Document-Level Machine Translation with Hierarchical Attention

A Project Report

Presented to Prof. Chris Pollett

Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements of the Class

Fall 2022: CS 297

By

Yu-Tang Shen

December 2022

Abstract

Machine translation (MT) aims to translate texts with minimal human involvement. MT has evolved to generate more comprehensive outputs with less human editing, and the utilization of machine learning methods is pivotal to its success. The purpose of this report was to gain insights into MT technologies in hopes of understanding the shortcomings and strengths of different approaches. This report surveyed and showed experiments on MT technologies, including rule-based MT, statistical MT, neural MT, and neural MT with attention mechanism.

Keywords **–** Attention model, machine translation (MT), neural machine translation (NMT), rule-based machine translation (RBMT), statistical machine translation (SMT)

TABLE OF CONTENTS

I. INTRODUCTION

Language barriers have been less significant since machine translation technologies have bridged the gap between people using different languages. Machine translation (MT) has been evolving to produce a more natural translation, so applications like Google Translate and YouTube translated subtitles have become critical for people to understand content in foreign languages. Knowing MT plays an important role in people's lives, this project aims to extend the current technologies from translating sentences and paragraphs to producing coherent translations of documents.

Rule-based models are the rudimentary techniques used in MT. They form a 1-to-1 table by mapping all words to their corresponding translations, but that can result in incoherent translations when a word has various meanings under different contexts. Statistical machine translation (SMT) seeks to resolve this issue, by translating words with probability aligning to word usage. Although SMT provides a solution to translate languages without specifying every rule between languages, the complicated relationship between languages cannot be well captured with SMT.

As neural networks became a powerful tool across different domains, neural machine translation (NMT) was also deployed to the machine translation field, by utilizing neural networks to incorporate the context to determine the most reasonable translation. Yang, Wang, and Chu [1] and Felix [2] showed that NMT models have grown, have improved the translation quality, and have become the pillar of MT.

We now discuss the organization of this report: this report aimed to explore the path of machine translation history, where four machine translation techniques, including rule-based MT, statistical MT, neural MT, and attention-based neural MT, would be studied and experimented with. Each of the technologies was listed as a deliverable for this report, presented in Sections III to VI.

II. Machine Translation Types

Before the discussion on the deliverables, classifications of MT techniques would be presented. All machine translation technologies can be categorized into the following three types or mixtures of them. The three types of MT, direct translation, indirect translation, and transfer translation, describe how source languages (SL) are mapped to target languages (TL).

Direct translation maps SL directly to TL, so the relationship between translations can be easily observed. However, the disadvantage of this strategy is the exponential growth of the set of relationships: for translation between N languages, $N(N-1)$ sets of relationships were required.

Indirect translation adds an additional interlingua (IL) layer between two languages so that fewer sets of rules are required for multi-language MT. Instead of translating SL to TL directly, SL was first mapped to IL, which preserved the semantic information, and TL was generated from IL information. With this approach, merely $2N$ sets of rules are required for translation between N languages.

Transfer translation attaches an extra layer of IL from the indirect translation schema, so the two layers of IL act as abstractions of SL and TL respectively. This approach resolves the challenge of synthesizing different TL with the same piece of IL. Instead of synthesizing TL from the abstraction of SL, an additional transition to convert abstracted information from SL to TL was deployed.

III. Rule-Based Machine Translation

The goal for the first deliverable was to understand one of the most apparent ways of implementing machine translation: manually describing the transition between languages, namely, Rule-based machine translation (RBMT). Although the concept of MT was brought up in the seventeenth century, it was until 1933 that Artsrouni and Sminrnov-Troyanskii published a concrete proposal utilizing a paper tape machine for translation, which could be categorized as RBMT [3].

RBMT models follow a set of rules to perform translation, and the difference between swapping every word in the SL into TL was more complex rules, such as re-ordering, could be specified in RBMT. The advantage of RBMT is the results are deterministic and predictable: once the rules were set, the outputs are determined. This makes the rules calibration easier: one can trace the rules that produce unexpected results and make changes accordingly.

The experiments aimed to perform translations on sentences that had simple structures, and Universal Rule-Based Machine Translation toolkit [4] was utilized.

Sentences that had one-to-one mapping between English and Chinese, as shown in Fig. 2, were tested. Since the relationships between the two languages were straightforward, once the rules were properly configured, the translation would be correct.

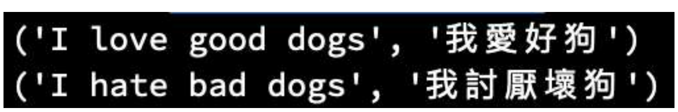

Fig 1. Translations from English to Chinese

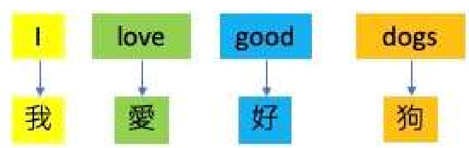

Fig 2. One-to-one relationships without reordering required

With successful results, experiments on sentences that contained words that could be ambiguous were wanted. In the two source sentences in Fig3., the two 'make's mean differently. Since the rules must be deterministic, RBMT failed to resolve different meanings in these two sentences.

```
('They make perfect steaks', '他們做完美的牛排')
('They make a cute couple', '他們做一可愛的情侶')
```

Fig 3. Tuples of translations from English to Traditional Chinese

RBMT showcased the practicality of translating with machines. Despite its inability to provide more than one translation from the same words, it provided users with a general idea of the information written in languages one didn't understand.

RBMT generates deterministic translations once the rules were set, and that can be a beneficial property in MT systems, where developers could easily adjust the system for desired effects. On the other hand, determinism also hindered the ability of the system to provide proper translations under different contexts.

IV. Statistical Machine Translation

In this deliverable, we look at statistical machine translation (SMT), which aimed to provide flexibility to translations and resolve the problems seen in RBMT. While RBMT was thriving from the 1930s to the 1950s, Weaver proposed the idea of utilizing statistics in languages to produce translations [3]. Although RBMT provides satisfying translation, it requires complex rule definition, pre-editing, and post-editing. In contrast, SMT simply looks at the relationship between the statistics of SL and the probability of all the possible candidates in TL. With SMT, programmers didn't need to define every single rule between SL and TL, instead, the task became gathering an abundant amount of data to obtain an unbiased statistic for every word.

SMT tasks are described in the following equation, which aims to find the best candidate $\hat{t}$ for a given source language vocabulary s from the TL distribution T.

$$\hat{t} = argmax_{t \in T} P(t|s)$$

In the formula above, there were various implementations of the function P, such as Charniak's model [5], greedy algorithm, etc. Charniak developed an English parser that describes the probability of a specific word being which part of speech [6], i.e., the parser informs users about the probability of a sentence being legitimate. Therefore, the translation model from Charniak not only considers the best candidate from the statistics of SL and TL, but it also utilizes the parser to eliminate those candidates that would decrease the sentences' validity.

The greedy approach for SMT is relatively simple: it chooses the best candidate based on its current options and disregarded all other information. A greedy SMT pseudo-code was presented:

```
1  translate(S)
2   result = φ
3   for s in S
4     candidates = getCandidate(s)
5     bestCandidate, bestScore = φ, -1
6     for candidate in candidates
7       if(probability(s, candidate) > bestScore)
8         bestCandidate = candidate
9         bestScore = probability(s, candidate)
10      end if
11    end for
12    result = result + bestCandidate
13  end for
14  return result
```

The `getCandidate` function on line 4 retrieves all the possible translations for source language vocabulary `s`, and the `probability` function on lines 7 and 9 returns the probability of the first argument being translated into the second argument. The `translate` function translates all the vocabulary in the SL sentence into TL based on the most popular translation and joined all the TL vocabulary together.

In the next experiment, a more complicated version of greedy SMT was tested, where the translating function could be abstracted as:

```
1  translate(S)
2      result := get_current_translation(S)
3      for i in S.length
4          //maximize the probability on both SL-TL mapping and TL usage
5          result[i] := argmax_prob(result[i-1], word, result[i+1], dictionary)
6      end for
7      for i in result.length
8          // swap if two TL words gives a higher probability when flipped
9          if(TL_prob(result[i] + result[i+1]) < TL_prob(result[i+1] + result[i])
10             result[i], result[i+1] := result[i+1], result[i]
11          end if
12     end for
13     return result
```

In line 4, the `argmax_prob` function can be described as $argmax_{t_i \in T} P(t_i|s_i) + P(t_{i-1}, t_i, t_{i+1})$, where the former term in the formula, $P(t_i|s_i)$, indicates the probability of an SL word s_i being translated into the TL word t_i, and the latter term, $P(t_{i-1}, t_i, t_{i+1})$, defines how probable [t_{i-1}, t_i, t_{i+1}] was in TL usage (where t_{i-1} is the TL word before t_i, and t_{i+1} is the TL word after t_i). To address the ordering problem, the program examined if swapping items in bigrams increased the bigram probability in TL usage in the loop in line 7.

Fig. 4. SMT results

The function `translate` was repeated multiple times until the probability stopped increasing. Despite the accuracy, which was calculated by the number of matched words divided by the length of the sentence without considering synonyms or any other information, shown in Fig.4 was within the range of 32% to 0%, the translation was comprehendible.

V. NEURAL MACHINE TRANSLATION

The purpose of this deliverable was to perform experiments on neural machine translation (NMT) to gain fundamental knowledge about sequence-to-sequence (seq2seq) models. As neural networks became dominant in various fields such as visual object detection, trend prediction, etc., researchers tried to utilize neural networks to further improve machine translation. Although neural networks have presented excelling capabilities, the nature of neural networks might contradict some tasks. Different from trend prediction and object detection, machine translation has a variable length in both input and output. The input shape of a machine translation project can vary from 5 (a short sentence) to 500 (a paragraph); moreover, the output shape is also indeterministic: two sentences having the same shape can have different lengths when translated into TL.

Without the capability to consume variable-sized input and generate outputs accordingly, it will constrain MT applications to be flexible enough to become practical. To enable neural networks to consume and output variable-shaped sentences, Hochreiter [7] designed a neural network that can generate translations in different shapes as well as consume variable input shapes.

Hochreiter [7] presented a design that allows neural networks to produce variable-shaped outputs with two long short-term memory (LSTM) neural networks. Since the input and output shapes were variable, those having such properties are later referred to as seq2seq models. LSTMs are similar to recurrent neural networks (RNNs), but they solve the vanishing gradient problem [8] that is often found in RNNs. The inputs were first preprocessed by padding empty strings to the maximum length that this application aims to take so all the inputs had equivalent lengths without introducing significant bias. The first LSTM summarized the input sentences, producing a vector that acted as activations for the next LSTM. The second LSTM then took the summarized vector and produced the translation until it generated a special token <EOS>, abbreviated from “end of sentence”.

A dual LSTM setup was implemented, aiming to translate English sentences into Chinese from UM-corpus [9]. The model was designed with two LSTMs, one for encoding information in English sentences while the other one decoded the information. To reduce the workload on hardware, an embedding layer was introduced to first shrink the input size from 15k to 1k. With a reduced size input, the encoder LSTM then summarized the embeddings, and the decoder LSTM eventually generated corresponding output from the summary given by the encoder LSTM.

With the embedding layer generating a 1024-long vector and configuring both encoder and decoder LSTM to be the shape of 128 after 3000 epochs of training, the model gave the accuracy as shown in Fig. 5. The training accuracy kept increasing, but the validation accuracy essentially remained the same. Therefore, the current configuration might not improve further with more training epochs.

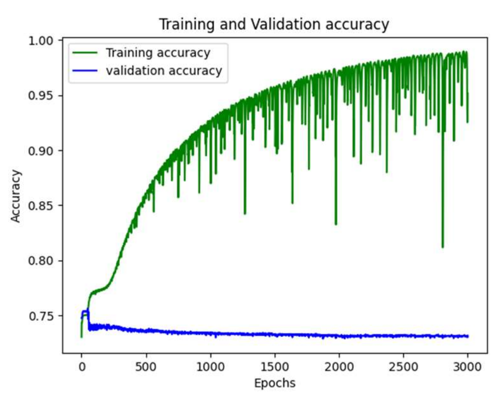

Fig. 5. Training and validation accuracy to number of epochs

Looking at the translations from the model, some sentences were translated properly while some were not shown in Fig 7. The first translation completely failed, where the decoder couldn't properly generate any meaningful words but kept repeating the word "的". The reason for getting this result was the encoding for the word "的" was close to the untrained model output (shown in Fig 6.), i.e., it could be viewed because of not being trained at all. The second sentence improved on its sentence structure, where less stuttering was observed, but the translation didn't match the meaning of the original sentence. The third translation had a high quality: the model translated the meaning correctly, produced a valid sentence structure, and the difference between the ground truth (时事新闻节目) and the output (新闻报导) was acceptable as they were synonyms. With these three sentences, a gradient of how this model could learn was presented.

```
English sentence:
        She had spent so much of her life feeling secondary ( though to a man she adored ) that she burned to undo the laws that hemmed women in . She was only an ordinary woman , she like
d to say .
Correct translation:
        在她一生的大部分时间中，她总感觉自己作为女性得不到重视（虽然她受到一个男人的宠爱），于是尽自己所能解除那些禁锢女性的法律。
Translated as:
        他，，的，的的的，热油烫伤，的的，，，，，的的，，，，，，，，，，，的的的。。
```

Fig 6. initial translation without training

```
English sentence:
        Even Amazon , an online shopping giant , may feel the urge to splash out and buy , perhaps , Sony-Ericsson , which like Motorola seems too small to make it on its own .
Correct translation:
        甚至连网购业巨头亚马逊也迫不及待地想进来，它可能会收购索尼爱立信，这家公司与摩托罗拉一样，已经萎缩到无法支撑下去。
Translated as:
        自参议院拉加，的的的，，，，，，，，，，，飞机，，，，，电锯万份肖像传输放。
English sentence:
        So , something at 30 decibels would be pretty faint .
Correct translation:
        因此,30分贝的声音相当微弱。
Translated as:
        乍一看，，是不用费时既的的指头。
English sentence:
        This is IN THE NEWS in VOA Special English .
Correct translation:
        这里是美国之音慢速英语时事新闻节目。
Translated as:
        这里是美国之音慢速英语新闻报道。。
English sentence:
        This is English as a Second Language Podcast episode 443 .
Correct translation:
        这是ESL博客第443篇。
Translated as:
        我是 你觉得很成功。
English sentence:
        IF THERE is any endeavour whose fruits should be freely available , that endeavour is surely publicly financed science .
Correct translation:
        如果有人试图把成果免费公开的话，那他一定是科学资助者，
Translated as:
        当合法买单的的的，授权的的的的年长的的的罪行。
```

Fig 7. translation after 3000 epochs

VI. BASELINE ATTENTION-BASED MACHINE TRANSLATOR

The last deliverable focused on the attention mechanism, which would be the main technology used in CS 298 project. LSTMs address the vanishing gradient problem [8], but the vanishing effect remains noticeable when the distance between two related items in a sequence is beyond the capability of the memory. Bahdanau et al. [10] introduced the attention mechanism to mimic human translators: cutting a long sentence into smaller fragments and processing each of them. Instead of trying to memorize all the necessary context, the attention mechanism scanned through the passage and looked for the related terms, mimicking human translators paying attention to the keywords in passages. “Attention” described how related two items were. For instance, the term “device” had high attention with “computer” could imply “device” was referring to “computer” in a sentence.

Attention mechanism applied three different information extraction functions to each component in the input sequence: query, key, and value, where all three functions were linear transformations of the input value as follows:

Query: $q_i = W_q a_i$ where W were trainable weights for different functions

Key: $k_i = W_k a_i$ and a were inputs to be parsed

Value: $v_i = W_v a_i$

With all *q, k,* and *v* values for each component in the sequence, the attention score between components is computed as $\alpha_{1,2} = q_1 \cdot k_2$, where $\alpha_{1,2}$ is the attention score between the first and second items. All *q* and *k* permutations are multiplied to obtain all attention scores, and all the attention scores will be normalized and multiplied to all corresponding *v*. The output of an attention layer is $softmax(QK^T)V$, where Q, V are matrices of *q* and *v* values, and K^T is the transpose of *k* values.

Ideally, performing attention to all components in sequences would preserve all information embedded between them, but doing so would require a great number of computing resources. Calculating attention between components in an n-length sequence would require n^2 computations. Thus, attention models such as Transformer [11], BERT [12], etc., had maximum input length limitations such that the models could be efficient and powerful.

Manzil et al. [13] presented a pattern for performing attention: instead of computing attention to all permutations, compute attention for those i) were close to each other, ii) were the first or second component, and iii) were randomly selected, as shown in Fig. 8. It demonstrated similar, some even better, performance compared to other attention models.

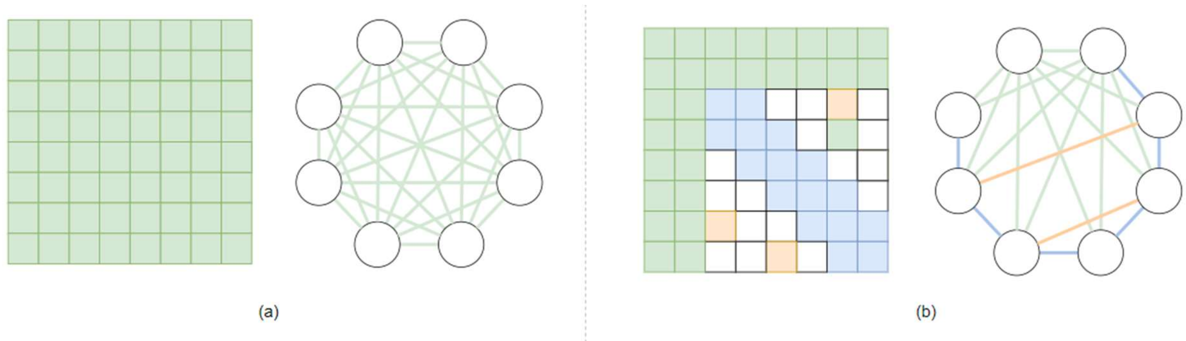

Fig. 8. (a) Full attention compared to (b) Big Bird attention

In the experiment, the Transformer model obtained a BLEU score of 10.45 after training for 30 epochs. It was much lower than the original paper listed, where it achieved a 41 BLEU score on English to French translation. The original paper trained with 6 encoder layers and 6 decoder layers on 45 million pairs of sentences, while the experiment conducted in this report had 1 encoder layer and 1 decoder layer trained on 130 thousand pairs of sentences. However, some intuitive results would be shown to provide insights into the attention model. Fig 9. showed that the model could learn phonetics ("Diego" was translated in an acceptable but different way), semantics ("can", "how", and "find out" were all translated into synonyms), and sentence structures (the model successfully attend 2005 to tokens that were in the front and put it in the front).

```
[Source]: diego had to get off at the same stop .
[Prediction]: 迪戈必须在同一个停止停止 。
[Original]: 蒂亚哥也在同一个车站下了车 。

[Source]: do you think he can get down from the tree ?
[Prediction]: 你觉得他可以从树上下来 ?
[Original]: 你认为他能从树上下来吗 ?

[Source]: people love to find out " how " to do things .
[Prediction]: 人们喜欢找出 " 怎么做 " 的事情 。
[Original]: 人们都喜欢了解 " 如何 " 做事情 。

[Source]: statistical data from the chinese authorities shows that bilateral
          trade between portugal and china in the first half of 2005 totaled us$908 million .
[Prediction]: 统计数据显示 , 2005年上半年葡萄牙和中国当局之间的贸易数据显示 ,
          2005年上半年增长9.08亿美元 。
[Original]: 中国官方的统计数字显示 , 2005年上半年的中葡双边贸易额为9.08亿美元 。

[Source]: senior u . s . and pakistani military leaders met this week on
          an american aircraft carrier to discuss the violence .
[Prediction]: 美国和巴基斯坦军事领导人本星期在美国飞机上讨论了暴力活动 。
[Original]: 美国和巴基斯坦军方高级领导人本周在美国一艘航空母舰上讨论了暴力问题 。
```

Fig. 9. Acceptable results from the Transformer model

VII. Conclusion

From Section II to Section VI, various machine translation techniques were reviewed. It strengthened the belief that document-level MT was needed because it required an impractical amount of computing power with current technologies to perform document-level MT. During the experiments completed in Sections V and VI, compromises were needed due to insufficient computing power.

Besides the proposed hierarchical attention, the sparse attention mechanism [13] indicated another path for achieving document-level MT: as sparse attention models required the number of attention linear to the length of the input size, it could handle inputs that were quadratically longer than models like Transformers. Common configurations for Transformers had an input length limitation of 1024-long, with sparse attention, the limitation could be increased to 2^{20}-long, which should be sufficient for most documents. If sparse attention itself was insufficient to obtain acceptable performance, the concept could be applied to hierarchical attention. The proposed hierarchical attention model was to first apply attention mechanisms to sentences or paragraphs then apply attention to abstracted information from the first attention. References across multiple paragraphs were possible but most likely less often, and therefore sparse attention could be applied on the second level of attention.

The conducted studies and experiments reinforced my sequence processing skills and introduced useful libraries with which I was not familiar. Older technologies such as RBMT and SMT were explored in Deliverables 1 and 2, which involved a great deal of sequence processing, especially regular expression, skills. Although the data planned to be used in the final project might be already preprocessed, regex skills would still be handy to fix small defects in the dataset. In Deliverables 3 and 4, several libraries, such as SentencePiece [14], SacreBLEU [15], fairseq [16], etc., were used. Those libraries would be helpful in prototyping and proving concepts.

REFERENCES

[1] S. Yang, Y. Wang and X. Chu, *A Survey of Deep Learning Techniques for Neural Machine Translation,* arXiv, 2020.

[2] F. Stahlberg, *Neural Machine Translation: A Review and Survey,* arXiv, 2019.

[3] W. J. Hutchins, "Machine translation: A brief history," in *Concise history of the language sciences*, Elsevier, 1995, pp. 431-445.

[4] T.-P. Nguyen, *URBANS: Universal Rule-Based Machine Translation NLP toolkit,* GitHub, 2021.

[5] E. Charniak, K. Knight and K. Yamada, "Syntax-based language models for statistical machine translation," in *Proceedings of Machine Translation Summit IX: Papers*, 2003.

[6] E. Charniak, "Statistical Techniques for Natural Language Parsing," *AI Magazine,* 1997.

[7] I. Sutskever, V. Oriol and Q. V. Le, *Sequence to Sequence Learning with Neural Networks,* arXiv, 2014.

[8] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 6, no. 02, pp. 107-116, 1998.

[9] L. Tian, D. F. Wong, L. S. Chao, P. Quaresma, F. Oliveira, Y. Lu, S. Li, Y. Wang and L. Wang, "UM-Corpus: A Large English-Chinese Parallel Corpus for Statistical Machine Translation," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[10] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, pp. 11-21, 2017.

[12] D. Jacob, C. Ming-Wei, L. Kenton and T. Kristina, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 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)*, 2019.

[13] Z. Manzil, G. Guru, D. Avinava, A. Joshua, A. Chris, O. Santiago, P. Philip, R. Anirudh, W. Qifan, Y. Li and A. Amr, "Big Bird: Transformers for Longer Sequences," *arXiv,* 2020.

[14] T. Kudo and J. Richardson, *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing,* arXiv, 2018.

[15] M. Post, "A Call for Clarity in Reporting BLEU Scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, 2018.

[16] O. Myle, E. Sergey, B. Alexei, F. Angela, G. Sam, N. Nathan, G. David and A. Michael, "fairseq: A Fast, Extensible Toolkit for Sequence Modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*,2019.