Investigation into the Impact of Tokenization, Model Architecture and Word Embeddings on Neural Machine Translation

A0239298L, A0250934E, A0239754R, A0276597E, A0234622M, A0218236H
Group 42
Mentored by Tan Yong-Jia, Naaman
{e0773896,e0949099,e0774352,e1132330,
e0726622,e0544272}@u.nus.edu

Abstract

Neural Machine Translation (NMT) has become the dominant approach for automated translation between languages in recent years. However, most papers focus on novel model architectures to perform the task and it is accepted that certain architectures can achieve better performance on this task.

In this paper, we attempt to partition the translation task into smaller sub-components, and qualitatively explore various approaches to the pre-processing and representations of texts and their effects on downstream translation performance. We evaluate various approaches to tokenization and text embedding representations and their effects on model performance over various input text lengths. Finally, we also explore if attention-like mechanisms from newer model architectures such as Transformers can bring the performance of reference architectures such as RNNs above newer architectures such as Transformers and LSTMs.

We prove that certain tokenization approaches perform better than others, with pre-training being a decisive factor. We also learn that raw embedding approaches work better than frozen pre-trained embeddings due to their learning ability, and the effects of replacing the raw embedding layer with frozen pre-trained embeddings degrade model performance due to the depth of the model architecture networks. Lastly, we demonstrate that attention-like mechanisms allow RNNs to outperform LSTMs on the translation task and bring it closer to Transformers in terms of performance.

1. Introduction

Machine translation has two main approaches: statistical machine translation (SMT) and neural machine translation (NMT). While SMT has been a foundational approach for decades, NMT is a relatively recent approach that was only introduced a decade ago (Kalchbrenner and Blunsom, 2013; Cho et al., 2014a; Sutskever et al., 2014; Bahdanau et al., 2015), made possible by advances in deep learning. The continued advancements in the areas have catalysed the development of new tools and algorithms that allow for highly customisable translation frameworks.

NMT operates through several distinct stages that will influence the final translation quality. These stages include tokenization, embedding, translation by the model architecture, and decoding. Each of these components offer various implementation options that will affect performance in different ways. Our project aims to explore three of these variables:

- Tokenization
- Model architecture
- Embedding

By systematically varying these while keeping other variables constant, we aim to identify the optimal combination for English-Chinese translation for the IWSLT2017-en-zh dataset.

Our code repository is hosted at this [link].

2. Related Work/Background

Tokenization

Tokenization plays an important role in the preprocessing phase of neural machine translation. The effectiveness with which raw text is separated into tokens and converted into numerical formats directly influences how neural networks learn and interpret underlying semantic structures. Fundamentally, there are three types of tokenization: character-based, subword-based, and word-based. Subword tokenization is the prevalent method of tokenization (Wolleb et al., 2024) due to its balance between semantic richness and vocabulary size management. It captures more meaningful semantic information than character-based tokenization and addresses out-of-vocabulary issues more effectively than word-based tokenization by breaking down unknown words into recognizable subunits.

Byte-Pair Encoding (BPE) and WordPiece are similar subword-based tokenization methods that incrementally build the vocabulary by merging pairs of characters or symbols in the training corpus. Their main difference lies in the criteria for merging – BPE (Sennrich et al., 2016) merges the most frequently occurring pairs of characters or subwords, while WordPiece (Wu et al., 2016) merges pairs based on a criterion that maximises the likelihood of the training data given the vocabulary. Other than training these algorithms from scratch, there are also many popular ready-to-use tokenizers that have been pre trained on large and diverse corpora, such as spaCy and Stanza. These tokenizers may be

rule-based or utilise machine learning models to handle a variety of languages and linguistic contexts.

Domingo et al. explored the use of different tokenizers for five different language pairs, and found that there is no one tokenizer that yields the best results for every language pair and language pair direction. To investigate the effect of different tokenization methods, we will be employing three variations of tokenizers to observe their performance for our language pair of English-Chinese translation.

Architectures

Using deep learning models is the common approach to the machine translation problem. The article Recurrent Continuous Translation Model(Kalchbrenner, et al, 2013) was the first to pioneer the idea of generating a vector representation of a sentence, but it was only in the article Sequence to Sequence Learning with Neural Networks (Sutskever et al, 2014) that took this idea and applied it to the encoder and decoder architecture with LSTM achieving good results of BLEU score of 34.8 on the entire test set and also performing well on long sentences. In the article Neural Machine Translation by Jointly Learning to Align and Translate (Dzmitry et al., 2014), the authors further further improve on RNN performance by implementing an automatica search for parts of a source sentence that are relevant to predicting a target word, essentially adding an attention mechanism to the RNN model.

Word Embeddings

Numerous studies have investigated word embeddings in the context of neural machine translation. Hill et al. (2020) found that translation-based embeddings had desirable properties over monolingual embeddings due to the pressure exerted on the embeddings by the translation objective. Translation-based embeddings were better at modelling word similarity and lexical function while monolingual embeddings are better at modelling non-specific inter-word relatedness. Qi et al. (2018) explored the different contexts where pre-trained word embeddings were particularly useful in machine translation. They found that pre-trained embeddings, particularly for the source language, improved the model's BLEU score. Additionally, for languages with high similarity, it was observed that pre-trained embeddings reaped more gains.

Against the conventional notions of the semantic importance of word embeddings, Uri et al. (2021) proposed a neural machine translation model with embeddings replaced by simple byte-level (Unicode) representation of each character. With a vocab size of 256 (representing the one-hot encoding of each Unicode token) and no preprocessing required, the byte-to-byte Transformer model achieved improvements in BLEU score over embedding-based models. It is of interest to note that removing trainable embedding layers is viable in byte token models as this is analogous to our tests where pre-trained embedding layers for a sub-word tokens model were frozen during training.

3. Corpus Analysis and Method

We first analysed a few samples of the training dataset to understand its characteristics.

In the dataset, we noticed some long sentences which contained long range dependencies. Shown in Table 1 below.

Source (English)	**Target (Chinese)**
Now, Conor did not come home one day and announce, "You know, hey, all this Romeo and Juliet stuff has been great, but I need to move into the next phase where I isolate you and I abuse you" — — "so I need to get you out of this apartment where the neighbours can hear you scream and out of this city where you have friends and family and coworkers who can see the bruises."	康纳并不是回到家, 向我宣布 "嘿, 虽然罗曼蒂克之类的很棒, 但是我们要进入下一阶段了 我要孤立你然后虐待你。" "所以我要你离开你自己的公寓, 防止你的邻居他听见你的惨叫, 我还要让你离开这个有你的朋友、家人和同事的城市不然他们会看到你的伤痕。"

Table 1. Sample of long sequence

In the given source sentence above, the main subjects ('Conor', 'You') have long range dependencies with some descriptors near the end of the sentence ('bruises, coworkers, friends and family). This could cause problems if the translation model is unable to attend to long contexts.

It is common knowledge that the sequential input nature and vanishing gradient problem of Recurrent Neural Network (RNN) models hinder their performance on longer sequences such as these, where the context of the subjects may be needed to perform a more accurate translation of descriptors and other terms near the end of the sentence. Hence, we propose exploring the addition of various mechanisms to improve the long-range context performance of RNN models and compare the model performance with reference LSTM and Transformer models.

In the dataset, we also noticed that sentences with shortest lengths of 0-20 take up a vast majority of the training dataset, shown in Table 2 below.

Sentence Lengths (en)	Counts	%
0-20	135177	58%
21-30	49862	22%
31-40	24622	11%
41-50	11367	5%
51-60	5122	2%
60+	5116	2%

Table 2. Distribution of Sentence Lengths in defined bins

The skewed nature of the training set towards shorter sequences may result in models trained on this dataset being less capable of handling longer sentences.

An even closer look at the dataset reveals many identical short sentences of “Thank you” and its variations. We hypothesise that this could result in our models being particularly adept at translating English sentences with the phrase “Thank you”. Shown in Table 3 below.

Frequent Sentences	Counts
“Thank you.”	57
“Thank you very much.”	15
“Thanks.”	3

Table 3. Counts of frequently occurring sentences

Lastly, we noticed some inaccurate representation between the source and reference translations in the test dataset. For example, the phrase “Hard is hard.” in English is a simple, direct statement emphasising the difficulty of something. However the reference translation “各有各的难处” carries a slightly different nuance. We believe this phrase translates more closely to “Everyone has his own difficulties”, according to Google Translate. This suggests that the current dataset may not fully capture certain nuances essential for accurate translation, a problem that could potentially affect the overall quality of our translation results. Such discrepancies possibly highlights the prevalent challenges in translation work, particularly in maintaining the literal meanings across languages.

4. Experiments

To determine which methods for each subdomain produce the best translation results, we investigate the following research questions:

RQ-1: Between pre-trained word tokenization algorithms, byte-pair encoding (BPE), and another subword tokenization algorithm, which approach gives the best tokenization performance of the dataset to give the best downstream translation result?

RQ-2: Given the common pitfalls of RNN-based architecture for Seq-to-Seq tasks, how would the performance of an advanced RNN variant with an ***alignment*** layer measure against LSTM and Transformer architectures?

RQ-3: How effective is transfer learning using pre-trained word embeddings, including both static and context-aware types, in improving the performance of machine translation tasks?

Experimental Setup

For both RQ-1 and RQ-2, all models were constrained to a maximum of 30 epochs. To mitigate overfitting, early stopping is enabled within this epoch limit. For each different configuration, we retained the model state corresponding to the minimum value of the used loss function on the validation dataset for evaluation purposes.

For **RQ-1**, we opted for the baseline RNN with alignment mechanism model as the control. We trained 3 separate instances of this model on a subset of 10,000 rows of the train dataset, changing only the tokenization strategy. The instances are as follows:

1. **Pre-Trained Tokenization**. We used spacy’s en_core_web_lg model (Explosion, 2023) for English, and Stanford’s Stanza (Peng, et al., 2020) for Chinese. This had vocabulary sizes of 62,716 tokens for English, and 101,686 tokens for Chinese.

2. **Byte-Pair Encoding (BPE)**. We trained a custom SentencePiece BPE model on our train dataset with 16,384 tokens for each language.

3. **WordPiece Tokenization**. We trained custom WordPiece tokenizers using configurations from bert-base-uncased (Devlin et al., 2018) for English and bert-base-chinese (Huggingface) for Chinese. These are both reference tokenizers for the BERT model architecture.

This approach is a middle ground between the configurations 1 and 2 above. We specified vocabulary sizes of 32,000 for both languages, but resulted in vocabulary size of 30,522 and 21,128 for English and Chinese respectively.

For **RQ-2**, we opted for the BPE tokenizer to keep the tokenization algorithm as the control, changing only the model architecture. This kept the embedding layer size of our models down so that they could fit on the training devices. We also trained the models on the entire training set of 231,266 rows, and restricted them to a maximum of 30 epochs with early stopping.

Using these settings, we evaluated the performance of our three model architectures:

1. RNN with alignment. Referencing Bahdanau et al. (2014), this uses a bidirectional RNN for input sentence processing and adds an **alignment** layer to a vanilla RNN model to improve its performance on longer sequences.

2. LSTM. Referencing Sutskever et al. (2014), this uses a multilayered LSTM to map the input sequence to a vector, and then another deep LSTM to decode the target sequence. This is a reference model for comparison against the RNN.

3. Transformer (reference). Referencing Vaswani et al. (2017) and their original Transformer implementation, we shrank the model's parameters to 256 dimensions from 512 for the model's internal representation, as well as the fully connected layers to 1024 dimensions from 2048 to allow the model to fit in memory. We also referenced the PyTorch implementation (Song, 2021) to convert it into code.

For **RQ-3**, we picked our Transformer model and BPE tokenizer as the control, swapping out only the embedding layer. We trained for a maximum of 20 epochs on the entire train dataset of 231,266 rows. The configurations we experimented with are as such:

1. Raw Embedding (RE). Our base implementation is a randomly initialised Embedding layer with updatable weights in training. Any context it learns is through backpropagation from the model output loss all the way to the attention heads and layers.

2. Context-Aware Embedding (CAE) Taking inspiration from BERT-NMT (Zhu et al., 2020), we opt to take only the BERT Embedding layer and use it as a source embedding layer, instead of fusing it with all of our attention layers. We do this to fit the model on our training device and also to observe the impacts of transfer learning. We pre-train 2 BERT models for each language from scratch on our training set for 20 epochs to use for this layer, and freeze it to reduce the trainable parameters of our model.

3. Static Embedding (SE). Building on the Word2Vec model, FastText (Bojanowski et al., 2016) utilises n-grams of characters within each word to generate embeddings. This utilises subword n-grams to generate embeddings for unseen words based on the morphology of new words. We train a FastText model for each language, for 5 epochs, on the train dataset and use it as an embedding layer. We also freeze this layer for control with the CAE setup.

---

5. Evaluation

RQ-1: Tokenization

We start off by evaluating various tokenization strategies, as shown in Table 4:

Type	Config	Description
Pretrained	1	Industry-standard pre-trained word-based tokenizers
Byte-Pair Encoding	2	SentencePiece BPE (for both languages)
WordPiece	3	WordPiece tokenizer using settings of pretrained models

Table. 4. Tokenization configurations and descriptions

We hypothesised that configuration 1 would have the best performance. All of the tokenizers' vocabularies were trained on the entire training set (231,266 rows). Given the learned vocabulary, each of the tokenizers will output a vocabulary index that maps to each input token.

To minimise training time for benchmarking purposes, the scores are based on a subset of 10,000 rows in the training set, and evaluated on the entire test set. We denote BERTSCORE as 'BERT', and Recall, Precision and F1 as 'R','P' and 'F' respectively. We refer to the tokenizers by their configuration shown in Table 5 below.

Evaluation Metrics	Tokenizer Configurations		
	1	2	3
BLEU-4	**4.08**	2.41	0.10
BERT-R	**0.54**	0.51	0.44
BERT-P	**0.56**	0.55	0.54
BERT-F	**0.55**	0.53	0.49

Table. 5. Evaluation scores from baseline RNN with different tokenization configurations, trained with subset of training set

The results confirm our hypothesis that configuration 1 would outperform the others, due to the pretrained tokenizers having better linguistic representation of the languages. The effectiveness of spaCy and Stanza, as underscored by their widespread adoption in the industry, makes them ideal choices as our default tokenizers. However, we opted not to use them in subsequent experiments due to their extensive vocabularies, which impose significant memory space constraints.

The more surprising aspect of the results, however, is the poor performance of configuration 3. Theoretically, configurations 2 and 3 are relatively similar - (i) both did not benefit from pre-training; (ii) BPE and WordPiece are also relatively similar subword tokenization algorithms, with the main difference being how tokens are added to the vocabulary - BPE chooses the highest frequency pairs while WordPiece chooses pairs that maximise the

likelihood of the dataset. We surmise that the poor performance of configuration 3 could be due to suboptimal compatibility of the tokenizer with the reference RNN-alignment model architecture. Since BERT tokenizers are designed for BERT, which is a bidirectional transformer model, BERT tokenizer settings might not align well with the sequential processing nature of RNNs.

Upon further investigation, we found that configuration 3's Chinese vocabulary lacked continuation tokens for Chinese characters, and contained a low percentage of Chinese tokens. In the original vocabulary of bert-base-chinese, there are initial tokens and continuation tokens for each of the 7,322 Chinese characters it contains (e.g. '非' which can be standalone or the initial segment of a word, and '##非' which would be a continuation of the previous token). This amounts to 69% of the original vocabulary being Chinese tokens. In contrast, configuration 3's Chinese vocabulary only contained initial tokens for the 4,815 Chinese characters it contains, which is only 29% of the vocabulary, indicating noise in the dataset that may have resulted in a less representative vocabulary. Continuation tokens help to preserve linguistic context, especially for a sequential model like RNN; hence, the lack of continuation tokens in configuration 3's Chinese vocabulary could be another reason for its poor performance as it likely leads to increased fragmentation of semantic information, affecting the model's ability to understand and translate.

The table at Appendix RQ-1-3 shows illustrative examples of tokenization by each configuration. We observe that the tokenization of English sentences are very similar, which could indicate that the Chinese tokenizer is the differentiating factor for performance. Configurations 1 and 2 are able to separate Chinese text into meaningful tokens, but configuration 2 runs into out-of-vocabulary instances. Configuration 3, however, separates Chinese text by character, which increases fragmentation of semantic information.

For further evaluation on candidate translations, we measured the diversity of each model's output by counting unique characters in the generated text. The results varied significantly: configuration 1 yielded translations with 131 unique characters, configuration 2 showed a higher diversity with 342 unique characters, and configuration 3 had only 4 unique characters.

Figure 1 below shows the BLEU-4 and BERT-F scores of each configuration by sentence length. Performance degraded considerably for longer sentences for all configurations. Configuration 1 consistently outperforms the rest across sentence lengths. While translations from configuration 1 are far from perfect, Appendix RQ-1-4 demonstrates that configuration 1 still has some semblance of meaning compared to the other configurations.

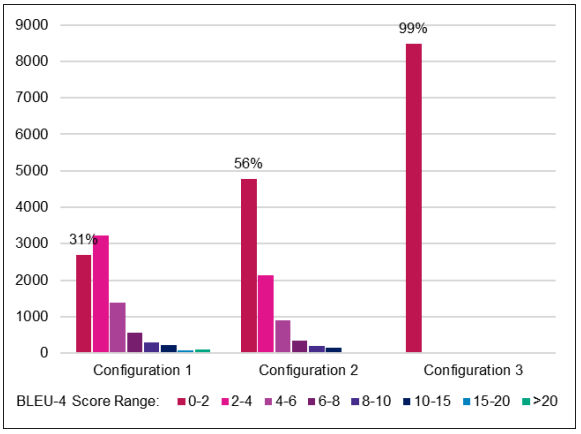


Figure. 1. BLEU-4 score distribution for each configuration

These specific results as well as the overall test set performance demonstrates that our word-based tokenization using spaCy and Stanza for English and Chinese respectively gives the best downstream results due in part to its vocabulary size and extensive vocabulary rules.

RQ-2: Model Architectures

To evaluate the various model architectures on the translation task, we decided to keep the tokenizer as the control, and vary only the model architectures. To keep training and fit our training device memory, we opted for configuration 2 with smaller output vocabulary size of 16,384 instead of configuration 1 to reduce the number of learnable parameters for the models.

In this section, we compare three established architectures in sequence modelling: an RNN-based architecture, with recurrent LSTM units, (LSTM) an advanced RNN variant with ***alignment mechanism*** (RNN) and a transformer-based model (T-MODEL).We assess their effectiveness through average BLEU scores across the full training set. In detail, we also examine the models' performance with short and long sentences through computing the BLEU scores by sentence lengths.

We hypothesised that the T-MODEL would achieve the highest overall effectiveness, consistently outperforming the other models across all sentence lengths. Additionally, we expected that the RNN, equipped with a bidirectional encoder in addition to the alignment mechanism, would surpass LSTM's performance across both short and long sentences.

Table 6 below displays the average BERT-based metrics and BLEU scores for all three architectures across the entire test set. The results support our initial assumption, demonstrating T-MODEL's effectiveness across BERT-F and BLEU scores for all sentence lengths. The results

also indicate a significant gap in performance between the RNN and LSTM models. Additionally, the calculation of average scores, categorised by sentence length bins reflected these trends. (Appendix RQ-2-1 to RQ-2-2).

Evaluation Metrics	RNN	LSTM	T-MODEL
BLEU-4	9.74	0.70	**13.78**
BERT-F	0.67	0.46	**0.73**

Table. 6. Evaluation scores from selected architectures with trained BPE tokenizer (config 2) , trained on full training set

Meso-Analysis

Plotting BLEU and BERT-F scores against sentence lengths individually (Appendix RQ-2-7), we note the general decline in BLEU scores as sentence length increased for all models. This decline was the most gradual with our T-MODEL, less so in our RNN, and most steep for LSTM. For the RNN and T-MODEL, the decline was consistent throughout the sentence lengths. The LSTM showed a sharp decrease from sentence lengths 2 to 20, with scores consistently remaining below 1 for sentences beyond this length. Regarding BERT-F scores, all three models exhibited a convergence towards the average score presented in Appendix RQ-2-7.

For further analysis, we handpicked some sentences representing short (0-20), medium (21-30), and long (60+) lengths for further analysis, noting that the bulk of our test dataset consisted of instances not exceeding 30 words in length.

For short sentences, the selected examples are shown and corresponding model outputs are in Appendix RQ-2-4. A qualitative look at the model outputs compared to the reference translations shows that T-MODEL outputs are more coherent and closer to the reference, while LSTM outputs are either incomplete or have significant errors. RNN outputs are generally more complete and coherent compared to those of LSTM.

It could be seen that the LSTM model could only perform partially well in translating the short sentences. From its translation of the second example in Appendix RQ-2-4, its output “我 试着” translates “And I tried”, the first three words of the source sentence. The poor performance of our LSTM demonstrates its limitations in handling short sentences as defined in our study. It is plausible that the LSTM model may exhibit better performance on a select few examples, particularly those comprising fewer than 10 words. A possible reason why the model performed poorly is insufficient LSTM layer size. Due to compute constraints, we utilised a 2-layer instead of 4-layer LSTM.

The narrow margin between the BERT-F scores of both RNN and T-MODEL suggests that the translations from both architectures were relatively similar in capturing the intended meaning. Essentially, despite the transformer model potentially being more accurate or fluent overall, the RNN was competitive in how well it preserved the semantics of the source text in its translations. For example, the translations from both models for the second example translated back to the same English sentence using Google Translate, “I tried some interesting things”.

For medium and long sentences, as demonstrated in Appendix RQ-2-5 and Appendix RQ-2-6, we will selectively examine examples from the RNN and T-MODEL translations, acknowledging the constraints our LSTM model demonstrated with shorter sentence lengths.

Compared to short sentences, there is an overall decrease in scores for both RNN and T-MODEL for both medium and long sentences. Additionally, we can see an increased gap in both BLEU and BERT-F scores between the two models, highlighting the RNN model’s limitations as the sentence length grows. Appendix RQ-2-5 reveals that while the RNN model captures essential elements in the first example of length 22, including phrases like "he was talking about," "smoke," and "poisoning him in his sleep," its ability to capture and translate information declines when dealing with the second example, which has a length of 30. While the RNN model could somewhat translate “because” , “active”, and “most frail”, its translation pales in comparison to T-MODEL’s output, which, though not fluent, captures most information, except for the location (mid-area), and translating ‘active’ to ‘积极’.

From Appendix RQ-2-6, with long sentences, we noted not only a loss of information in the RNN model’s outputs but also the emergence of repetitive patterns, which suggest difficulties in sustaining long-term dependencies. Although this might not be the sole factor, it is highly likely that the difficulty in sustaining long-term dependencies is a significant cause of these repetitions, highlighting the model’s failure to effectively recall and utilise previously processed information as the sequence lengthens. This is not surprising since the limitation of vanishing gradients is a fundamental challenge in the RNN architecture. Conversely, the T-MODEL outputs preserved coherency, as evidenced from both evaluation scores.

To visualise the attention within our RNN variant, we included attention plots for both the short and medium sentences in Appendix RQ-2-8. Although these plots lack sharp distinction – hinting at possible inaccuracies in the distribution of attention, another factor that may contribute to issues such as repetition – the overall data suggests that the attention mechanism still positively influences the

performance of our RNN variant, most notably up to a sentence length of 30, offering a performance advantage over the LSTM model. Despite these positive attributes, the RNN variant does not reach the performance benchmark set by T-MODEL, as evidenced by our comparative analysis of output quality and evaluation scores.

RQ-3: Transfer Learning via Word Embeddings

Word embeddings capture crucial semantic value for input words and can have a massive impact on a model's performance. Hence, we sought to investigate the impact of transfer learning via pre-trained word embeddings on the performance of an NMT model. The tests used the Transformer architecture as the control, as it is parallelizable and achieved faster training speeds on our device. Additionally, we kept the dimensions (n = 256) and maximum vocabulary size (n = 16,384) of the word embeddings constant. Pre-training for the embedding models was carried out for both Chinese and English tokens, using the SentencePiece model (configuration 2) from RQ1 for tokenization. We compare the performance of the following 3 configurations:

Type	Conf	Description
Raw Embedding	RE	Randomly initialised, values updated during training of model. Static, not context-aware.
Context-Aware Embedding	CAE	Pre-trained BERT on dataset, values frozen during training of model. Deeply context-aware.
Static Embedding	SE	Pre-trained FastText on dataset, values frozen during training of model. Static, not context-aware.

Table. 7. Description of the three configurations of word embeddings tested

We hypothesise that the model trained with CAE will perform better than SE as CAE will better represent the meaning of words in their specific usage. However, we postulate that the RE layer will perform the best out of all three as it is continually fine-tuned during the training of the model, thereby learning the nuances of the dataset as well as fitting to the transformer architecture.

We also hypothesised that transfer learning from pre-trained embeddings will allow the model to converge in less epochs. This is because the embedding representations are already learned, and the model in CAE and SE configurations has fewer parameters to learn.

Metric	RE	CAE	SE
BLEU-4	**13.8**	6.02	11.3
BERT-R	**0.71**	0.61	0.68
BERT-P	**0.75**	0.68	0.72
BERT-F	**0.73**	0.65	0.70

Table 8. Evaluated metrics of models trained in RE, CAE and SE configuration.

We found that the RE configuration achieved the best score across all four metrics out of the three configurations we tested.

This is likely due to the high specificity of the task of machine translation, where a deep understanding of linguistic patterns from both the source and target languages is required. The pre-trained embeddings for Chinese and English were trained separately, which means cross-language relationships in the embeddings could not be learned, especially as the embedded layers were frozen. The raw embeddings could pick these patterns up as the gradients flowed through the embedding layers of both languages during training. Furthermore, the trainable raw embedding layers meant that the RE configuration has more trainable parameters (16,384 * 256) and thus is a more expressive model.

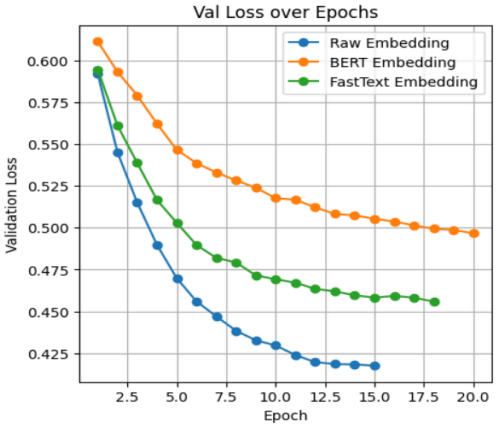


Figure. 2. Validation loss over training epochs. Number of epochs trained for RE = 15, SE = 18, CAE = 20

Conventionally, with transfer learning, a model is expected to converge at a faster rate as the model is not learning from scratch and can make use of the semantic nuances captured by the trained word embeddings. However, in our case, we did not reap the benefits of transfer learning. From Figure 2 above, the RE configuration's validation loss exhibited the fastest rate of convergence during training. This is likely also due to the fact that the CAE and SE configurations froze the value of the embedding layers during training.

As the Transformer model is relatively deep, we also posit that the transfer learning from the embeddings had limited utility due to the depth of the neural network reducing the effect of the embedding layer as well as the freezing of the embedding layers to keep the control between the context-aware hence difficult to train BERT CAE configuration and the FastText SE configuration.

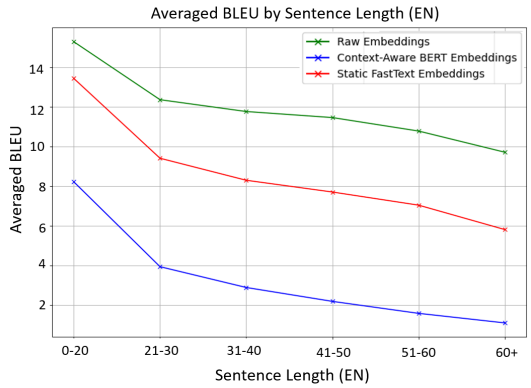


Figure. 3. Averaged BLEU scores by Sentence Length (EN)

Another surprising result we observed is that the CAE configuration performed worse compared to the SE configuration as seen in Figure 3 above. This could be attributed to limitations in the size of the training dataset. The context-aware embeddings were generated by a BERT model which was trained from scratch on only our dataset. The BERT model, which uses a Transformer architecture, has far more parameters compared to the FastText model which uses neural networks in Word2Vec. Thus, the BERT model struggled as it lacked enough data to learn generalised linguistic patterns without overfitting on irrelevant patterns in word context. On the other hand, the simpler FastText model had less capacity to overfit and produced more generalisable representations of the words.

Furthermore, the BERT architecture is not suited well for seq2seq tasks such as the translation task, and is better suited for Masked Prediction or Next Sentence Prediction. Hence, we posit that our BERT training procedure was inappropriate as it used the training loop for the Masked Language Modelling task, without the Masked Language Modelling model head output atop the BERT model. This could have limited the usefulness of the embeddings for our translation task.

6. Conclusion

In this study, we compared the effectiveness of different translation frameworks for English-Chinese translations. Based on our experiments in the three subdomains of tokenization, model architecture and word embedding, our best performing configuration was the pre-trained tokenizer, transformer architecture and raw embeddings. These results do not entirely concur with current state of the art findings, where subword tokenization (Sennrich et al., 2016) and pre-trained embeddings (Qi et al., 2018) are known to produce better performing translation models. This incongruence can be attributed to key differences in size of dataset, size of VRAM of training device and training parameters. This highlights the myriad of factors that contribute to the distilled metrics that machine learning papers tend to derive insights from. Our incongruent findings do not disprove scholarly wisdom in machine learning; rather, it shows that results for a particular experiment's configuration cannot be generalised, unless widely reproduced. An interesting parallel between knowledge discovery in academia and machine learning can be drawn here.

Limitations

One major obstacle for the project was the lack of available computation resources. Additional preprocessing steps such as the reduction of the vocabulary size was used to workaround this limitation. The effect of downscaling parameters was not explored and accounted for.

Furthermore, given compute constraints for the LSTM, it was not able to utilise sufficient model layers to learn the complex representations for longer sentences, hindering its performance for those. This in no way invalidates the better known performance of LSTMs on longer sequences, and we believe that if sufficient layers are added to the LSTM models, it will bring its performance closer to the Transformer model.

Future Extension

We recognize the importance of interdisciplinary collaboration and feel that insights from fields such as linguistics and sociology can be integrated into the project. For example, linguistic features such as POS Tagging and Named Entity Recognition can be extracted and added as an additional dimension for training. This can potentially help resolve ambiguities in certain translations, given Chinese's propensity for ambiguous translations of the same word in different contexts.

We also wish to scale the model architectures to their full sizes and train on larger datasets, so that the output models may be able to model and translate more complex data more accurately.

In addition, should sufficient compute be available for contextual embeddings to be fused into all of the Transformer Attention Layers, we may be able to see the full effects of our word tokenizer and embeddings in the model's performance.

7. Ethical Statement

We are committed to conducting our language translation project with ethical standards by striving to minimise bias, ensuring respect for cultural nuances, and transparency in our methodologies.

References

- Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Retrieved 14 April 2024 from https://aclanthology.org/D14-1179.pdf

- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

- Domingo, M., García-Martínez, M., Helle, A., Casacuberta, F., & Herranz, M. (2018). How Much Does Tokenization Affect Neural Machine Translation?. Retrieved Retrieved 15 April 2024 from https://arxiv.org/pdf/1812.08621.pdf

- Dzmitry, B., KyungHyun, C., Yoshua, .B. (2014) *Neural Machine Translation By Jointly Learning to Align and Translat*e Retrieved 12 April 2024 from https://arxiv.org/pdf/1409.0473.pdf

- Explosion (2023) *Release en_core_web_lg-3.7.1.* spacy. Retrieved 18 April, 2024 from https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-3.7.1

- Google-Bert (2024) *Google-Bert/Bert-Base-Chinese · Hugging Face*. Retrieved 18 april, 2024 from huggingface.co/google-bert/bert-base-chinese

- Hill, F., Cho, K., Jean, S., Devin, C., & Bengio, Y. (2015). *Embedding Word Similarity with Neural Machine Translation*. Retrieved 12 April 2024 from https://arxiv.org/pdf/1508.01582.pdf

- Kalchbrenner, N., & Blunsom, P. (2013). *Recurrent Continuous Translation Models* 2013 Retrieved 15 April 2024 from https://aclanthology.org/D13-1176.pdf

- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. *Stanza: A Python Natural Language Processing Toolkit for Many Human Languages*. In Association for Computational Linguistics (ACL) System Demonstrations. 2020.

- Qi, Y., Sachan, D. S., Felix, M., Padmanabhan, S. J., & Neubig, G. (2018). *When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?*. Retrieved from https://arxiv.org/pdf/1804.06323.pdf

- Sennrich, R., Haddow, B., & Birch, A. (2015). Neural Machine Translation of Rare Words with Subword Units. Retrieved 13 April 2024 from https://aclanthology.org/P16-1162.pdf

- Shaham, U., & Levy, O. (2021). *Neural Machine Translation without Embeddings*. Retrieved Retrieved 13 April 2024 from https://arxiv.org/pdf/2008.09396.pdf

- Song, J. (2021, Dec 10) *transformer-translator-pytorch.* GitHub. Retrieved 17 April, 2024 from https://github.com/devjwsong/transformer-translator-pytorch

- Sutskever, I., Vinyals, O., & V. Le, Q. (n.d.). *Sequence to Sequence Learning with Neural Networks*. Retrieved 13 April 2024 from https://arxiv.org/abs/1409.3215

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention is all you need*. arXiv.org. https://arxiv.org/abs/1706.03762

- Wolleb, B., Silvestri, R., Vernikos, G., Dolamic, L., & Popescu-Belis, A. (2023). Assessing the Importance of Frequency versus Compositionality for Subword-based Tokenization in NMT. Retrieved 16 April 2024 from https://arxiv.org/pdf/1508.07909v5.pdf

- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., & Norouzi, M. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Retrieved 15 April 2024 from https://arxiv.org/pdf/1609.08144v2.pdf

Appendix

Code Repository:

https://github.com/SeeuSim/cs4248-neural-translation

RQ-1-0 CHRF, CHRF++, ROUGE-L (F1) Evaluations for 10th example of test set

Tokenizer	Reference	Candidates	CHRF	CHRF++	ROUGE-L (F1)
Config 1	他们用了大量的时间 把意大利面条组装得越来越高	他们的他们的了，，在一个的的的的的的的的的的的	3.50	3.00	0.32
Config 2		它们它们他们的 ⁇ ， ⁇ ⁇ ⁇ 。	5.24	3.93	0.10
Config 3		蜒 的	25.00	12.5	0.13

RQ-1-1 Evaluations for proposed metrics with baseline RNN

Evaluation Metrics	Tokenizer Configurations		
	1	**2**	**3**
BLEU-4	4.08	2.41	0.10
CHRF	5.03	3.63	12.39
CHRF++	4.30	2.89	7.60
TER	96.55	93.93	97.42
ROUGE-1-R	0.19	0.08	0.05
ROUGE-1-P	0.54	0.22	0.33
ROUGE-1-F	0.26	0.11	0.08
ROUGE-2-R	0.02	0.00	0.00
ROUGE-2-P	0.05	0.00	0.00
ROUGE-2-F	0.03	0.00	0.00
ROUGE-L-R	0.21	0.77	0.05
ROUGE-L-P	0.18	0.10	0.33
ROUGE-L-F	0.19	0.08	0.09
BERT-R	0.54	0.51	0.44
BERT-P	0.56	0.55	0.54
BERT-F	0.55	0.53	0.49

RQ-1-2: BLEU, CHRF, CHRF++, TER SacreBLEU signatures

BLEU	nrefs:1|case:mixed|eff:yes|tok:zh|smooth:exp|version:2.4.1
CHRF	nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.4.1
CHRF++	nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.4.1
TER	nrefs:1|case:lc|tok:tercom|norm:yes|punct:yes|asian:yes|version:2.4.1

RQ-1-3: Illustrative examples of tokenization by each tokenization configuration for short, medium, and long sentences from the training dataset

Sentence	**Tokenization by each Configuration**		
	1	**2**	**3**
Thank you.	['Thank', 'you', '.']	['Thank', 'you', '.']	['thank', 'you', '.']
谢谢	['谢谢']	['谢谢']	['谢', '谢']
We shouldn't be doing it from outside.	['We', 'should', "n't", 'be', 'doing', 'it', 'from', 'outside', '.']	['We', 'shouldn', "'", 't', 'be', 'doing', 'it', 'from', 'outside', '.']	['we', 'shouldn', "'", 't', 'be', 'doing', 'it', 'from', 'outside', '.']
我们不能只做外在的努力	['我们', '不能', '只', '做', '外在', '的', '努力']	['我们不能', '只', '做', '外', '在的', '努力']	['我', '们', '不', '能', '只', '做', '外', '在', '的', '努', '力']
This piece is inspired by all of the hard work that men and women are doing on the inside to create better lives and futures for themselves after they serve their time.	['This', 'piece', 'is', 'inspired', 'by', 'all', 'of', 'the', 'hard', 'work', 'that', 'men', 'and', 'women', 'are', 'doing', 'on', 'the', 'inside', 'to', 'create', 'better', 'lives', 'and', 'futures', 'for', 'themselves', 'after', 'they', 'serve', 'their', 'time', '.']	['This', 'piece', 'is', 'inspired', 'by', 'all', 'of', 'the', 'hard', 'work', 'that', 'men', 'and', 'women', 'are', 'doing', 'on', 'the', 'inside', 'to', 'create', 'better', 'lives', 'and', 'futures', 'for', 'themselves', 'after', 'they', 'serve', 'their', 'time', '.']	['this', 'piece', 'is', 'inspired', 'by', 'all', 'of', 'the', 'hard', 'work', 'that', 'men', 'and', 'women', 'are', 'doing', 'on', 'the', 'inside', 'to', 'create', 'better', 'lives', 'and', 'futures', 'for', 'themselves', 'after', 'they', 'serve', 'their', 'time', '.']
而这些才华正是男人们女人们在狱中，所有的辛勤工作所激发产生的。为了让他们在结束服役后，能够为他们自己创造更好的生活。	['而', '这些', '才', '华', '正是', '男', '人', '们', '女', '人们', '在', '狱中', '，', '，', '所有', '的', '辛勤', '工作', '所', '激发', '产生', '的', '。', '为了', '让', '他们', '在', '结束', '服役', '后', '，', '，', '能够', '为', '他们', '自己', '创造', '更好', '的', '生活', '。']	['而这些', '才华', '正是', '男人们', '女', '人们在', '狱', '中', '，', '所有的', '，', '？', '工作', '所', '激发', '产生的', '，', '。', '为了', '让他们', '在', '结束', '服', '，', '？', '后', '，', '能够', '，', '为', '他们自己', '创造', '更好的', '生活', '，', '。']	['而', '这', '些', '才', '华', '正', '是', '男', '人', '们', '女', '人', '们', '在', '，', '狱', '中', '，', '，', '所', '有', '的', '辛', '勤', '工', '作', '所', '激', '发', '产', '生', '的', '。', '，', '为', '了', '让', '他', '们', '在', '结', '束', '服', '役', '后', '，', '，', '能', '够', '为', '他', '们', '自', '己', '创', '造', '更', '好', '的', '生', '活', '，', '。']

RQ-1-4 Translation Performance Samples by Config

Chinese Reference	Translation by each Configuration		
	1 (Word Based)	2 (BPE)	3 (WordPiece)
我说:“好的姥姥。”	我说:“她的了。”	我 说 : “ : “ , , ,	蜒 的
BLEU-4	33.01	9.29	0.92
BERT-F	0.85	0.72	0.53
我认为过去几年 发生的什么事是卓越的 是, 过去的几十年里, 我们经历了物理世界到数字化世界的转变。	我认为我的一个我的一个的的的的的的的的的的的, , 我们的一个世界上, 我们的一个世界上的一个世界。	我 认 为 , 在 ⁇ , , , , , , , , , , , , , , ,	蜒 的 的 的
BLEU-4	6.88	2.96	0.00
BERT-F	0.61	0.53	0.48
所以我们可以建立一个社区, 在这个社区里, 我们有来自小型企业背景的辅助商 在咖啡馆中、酒吧里为你出谋划策, 我们能为你提供这位满怀抱负的先生 所受到的帮助, 我们会问你:“你需要什么?	所以我们我们在一个, 的的的, 你的的, , , 你的的, ,	所 以 我 们 , ⁇ ⁇ ⁇ ⁇ ⁇ ⁇ ⁇ ⁇ ⁇ ⁇ ⁇ , ⁇ ⁇ ⁇ , , ⁇ ⁇ , , ⁇ , , , ⁇ , , , , , ⁇ , , , , , ⁇ , ,	蜒 的
BLEU-4	3.55	2.59	0.29
BERT-F	0.55	0.54	0.45

RQ-2-1 Average BLEU Scores by Sentence Length

Sentence Length (EN)	BLEU		
	RNN	LSTM	TRANSFORMER
0-20	12.30	1.18	15.27
21-30	7.60	0.07	12.36
31-40	5.84	0.06	11.76
41-50	5.35	0.06	11.46
51-60	4.50	0.07	10.78
60+	3.31	0.07	9.71

RQ-2-2 Average BERT-F1 Scores by Sentence Length

Sentence Length (EN)	BERT-F		
	RNN	LSTM	TRANSFORMER
0-20	0.69	0.49	0.73
21-30	0.65	0.43	0.73
31-40	0.63	0.42	0.73
41-50	0.62	0.43	0.73
51-60	0.61	0.43	0.72
60+	0.58	0.43	0.71

RQ-2-3 Graphical Representation of RQ-2-1 and RQ-2-2

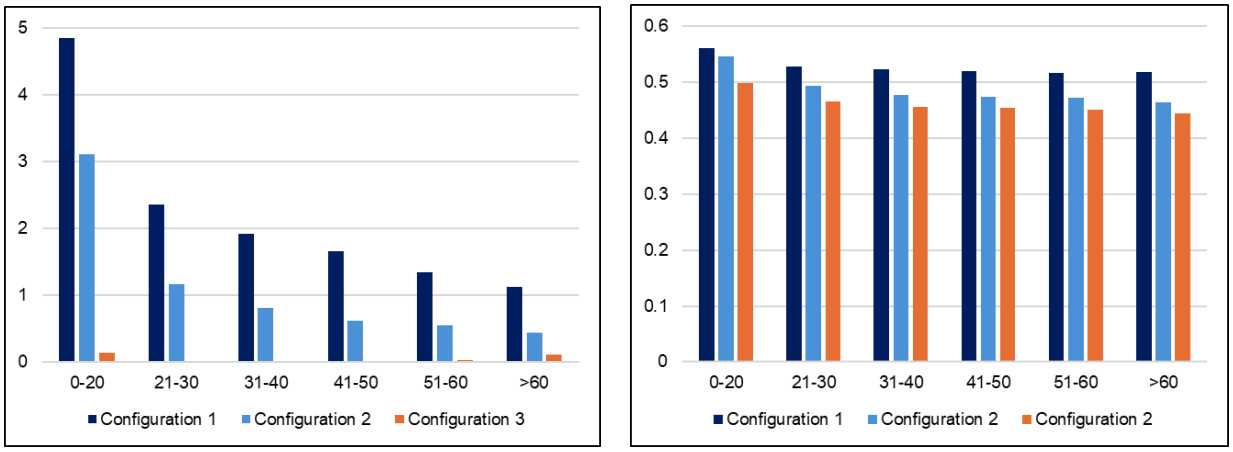


Figure. 2. (a) BLEU-4 for each configuration by sentence length; (b) BERTSCORE F1 for each configuration by sentence length

RQ-2-4 Model Architecture Evaluation on Short Sentences

Source (en)	Reference (zh)	Model	Model output	BLEU	BERT-F
I could be that person."	还好我不是那个人。	LSTM	我 们 怎 样 ⁇ ⁇ 。	5.09	0.63
		RNN	我 能 成 为 人 。”	17.11	0.72
		T-MODEL	我可以成为这个人。”	27.8	0.78
And I tried something interesting.	我做了一个有趣的尝试	LSTM	我 试 着 ⁇ 。	4.67	0.67
		RNN	我 尝 试 了 一 些 有 意 思 的 事 。	11.02	0.81
		T-MODEL	我尝试了一些有趣的事情	21.20	0.85
Probably the best part of it is what's coming down the pike in health.	这其中最好的部分或许是医疗	LSTM	这 是 一 只 ⁇ ,	2.54	0.59
		RNN	最 棒 的 部 分 是 , 健 康 领 域 。	12.69	0.75
		T-MODEL	也许最好的部分是健康水平下的	28.92	0.76

RQ-2-5 Model Architecture Evaluation on Medium Sentences

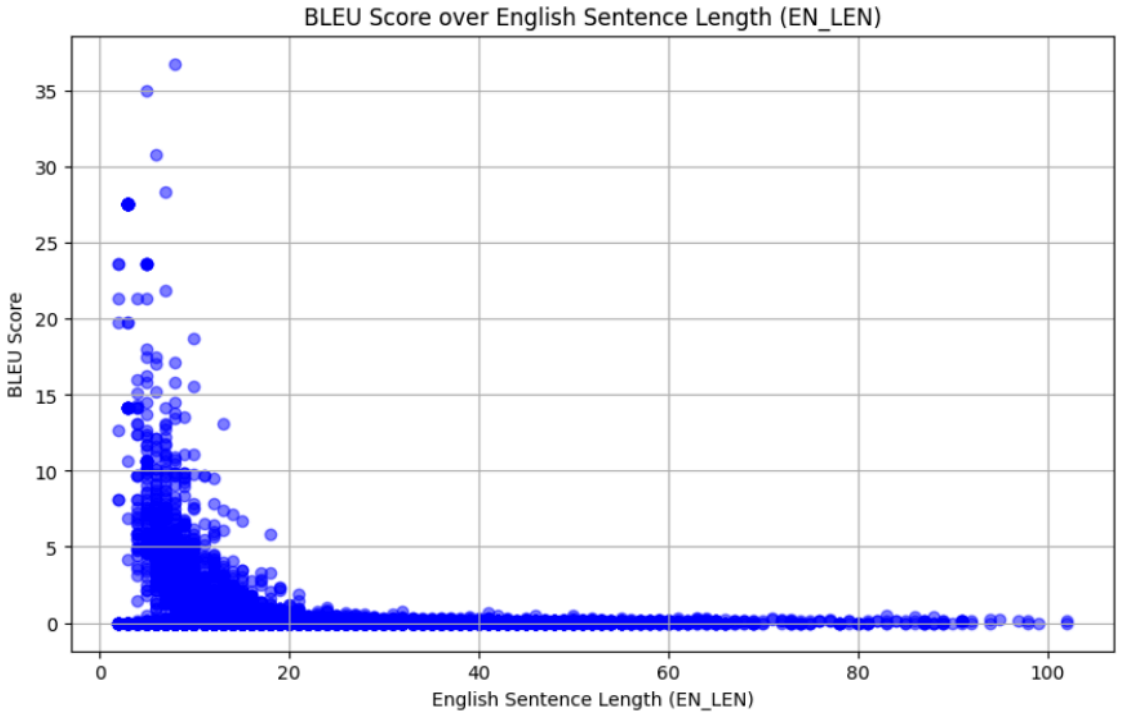
Source (en)	Reference (zh)	Model	Model output	BLEU	BERT-F
And he was talking about invisible demons and smoke, and how someone was poisoning him in his sleep.	他提到了 无形的恶魔和烟雾 有人怎样在他睡觉时给他下毒	RNN	他 谈 论 的 是 ⁇ 烟 和 烟 , , 还 有 他 被 毒 害 ⁇ 睡 。	4.22	0.67
		Transformer	他谈论的是看不见的恶魔和烟 ⁇ , 以及他为什么在睡眠中毒害。	14.87	0.76
This is because, in the mid-area here, people are at their most active, and over here they're at their most frail.	这是因为, 在中间这里 人们在他们最活跃的年龄 而在这里他们也是最体弱多病的时候	RNN	因 为 在 , , , , , , , , 活 跃 , , 最 最 脆 弱 。	2.75	0.62
		Transformer	这是因为在 ⁇ 拉这里, 人们都在非常积极, 这里他们处于最弱的状态	16.00	0.76

RQ-2-6: Model Architecture Evaluation on Longer Sentences

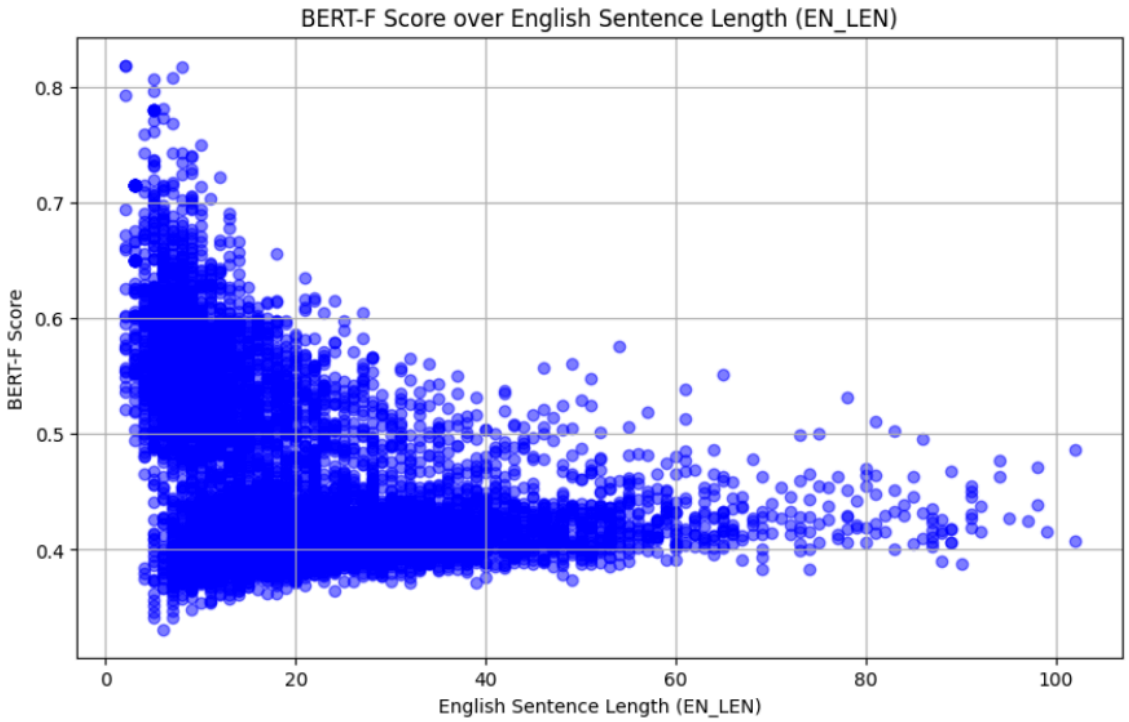
Source (en)	Reference (zh)	Model	Model output	BLEU	BERT-F
This building was already a part of this community, and then that first summer, when people started arriving and sharing the building on social media, the building ceased to be just an edifice and it became media, because these, these are not just pictures of a building, they're your pictures of a building.	这座建筑已经是社区的一个部分, 第一个夏天, 人们来到这里, 在社交媒体上分享照片, 这个只想成为建筑物的建筑 变成了媒体, 因为这, 这不仅仅是建筑的照片。它们是你拍摄的建筑相片。	RNN	这 个 建 筑 已 经 已 经 成 为 的 一 部 分 的 一 部 分 , 夏 天 夏 天 开 始 , , , 开 始 , , , , , , , , , , , , , , , , , , 照 片 , , 的 照 片 , , 照 片 的 照 片 ,	6.21	0.66
		Transformer	这个建筑已经是一个社区的一部分, 然后,第一年夏天, 人们开始参与这个社交媒体, 构建媒体, ⁇ 盖了 ⁇ , 它变成了媒体, 因为这些照片,并不只是你的建筑。	20.53	0.83

RQ-2-7 Individual scores against Sentence Length plots

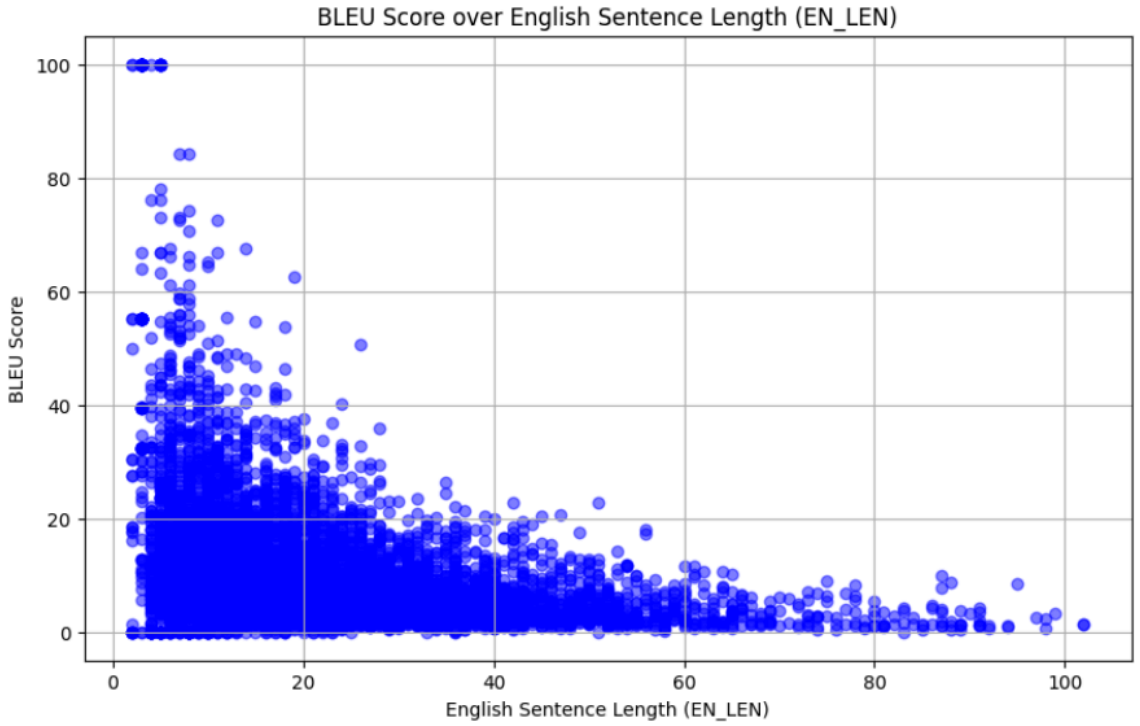
1. **Individual scores from LSTM translations against Sentence Length (BLEU)**



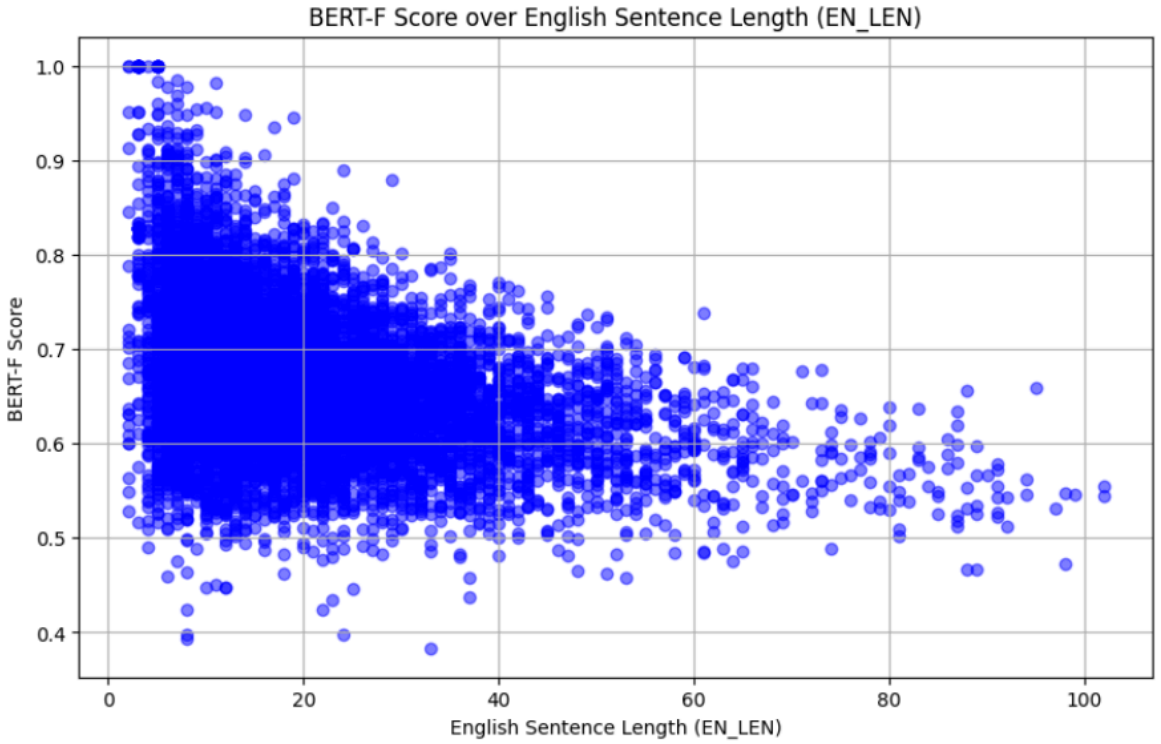
2. **Individual scores from LSTM translations against Sentence Length (BERT-F)**



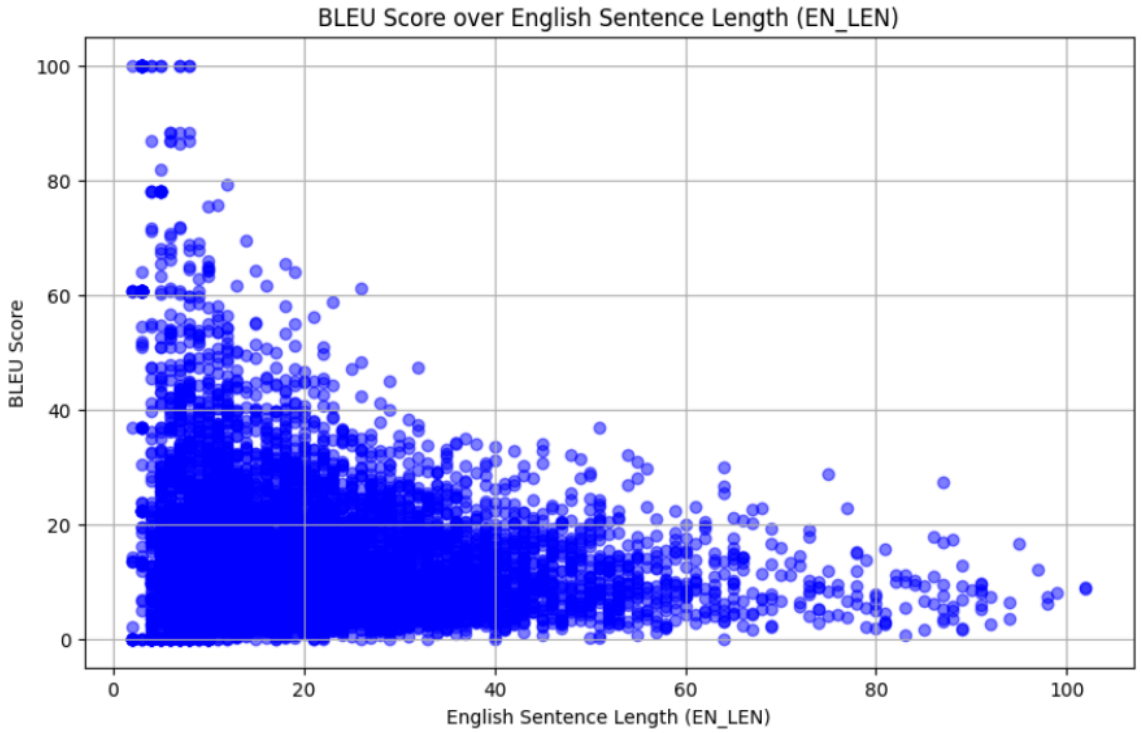
3. **Individual scores from RNN translations against Sentence Length (BLEU)**



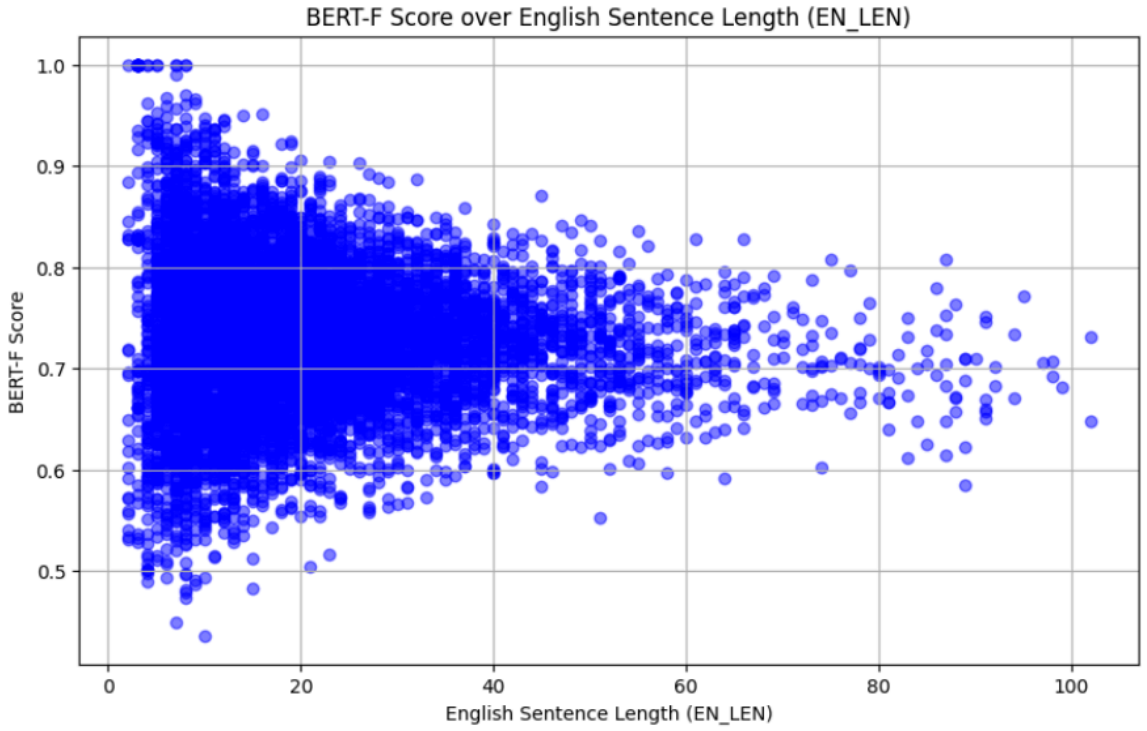
4. **Individual scores from RNN translations against Sentence Length (BERT-F)**



5. **Individual scores from T-MODEL translations against Sentence Length (BLEU)**

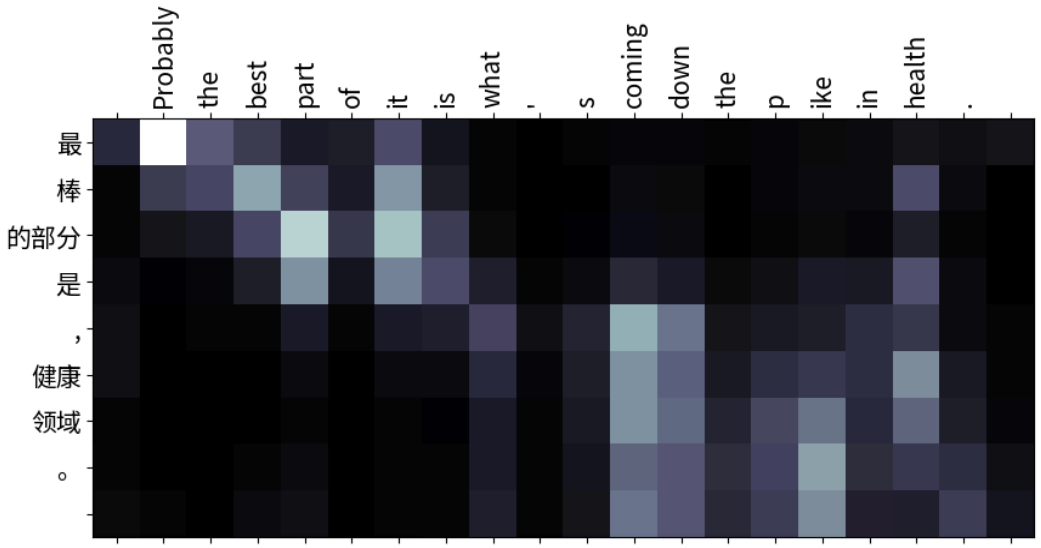


6. **Individual scores from T-MODEL translations against Sentence Length (BERT-F)**



RQ-2-8 Visualisation of attention in RNN variant for short and medium sentences

1. **Short sentence Attention visualisation on third example from Appendix 2.4**



2. **Medium sentence Attention visualisation on first example from Appendix 2.5**