training process, we set each mini-batch with 2.2K-2.4K source language and target language tokens. We used Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$.

There are two optimizing steps. Firstly, pre-training stage, we trained a base Transformer with learning rate 0.001 and 10 epochs. And then, based on base Transformer encoder, we trained the theme-rheme labeling model with learning rate 0.001 and 5 epochs. Combined with the input of theme-rheme labeling model, inherited encoder was used to represent the source-side text and the encoder states. In this process, we set each mini-batch with 256 sentences.

Finally, we trained the contextual attention layer based on the pre-trained model, with learning rate 5×10^{-6} for 3 epochs and the mini-batch size was same as the base Transformer. It is because the pre-trained Transformer parameters are partly overfitting. Overtraining the contextual attention will destroy the stability of the pre-trained Transformer parameter.

5 Results and Analysis

In this section, we verified the performance of the theme-rheme labeling model, standard Accuracy, Precision, Recall and F1-score was used. We use BLEU score (Papineni et al., 2002) to evaluate the overall performance of our cache-based model (calculated by multi-bleu.per[2]). Finally, we used paired bootstrap resampling method (Koehn, 2004) to measure the significance level[3].

[1] Available at https://github.com/ydc/ctrd.

[2] https://github.com/OpenNMT/OpenNMT-py/blob/master/tools/multi-bleu.perl.

[3] https://github.com/pytorch/translate/blob/master/pytorch_translate/bleu_significance.py.

Model	Precision	Recall	F1-Score
char-CNN+BiLSTM+CRF	87.29%	83.82%	85.52%
NMT-encoder+BiLSTM	86.73%	86.56%	86.65%
Scoring logistic linear	86.98%	86.61%	86.79%

Table 3: Comparison of the topic theme prediction performance. It shows that our model can recognize more topic theme components than the strong baseline model.

Model	NIST02	NIST04	NIST05	NIST06	NIST08	Avg
2-layers						
Transformer	42.04	42.82	39.53	38.78	29.44	38.52
Our model	**42.43** (+0.38) **	**43.77** (+0.95) *	**40.2** (+0.77) ***	**39.18** (+0.4) **	**29.77** (+0.33) ***	**39.09** (+0.57)
6-layers						
Transformer	42.6	44.14	40.46	39.52	30.47	39.44
Our model	**42.90** (+0.3) ***	**44.80** (+0.66) ***	**40.90** (+0.44) ***	**39.89** (+0.37) ***	**30.66** (+0.19) **	**39.83** (+0.39)

Table 4: BLEU score for our dynamic cache-based model compared with baseline model. We compared the overall performance both on 2-layers Transformer and 6-layers Transformer. The significance is identified by *. The corresponding p-value are described as follow: *<0.05, **<0.01 and ***<0.001.

5.1 Dynamic Cache Performance

As shown in Table 2, to analyze the effectiveness of our theme-rheme labeling model, we compared with the strong sequence labeling model that achieved by NCRF++ system (Yang and Zhang, 2018). The performance of our model is better because we used a pre-trained NMT model encoder to represent source-side text while the embedding layer of the char-CNN+Bi-LSTM+CRF model was trained on a small-scale of CTRD corpus.

Furthermore, the primary goal of the automatic theme-rheme labeling is to identify the topic theme components correctly. So, as shown in Table 3, we analyzed the topic theme predicting performance between our model and the baseline model, while our F1-Score is better. We noted that our model will benefit from a higher Recall because it allows the NMT to use more inter-sentence information.

Besides, we verified the performance of the scoring logistic linear model. Compared with the theme-rheme labeling model, the Precision and Recall are both closed to the performance of the topic theme prediction. It shows that the scoring logistic linear model can give topic theme components a higher score when topic theme components are recognized correctly.

5.2 Translation Performance

Table 4 shows the different BLEU scores between our dynamic cache-based model and baseline model. We carried out experiments on two different scale of models. Experiments result proved that the performance of our theme-rheme information guided model improved significantly over the state-of-art baseline model. The improvement of 2-layers cache-based model was larger than 6-layers one over the baseline model. It was because our cache-based model was trained from baseline model and the parameters of 6-layers baseline were partly overfitting.

Furthermore, we wanted to prove that the significant improvement was indeed brought by the theme-rheme information guided cache. For this purpose, as shown in Table 5, we carried out two experiments. Firstly, we set cache size as zero and re-trained the model on the FBIS corpus with same optimizer, which was equivalent to training the baseline model twice. The BLEU score of two-times trained-model has no significant improvement. Secondly, we want to compare our theme-rheme information guided caching strategy with the current frequently used context selecting strategy. So, to compare the performance of continuous cache with the theme-rheme information guided cache, we carried out experiments on a

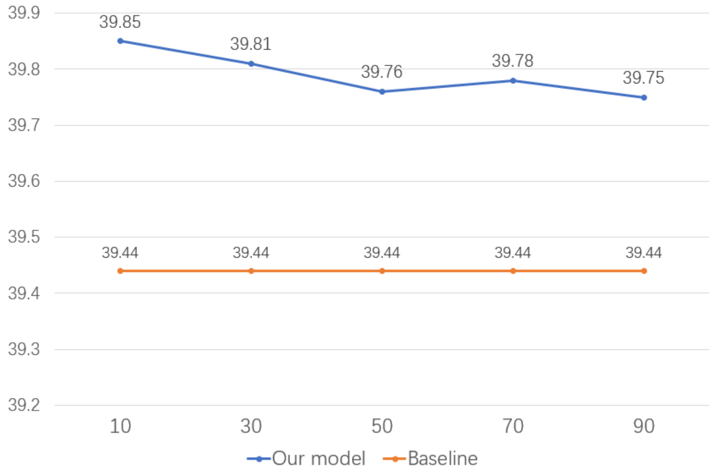

Figure 3: The comparison of our model and baseline model, where x-axis is the cache-size and y-axis is the standard BLEU score.

Model	NIST02	NIST04	NIST05	NIST06	NIST08	Avg
Base Model						
Transformer	42.6	44.14	40.46	39.52	30.47	39.44
+two-times training	42.73 (+0.13)	44.16 (+0.02)	40.57 (+0.11)	39.34 (-0.18)	30.55 (+0.08)	39.47 (+0.03)
Our Model						
+continuous cache	42.58	44.18	40.54	39.71	30.43	39.53
+theme-rheme information guided cache	**42.90** (+0.32) ***	**44.80** (+0.62) ***	**40.90** (+0.36) ***	**39.89** (+0.18) **	**30.66** (+0.23) ***	**39.83** (+0.3)

Table 5: BLEU score comparison with different training strategies. We compared the performance based on 6-layers. We tested significance of translation between theme-rheme information guided cache and continuous cache. The corresponding p-value are described as follow: $*<0.05$, $**<0.01$ and $***<0.001$.

continuous cache-based model that the dynamic cache only stored n history encoder states closed to the current sentence. The result shows that the overall performance of theme-rheme information guided cache was significantly improved over the continuous cache.

Finally, we wanted to study the effect of cache size on our model. Figure 3 gives the change of the BLEU score on our model with different cache size. Intuitively, if we expanded the cache size, our model can obtain more context information and the contextual attention layer will filter the unrelated information. On the contrary, the results show that excessive cache size will reduce the performance of our model. This indicates that a larger cache size allows model to store more context information and each state only related to few context information while others become noises.

5.3 Qualitative Analysis

In this part, as shown in Table 6, we give three examples to illustrate the improvement brought by our model. Example 1 shows the inconsistent translation of synonyms that produced by baseline model. Concretely, the most frequent translation of “总统 卡特” in the same document is “president carter”. But the baseline model translated “卡特” into “jimmy carter” in current sentence. Example 2 shows the under translation of topic word that produced by baseline model. In this example, “法国” is a topic word of the document, which appears many times in other sentences. The baseline model translated “法国 内政部” into “the interior ministry”, in which the word “french” was lost. As shown in Example 3, baseline model made mistranslation of the topic words. Concretely, “美国” is a topic word of the given document and was been translated into “the united states” correctly in other sentences. However, the baseline model incorrectly translated “美国” into “china” when translating current sentence. While our

#Example 1	
Src	美国 前 总统 卡特 在 民主党 总统 候选人 即将 在 爱阿华州 党 内 初选 首 度 对决 前夕 ,
Ref	former u.s. **president carter** praised democratic presidential candidate dean , a former vermont governor ,
Baseline	former us **president** jimmy carter , on the eve of the first confrontation between the democratic party's presidential candidate and former state governor of vermont ,
Our model	former us **president carter** , on the eve of the first confrontation between the democratic party's presidential candidate and former state governor of vermont ,
#Example 2	
Src	法国 内政部 发布 的 公报 说 , 截至 21日 中午 12 时 , 选民 的 投票率 是 21.41% , 低于 1995年 总统 选举 第一 轮 投票 时 22.52% 的 投票率 。
Ref	according to a communique issued by the **french interior ministry** , as of 12 noon on the 21st , the voters' turnout is 21.4%, lower than the 22.52% voters ' turnout at the first round of the presidential election in 1995 .
Baseline	the voter turnout rate was <unk> percent , lower than the <unk> percent in the first round of voting in 1995 , **the interior ministry said** .
Our model	according to the **french interior ministry** , the voter turnout rate was <unk> percent , lower than the voting rate in the first round of the 1995 presidential election .
#Example 3	
Src	他 指出 , 格 美 两 国 在 军事 领域 加强 合作 并不 是 针对 俄罗斯 的 。
Ref	he pointed out that the strengthened cooperation between **the united states** and georgia is not directed at russia .
Baseline	he pointed out : strengthening cooperation between georgia and china in the military field is not directed against russia .
Our model	he pointed out : strengthening cooperation between georgia and **the united states** in the military field is not directed against russia .

Table 6: Three translation examples that generated by our document-level NMT model, all sentences were segmented and lowercased.

model produced correct and better translation in these cases. These examples prove the coherence and cohesion of translation have been improved by our model.

6 Conclusion and Future Work

In this paper, we proposed a document-level NMT model with a dynamic cache guided by the theme-rheme information. By recognizing topic themes and giving them higher weights, the dynamic cache will keep the important history encoder states as contextual information. Then we used a contextual attention layer to integrate the cached context into the NMT model decoder. The experiment results showed that our model has achieved significant improvements over the state-of-the-art baseline model.

However, the proposed cache-based model only uses the theme-rheme information to capture linguistic information for translation. In the future, we would like to integrate thematic progression pattern and other linguistic knowledge into document-level NMT models.

Acknowledgements

The authors would like to thank Siming Shan and three anonymous reviewers for their comments on this paper. This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 61573294 and 62076211, in part by the Joint Research Funds of Ministry of Education of China and China Mobile under Grant No. MCM20180402, and in part by the Fundamental Research Funds for the Central Universities under Grant 20720181002.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 909–919.

Michael Halliday, Christian MIM Matthiessen, and Christian Matthiessen. 2014. *An introduction to functional grammar*. Routledge.

Christian Hardmeier. 2012. Discourse in statistical machine translation. a survey and a case study. *Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics*, (11).

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Xiaomian Kang, Chengqing Zong, and Nianwen Xue. 2019. A survey of discourse representations for chinese discourse annotation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):1–25.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Shaohui Kuang and Deyi Xiong. 2018. Fusing recency into neural machine translation with an inter-sentence gate model. *arXiv preprint arXiv:1806.04466*.

Roland Kuhn and Renato De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 12(6):570–583.

Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 899–907.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Elman Mansimov, Gábor Melis, and Lei Yu. 2020. Capturing document context inside sentence-level neural machine translation models with self-training. *arXiv preprint arXiv:2003.05259*.

Sameen Maruf and Gholamreza Haffari. 2017. Document context neural machine translation with memory networks. *arXiv preprint arXiv:1711.03688*.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2019. A survey on document-level machine translation: Methods and evaluation.

Thomas Meyer and Bonnie Webber. 2013. Implicitation of discourse connectives in (machine) translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 19–26.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. *arXiv preprint arXiv:1809.01576*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.

Eleni Miltsakaki, Rashmi Prasad, Aravind K Joshi, and Bonnie L Webber. 2004. The penn discourse treebank. In *LREC*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Karin Sim Smith. 2017. On integrating discourse in machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 110–121.

Jinsong Su, Shan Wu, Deyi Xiong, Yaojie Lu, Xianpei Han, and Biao Zhang. 2018. Variational recurrent neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*, 5:87–99.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. *arXiv preprint arXiv:1805.10163*.

Tian Wang and Kyunghyun Cho. 2015. Larger-context language modelling. *arXiv preprint arXiv:1511.03729*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Xue-feng Xi and Guodong Zhou. 2017. Building a chinese discourse topic corpus with a micro-topic scheme based on theme-rheme theory. *Big Data Analytics*, 2(1):9.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. *arXiv preprint arXiv:1806.05626*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. *arXiv preprint arXiv:1810.03581*.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2020. Toward making the most of context in neural machine translation. *arXiv preprint arXiv:2002.07982*.