
The University of Edinburgh

NLU+

*Natural Language Understanding, Generation and Machine Translation*

Coursework 2: Neural Machine Translation

*Authors*

s1805326 s1431314

# 1 Question 1: Comments on the code

## 1.1 Describe A

When self.bidirectional is True, there are two LSTMs(forward and backward) and thus two representations for each of the source words. Thus the hidden and cell states will now need to be a concatenation of forward and backward LSTM outputs. This is done using the combine function, where all forward representations(in the even indices of LSTM output) are separated from the backward representation(odd indices) and combined by concatenating the two. The hidden states in the LSTM store the LSTM output for that particular time step, whereas the cells output the hidden states, and the cell state stores the cell output for the past time steps, as long as forget gate is open. **Thus in *Long-Short term memory*, the hidden states store short term memory while the cell states store long term memory.**

## 1.2 Describe B

The attention scores calculated from the target input and encoder output and compared with the src_mask . This mask is used to tell the decoder to completely discard some words(like <pad> words, which have no contribution to target language) from consideration while calculating output word. This is done by using the src mask, which when multiplied with the attention scores sets the attention score for the unwanted words to -infinity, which in turn become 0 in the attention weight vector, which is a softmax of the attention scores. The attention weights and encoder output at the current timestep are then multiplied to get the attention context vector of that timestep.

## 1.3 Describe C

To calculate attention scores, we first project our encoder output from input dimension to output dimension by passing it through a fully connected linear layer. The attention scores are then calculated as a dot product of our target input and projected encoder output.This dot product is done by using batch matrix multiplication(i.e torch.bmm()), which performs matrix multiplication between the unsqueezed decoder representation of dimension(1,1,output_dims) and transposed projected encoder representation of dimension (1,output_dims,src_time_steps), which gives us our attention scores for the input sentence.

## 1.4 Describe D

The decoder is initialised as a single layered LSTM and caches previous hidden states implementing incremental decoding. This way, the model receives input at a single timestep corresponding to the immediately previous output token and must produce the next output incrementally. This way, the model caches any long-term state that is needed for the whole sequence. The cached states are loaded (if the cached state exists, otherwise they will be initialised as zeros). Cached_state is none when incremental_state is none, meaning that if we don't initialise incremental decoding, then no cached state will be initialised and the hidden and cell states will be initialised as zeros. Input_feed is the embedded word representation of the target word from the previous time step. This is used by the decoder to predict the consecutive word.

## 1.5 Describe E

Attention is integrated into the decoder through the attention() function called within the architecture of the decoder. The attention() function is used to force the decoder to focus on only certain parts of the input (achieved through masking). The attention weights are calculated through the AttentionLayer class which takes the the encoder output and applies the attentional weights to it. This is then fed to the decoder. The attention function is given the current target state as one of its inputs to be dot-producted with the hidden state of every source word. Dropout is applied to the embedding of the input words (input_feed) outputted from the attention layer. Generally it's used as a regularisation technique in order to prevent overfitting and boost the model ability to generalise. It works by cancelling neurons in the layer.

## 1.6 Describe F

A sampled mini-batch of source and corresponding target sentences, along with source sentence lengths is passed to the trained NMT model. The model-translated sentences for the input sample are then received

and error_loss between the translated sentences and target sentences is calculated and then normalised by the length of the sample. The calculated average loss for the mini-batch is back-propagated through the NMT model layers, and gradients are reset to 0 after the back-propagation to prevent any residual gradient value being used in the next iteration.

# 2 Question 2: Understanding the data

## 2.1

Plotting the distribution of the sentence lengths for both English and Japanese data shows that overall Japanese sentences are longer than the English ones. Moreover, for the same sentences, Japanese language uses more tokens than English language. [Figure 1] This is likely due to the fact that English language is more polysemic, and semantically productive, meaning that a single English word can express multiple meanings. This result in English sentences being overall shorter than Japanese, as they require less characters.

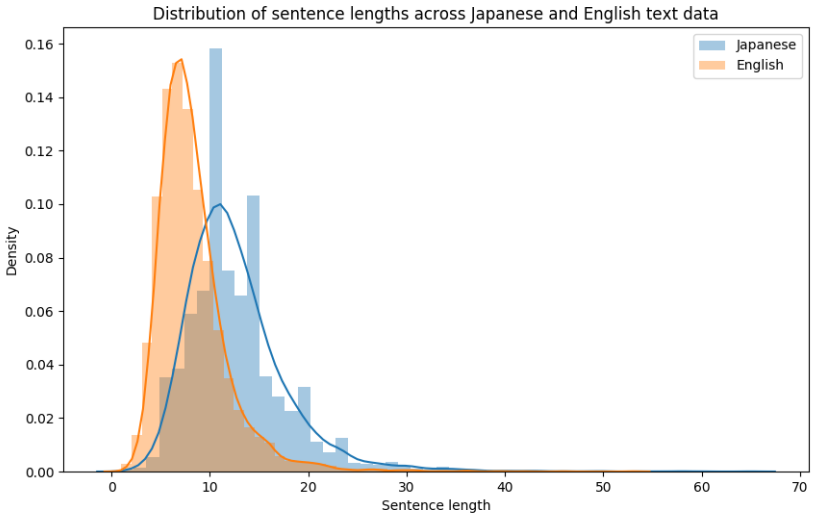


Figure 1: Distribution of Sentence Lengths

A great difference between the length of the sentences across the two languages may degrade the model's performance and more generally, neural machine translation models struggle to translate overly long sentences, outputting very short and incoherent translations [11]. Figure 2 shows a positive linear correlation between sentence length across the two languages ( Pearson Correlation Score = 0.768). This means that despite the overall differences in length, longer Japanese sentences correspond to longer English sentence. This is believed to mitigate to detrimental effect of length difference.

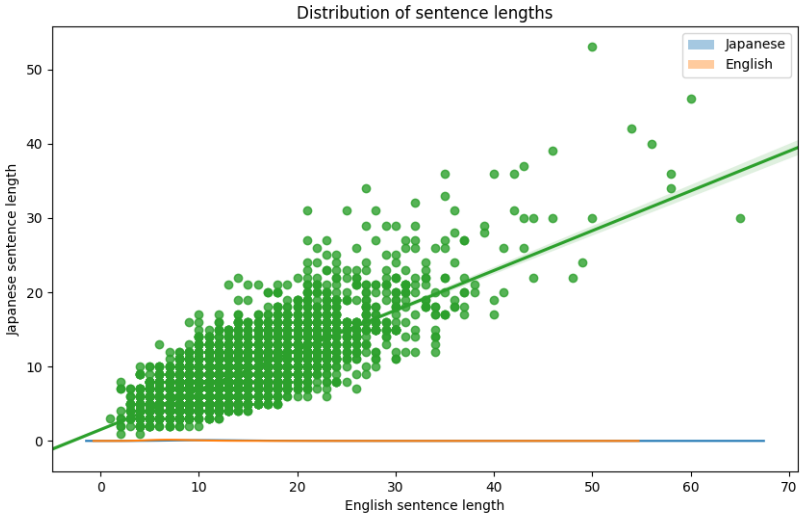


Figure 2: Correlation between sentences length

## 2.2

A word token is each lexical unit in a sentence. The number of tokens in each language corpus is:

**English : 93086**
**Japanese : 136900**

We can see that there are more tokens in the Japanese sentences than the English ones. This further corroborates our observation that the Japanese sentences generally use more tokens than their English counterparts and result in longer sentences.

## 2.3

Word types in a corpus refers to its vocabulary, i.e the number of unique word tokens in the corpus. The number of word types in each language corpus is:

**English : 7040**
**Japanese : 8059**

## 2.4

The Preprocess code for the NMT system has a threshold which specifies for what word frequencies does a word get replaced by <UNK>. The threshold is set for 2, meaning all words having frequency less than 2 are replaced by <UNK>. Thus the number of <UNK> words in a corpus are synonymous to number of words occurring only once in the corpus. The number of <UNK> words in each language corpus is:

**English : 3331**
**Japanese : 4114**

We can see that the number of words in each corpus with lowest frequency is quite large (almost 50%) of vocabulary. This is in accordance with Zipf's law [10] that shows the inverse relationship between word frequencies and word frequency counts.

## 2.5

1. *Sentences Length*:
   The difference in sentence length might lead to a degrade in the translation performance of the model. This is because the model will struggle to pair source and target words. This will practically result in difficulties with source and target words alignment which however will likely be mitigated by the linear correlation of the sentences length across the two languages.

2. *Type/Token ratio*:

   **English: 0.0756**

   **Japanese: 0.0589**
   The type/token ratio (TTR) is a measure used to assess lexical variation [2, 3] and morphological complexity within and across languages, where higher TTR values suggest higher lexical diversity [5]. This means that English as compared to Japanese shows a higher degree of lexical variation, and a single meaning can be expressed by different words (that is, there are a more interchangeable synonyms in English than in Japanese). This, together with the fact that English is also a polysemic language, means that the model is more likely to select a wrongful translation because it matches the target word context but does not actually represent the target word semantics.

3. *Volume of <UNK>*
   A rather high number of <UNK> words in vocabulary will likely worsen the performance. There are more <UNK> words in Japanese rather than English. This will likely decrease the translation performance since there will be more Japanese words that cannot be translated. This is a recurrent problem in neural machine translation and is sometimes referred to as Out Of Vocabulary Problem (OOV) [7].

# 3 Question 3: Improved Decoding

## 3.1 Greedy Decoder

A greedy decoder selects at each time step the word with highest probability in order to find the best possible translation. This doesn't always yield the most accurate translation because the algorithm computes a decision based on the information it has at the current time step, disregarding the greater word context. In other words, while the greedy algorithm is designed to find a locally optimal solution, it will often fails to find the global more accurate alternative.

In a translation task, the role of the greedy decoder is to find the most likely translated sentence (output) given the input. In other words, it maximises the conditional probability:

$$argmax P(output|input)$$

The greedy algorithm will select the word with highest probability at each time step, rather than the full optimal sentence.

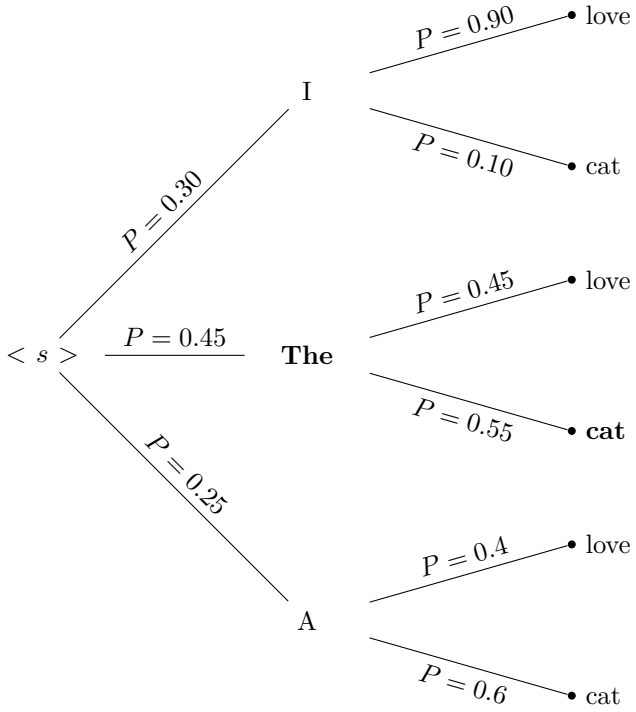


Figure 3: Probability tree of the vocabulary [I,The,A,love,cat]. The greedy choice is in **bold**.

For example, as shown in Figure 3, greedy search would choose the sentence ***the cat*** (p = 0.248) over the more probable ***I love*** (p=0.27) because it picks from the highest probability at every time step without knowing the future distributions.

## 3.2 Beam Search

The main difference between greedy and beam search is that the beam-search algorithm chooses $k$ best alternatives, rather than the 1-best for each time step. The number of best-items considered $k$ represents the *width* of the beam. Thus, in order to modify the algorithm we would firstly initialise the parameter $k$ to define the width of the search. This way, the algorithm will consider a $k$ number of best words at each time step. Then, for each of the $k$ possibilities, the algorithm will look for the best second words, and so on. In other words, the beam search algorithm becomes a greedy algorithm when the parameter $k$ is set to 1.

### 3.3 Length Normalisation

Decoder models tend to favour shorter sentences because for every extra word in a sequence, a new probability is added to the computation. Since every probability is smaller than 1, multiplying the words probabilities together will lead to an increasingly small number, thus reducing the probability for the whole sentence. This phenomenon is exacerbated by using a beam search algorithm, which with an increased beam size will tend to choose shorter sentences. This is referred to in the literature as **length bias** [8]. A way to tackle this problem is to use **length normalisation** [16], that is to normalise the log of the product of the words probabilities by the total number of words in the output. This can be seen in the equation below.

$$logP(T|S) = \frac{1}{|T|} \log P(T|S)$$

where T and S are respectively target and source sentence, and $|T|$ is the length of the target sentence. By doing this, the algorithm will now consider the average of the probability of each word. This procedure however is not deemed to be flawless and as suggested by Shao et al [13] it can produce incoherent and antithetical translations. As also mentioned by Li et al [6], length normalisation can produce redundant sentences, with the same word being translated multiple times for different source words (eg. *the cat jumped the cat*). This is due to the *averaging* effect of normalising the probabilities, and has been circumvented through different implementations such as deploying a coverage model to account for the times a source word is translated [6], or dividing by $length^{\alpha}$ where $\alpha$ is a hyper-parameter optimised during training [15].

## 4 Question 4

### 4.1

The command used to train the deeper architecture was typed in the terminal and was:

```
python train.py --encoder-num-layers=2 --decoder-num-layers=3 --save-dir ./question4
```

where `<question4>` is the name of the directory created to save the new checkpoints.

### 4.2

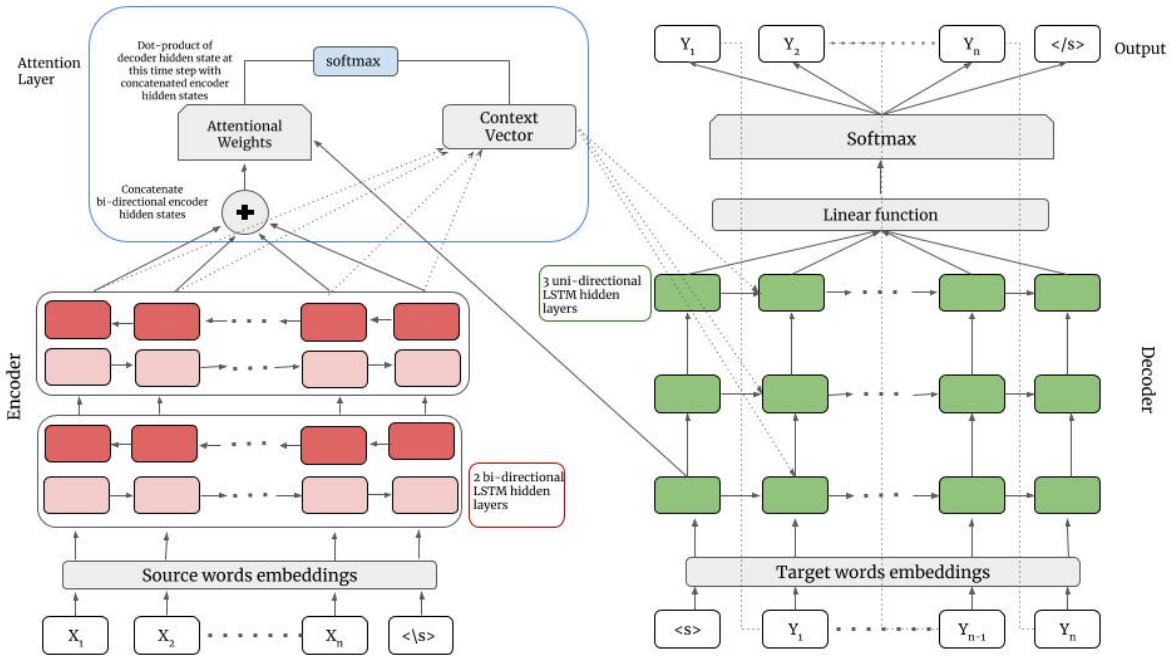


Figure 4: Encoder-decoder architecture using global attention

### 4.3

The new architecture achieved the following results:
*BLEU score*: 5.35
*Training loss*: 1.976
*Dev-set perplexity*: 26.8
Overall, the change in architecture worsened the model's performance in comparison to the baseline. This is probably due to overfitting, meaning that the additional layers made the model too flexible and "adjusted" to the training data, thus resulting in a worse performance on the testing unseen data. Moreover, while the architecture of the model increased in complexity with additional layers, other hyper-parameters (such as hidden units number) were not optimised to this new structure and could have contributed to the overall worsening in performance.

## 5 Question 5: Implementing the Lexical Model

The implementation can be seen in the submitted code in the lstm.py file. The parts of code have been marked by "Q5" comment and the implementation code has been explained using comments.

## 6 Question 6: Effects on Model Quality

| | **Baseline** | **Lexical Model** |
|---|---|---|
| **Train Loss for best perplexity** | 2.334 | 3.0 |
| **Validation Perplexity** | 22.4 | 20.0 |
| **Translation BLEU score** | 7.98 | 10.78 |

Table 1: Model training and test outputs.

The model training and test outputs for both baseline and lexical models are summarised in Table 1. Compared to the baseline model, we notice that the best validation perplexity lexical model has a slightly higher training loss than the equivalent baseline model. This is indicative of chances of over-fitting in the baseline model, thus lower generalizability of the baseline to test data. Furthermore, we see considerable improvements on the validation perplexity and BLEU score for the lexical model. This was due to the implementation of the lexical model, which noticeably improved translation performance. The lexical model allows to provide the decoder at each time step the source-words embeddings. This results in a better alignment of source and target words and eventually results in better translations, as indicated by the BLEU score.

# 7 Question 7: Effects on Attention

The figures that follow(Fig 5,6,7 & 8) are heat-maps for two source-target pairs generated using both baseline and lexical models (the heat-maps for the other sentences can be found in the Appendix). These heat-maps show the alignment between the source and target sentence words, where stronger alignments are represented by darker colours in the grid. As is evident from the figures and supporting tables , for the same source sentence, the alignments of source-target words differ between the baseline and lexical models. This difference could be due to the lexical model using attention-scored source embeddings directly alongside the LSTM decoder output. While the LSTM decoder tries to align words that fit into the context of the target sentence, the direct influence of source embeddings includes the source context as well when aligning input and output words.

**Sentence 1**

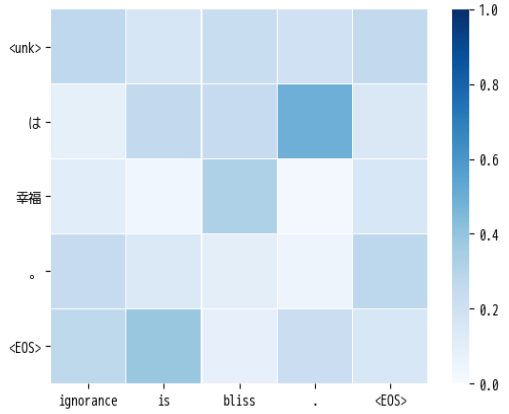


Figure 5: Source and target words alignment - Baseline Model

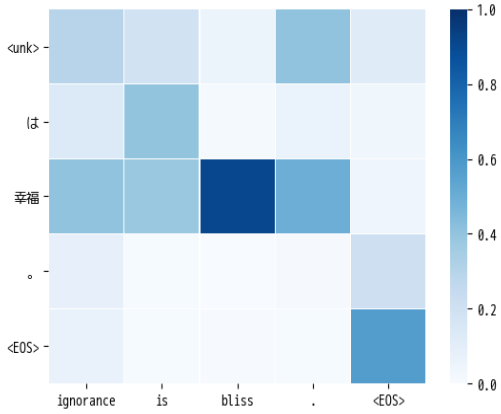


Figure 6: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | 無知は幸福。 | 無知は幸福。 |
| **Target Sentence** | ignorance is bliss. | ignorance is bliss. |
| **Translation** | I'm sorry . | happy is happiness . |

Table 2: Translations of the baseline and lexical model

We look at Figures 5 and 6, alongside Table 2 to see how the difference in the heat-maps lead to different translated outputs.For the source Japanese sentence " 無知は幸福。" , the translation generated by the baseline model is "I'm sorry", whereas the lexical model outputs "happy is happiness". Neither of these translations mean anything close to the target "Ignorance is bliss", but the lexical translation has "happiness" which we can assume means "bliss". This can be seen from the heat-map for lexical model where " 幸福" aligns strongly with the third translated word ("bliss" in target sentence) and thus the model outputs a word with similar context and better frequency ,i.e happiness. We also see for the first word to be translated in the output, the attention values for the source words in the baseline as evenly spread out, thus making it inconclusive for the baseline to select which word to output. Thus it probably outputs the most probable word that follows <s>, which is I'm. In case of the lexical model, the first decoded word receives high attention from source word " 幸福" , which translates to happy in English, and thus "happy" is generated as the first word in the lexical model. Thus it can be assumed that the attention weights are central to determining the output.

**Sentence 4**

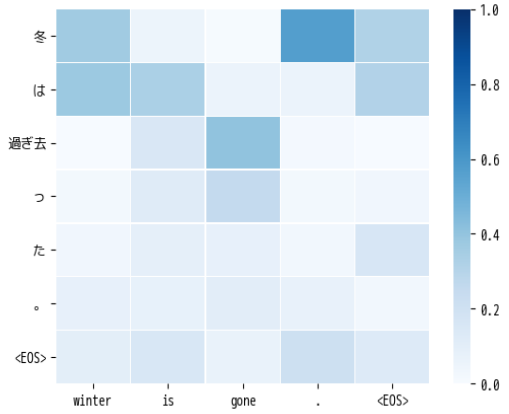


Figure 7: Source and target words alignment - Baseline Model

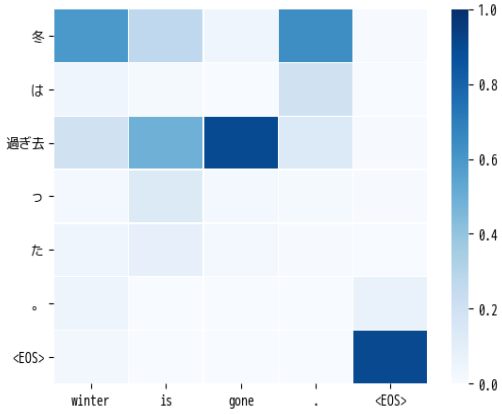


Figure 8: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | 冬は過ぎ去った。 | 冬は過ぎ去った。 |
| **Target Sentence** | winter is gone. | winter is gone. |
| **Translation** | we had a lot of finished in the winter | the winter was gone |

Table 3: Translations of the baseline and lexical model

However, the given example (Figures 7 & 8 , Table 3) shows that our assumption may have been untrue. Looking at the heat-maps, the attention follows an unusual pattern. It seems highly unreasonable for the first source word to be of high attention for generation of both first and fourth translated word. The Japanese word ” 冬” which means ”winter” is strongly aligned with both the first and last word of the sentence. While it seems plausible enough for both the source and translated sentences to start with ”winter” , it seems highly unlikely that the sentence source sentence with ”winter” only at the beginning leads to translated sentence ending in winter.

While the baseline model falls prey to this irregularity, the lexical model is still able to generate a more accurate translation. Even though the lexical model exhibits a fairly high alignment between ” 冬” and the last word of the sentence, it still manages to output ”.” . This is because apart from the attention, which deals with context, the lexical model also considers the source context at that time step. As a result, since the source sentence has a period ”.” after it sees the Japanese word for ”gone”, the lexical model tends to use this context to output ”.” once it sees ”gone”.

Thus we can infer from the two examples that while attention plays some role in determining the right output word using context, it may lead to pitfalls, since it does not directly consider the source sentence, only the contextual embeddings. A lexical model which provides additional direct consideration of source is required for a more accurate translation process.

# 8 Question 8: Effects on Translation Quality

## 8.1 Introduction and hypothesis

Neural Machine Translation (NMT) models have been knowingly plagued by a series of impasses such as the *coverage problem*, the *unknown problem* and the *inaccuracy problem* [14]. The first one refers to the inability of most NMT models to keep track of the source word translated, resulting in over- or under-translations. The UNK problem on the other hand refers to the mis-translations caused by rare

words [7, 9]. Finally, the inaccuracy problem refers to the proclivity of NMT models to produce words that fit the context of the target sentence but don't reflect accurate the semantics in the source words, resulting in incoherent translations [1]. Examples of each of these from our baseline LSTM translations are provided in Table 4.

| Problem | Baseline Model | Reference Sentence |
|---|---|---|
| Coverage Problem | she got the red red | she painted the wall red |
| Coverage Problem | where is this best station to the station ? | is this the right way to yokohama station? |
| Coverage Problem | i read a lot of fifty fifty fifty room. | i read fifty pages further. |
| UNK Problem | I met the **coffee** yesterday. | I met **Ken** yesterday. |
| UNK Problem | he has a good **news**. | he earns a good **salary** |
| UNK Problem | I have a picture of my **mind**. | I have a picture of an **airport**. |
| Inaccuracy Problem | everyone did his own experience. | everyone recognized his skill. |
| Inaccuracy Problem | i want to work a job. | at times i feel like quitting my job . |
| Inaccuracy Problem | he is said to be a man of great help. | he seems to be rich. |

Table 4: Examples of some common problems with NMT with excerpts from our baseline model translations

Previous research has mostly focused on the UNK problem, such as in relation to low resources languages [12], and have proposed a *lexical model* to mitigate the issue [9, 1]. The model would provide the decoder with the source embeddings at each decoding step, showing better translation performance of proper nouns and rare/unknown words. Such a model proves especially beneficial when translating between languages with very different lexical characteristics and structure, such as Japanese and English. Although they're very different languages, both English and Japanese lexicon is characterised by polysemy, meaning that single words and ideograms have multiple and different meanings [4]. Some examples of this are *milk* in English (the dairy product and the verb synonym of exploit) or 認めた in Japanese ( which means to recognise, to acknowledge but also more generally to do or assess). What distinguishes polysemes from homonyms is that the first ones encompass different meanings within the same context, while the latter represents casual semiotic similarity without bearing any semantic resemblance (such as the verb to bear and the animal bear). For this reason, polysemes might be quite problematic to be translated under our baseline LSTM model, which is likely to incur in the inaccuracy problem and be biased by polysemes. We believe that lexical information about the source word should mitigate this problem by helping the decoder choosing the correct translation and discard alternatives that would fit the context of the sentence but not be a meaningful or truthful translation. **Thus, we hypothesise that the lexical model helps with word-sense-disambiguation and is especially helpful with sentences containing polysemes**.

## 8.2 Evidence

**In order to test our hypothesis we qualitatively compared and analysed translations from the baselines and lexical model along with the source and target sentences.** While overall translations were still relatively poor, the lexical model overall outperformed the baseline (as shown by the BLEU score as well). Specifically, most of the improvements were relative to translations of rare words or word-sense-disambiguation of polysemes and ambiguous words. We narrowed our analysis around the instances in which the lexical model outperformed the baseline model and highlighted the structure of the sentence (in terms of subject, verb and predicate). This was done in order to isolate different part of the Japanese sentences and match them with the corresponding translation. **Our analysis showed the influence of the lexical model on translation coherence and accuracy especially in the case of polysemes**. An excerpt of the sentences we analysed is shown below [Figure 9].

| Baseline Model | Lexical Model | Japanese Source Sentence | English Target Sentence |
|---|---|---|---|
| everyone did his own experience . | everyone acknowledged his skill . | 誰 も が 彼 の 技術 を 認め た 。 | Everyone recognised his skill |
| we had a lot of finished in the winter . | the winter was gone . | 冬 は 過ぎ去 っ た 。 | Winter is gone |
| i want to work a job . | i want to quit my work . | 私 は ときどき 仕事 を やめ た い 気 が する 。 | At time I feel like quitting my job |
| i like to have a lot of people to get up . | i like spring better than autumn . | 秋 より 春 の ほう が 好き だ 。 | I prefer spring to fall |
| we are looking at all . | we are crying . | 私 たち は 泣 い て い る の 。 | We are crying |
| he has a good news . | he has a good salary . | 彼 は い い 給料 を もら っ て い る 。 | He earns a good salary. |

Figure 9: Sample translations from the baseline and lexical model with respective source and target sentences

## 8.3 Evaluation

**The evidence from the baseline and lexical model translations supported our hypothesis**. For example the first sentence in the table (”everyone recognised his skill”) was mistranslated by the baseline model (”everyone did his own experience”) but was correctly translated by the lexical model (everyone acknowledged his skill). The Japanese source sentence 誰もが彼の技術を認めた presented the word 認めた, a verb that means *to recognise*, *to acknowledge*, but also more generally *to assess* and *to do*. While this verb bears different meanings, the baseline model mistranslated it as *to do* and similarly mis-chose the word *experience* instead of the target word *skill*. Another example is the third sentence, ”at time I feel like quitting my job”. While the lexical translation (”I want to quit my work”)is slightly shorter and neglects some parts from the target one, it’s still a better result than the baseline (”I want to work a job”). Similarly, here the Japanese verb する bears multiple meanings such as *to do*, *to perform*, *to work as*, but also *to decide on* and *to choose*. In this example, it serves as an auxiliary to the other verb やめたい ( *to quit* ) but gets mistranslated by the baseline model as *to work*, leading to the wrongful and counter-intuitive ”I want to work a job” translation. **Along our initial hypothesis, we also observed an effect of the lexical model on the correct identification and translation of particles such as topic markers** ( は) and verb-tense markers (った). These type of particles are usually suffixed to the word they refer to, and while they do not bear semantic information themselves, they alter the meaning and function of the neighbouring words. は is a very common ideogram, mostly used as a topic marker or to add emphasis and it’s used to mark agency when suffixed to the subject of the sentence. For example, the Japanese source sentence 冬 は 過ぎ去った, in English ”winter is gone”, has the character 冬 as subject of the sentence (*winter*) and the marker は is used to signify the subject’s relationship to the verb. This leads to the lexical model to translate the sentence as ”the winter was gone”. On the other hand, the baseline model could not integrate this information and mistranslated the subject of the sentence producing ”we had a lot of finished in the winter”. Overall, the current evaluation suggests that lexical models have the potential to circumvent some of the main issues in neural machine translation and prove particularly helpful not only for low resource languages, but also for word-sense-disambiguation, dealing with polysemes and generally achieving better translation performance.

# References

[1] P. Arthur, G. Neubig, and S. Nakamura. Incorporating discrete translation lexicons into neural machine translation. *arXiv preprint arXiv:1606.02006*, 2016.

[2] M. A. Covington and J. D. McFall. Cutting the gordian knot: The moving-average type–token ratio (mattr). *Journal of quantitative linguistics*, 17(2):94–100, 2010.

[3] C. W. Hess, K. P. Ritchie, and R. G. Landry. The type-token ratio and vocabulary performance. *Psychological Reports*, 55(1):51–57, 1984.

[4] Y. Hino, S. J. Lupker, C. R. Sears, and T. Ogawa. The effects of polysemy for japanese katakana words. In *Cognitive Processing of the Chinese and the Japanese Languages*, pages 241–270. Springer, 1998.

[5] K. Kettunen. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics*, 21(3):223–245, 2014.

[6] Y. Li, T. Xiao, Y. Li, Q. Wang, C. Xu, and J. Zhu. A simple and effective approach to coverage-aware neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 292–297, 2018.

[7] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[8] G. Neubig. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619*, 2017.

[9] T. Q. Nguyen and D. Chiang. Improving lexical choice in neural machine translation. *arXiv preprint arXiv:1710.01329*, 2017.

[10] S. T. Piantadosi. Zipf' s word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

[11] J. Pouget-Abadie, D. Bahdanau, B. Van Merrienboer, K. Cho, and Y. Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. *arXiv preprint arXiv:1409.1257*, 2014.

[12] S. H. Ramesh and K. P. Sankaranarayanan. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. *arXiv preprint arXiv:1806.09652*, 2018.

[13] L. Shao, S. Gouws, D. Britz, A. Goldie, B. Strope, and R. Kurzweil. Generating high-quality and informative conversation responses with sequence-to-sequence models. *arXiv preprint arXiv:1701.03185*, 2017.

[14] X. Wang, Z. Lu, Z. Tu, H. Li, D. Xiong, and M. Zhang. Neural machine translation advised by statistical machine translation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[15] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[16] D. Zhang, J. Kim, J. Crego, and J. Senellart. Boosting neural machine translation. *arXiv preprint arXiv:1612.06138*, 2016.

**Sentence 2**

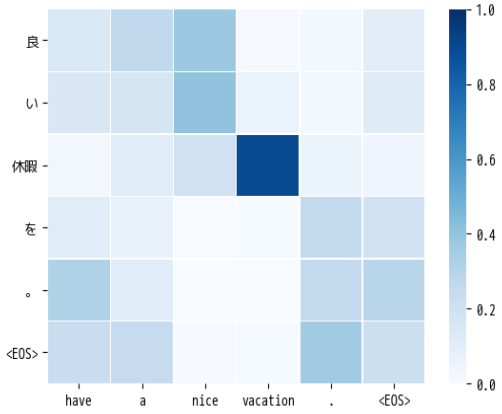


Figure 10: Source and target words alignment - Baseline Model

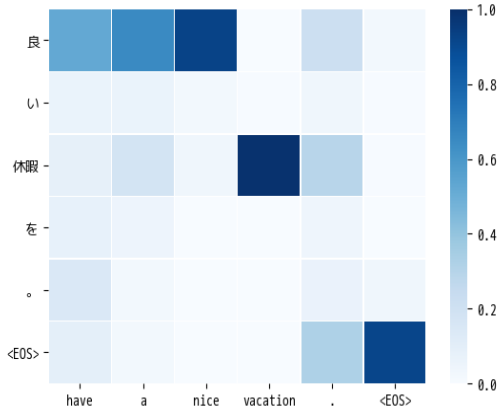


Figure 11: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | 良い休暇を。 | 良い休暇を。 |
| **Target Sentence** | have a nice vacation. | have a nice vacation. |
| **Translation** | I have a good time . | have a good vacation . |

Table 5: Translations of the baseline and lexical model

**Sentence 3**

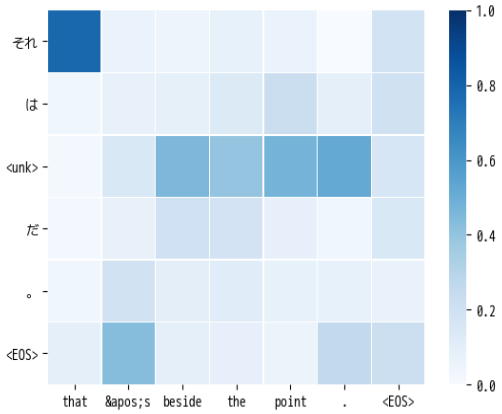


Figure 12: Source and target words alignment - Baseline Model

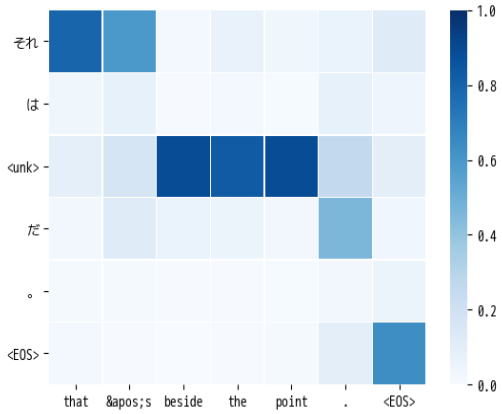


Figure 13: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | それは的外れだ。 | それは的外れだ。 |
| **Target Sentence** | that 's beside the point. | that 's beside the point. |
| **Translation** | it 's a good time. | it is a big condition. |

Table 6: Translations of the baseline and lexical model

**Sentence 4**

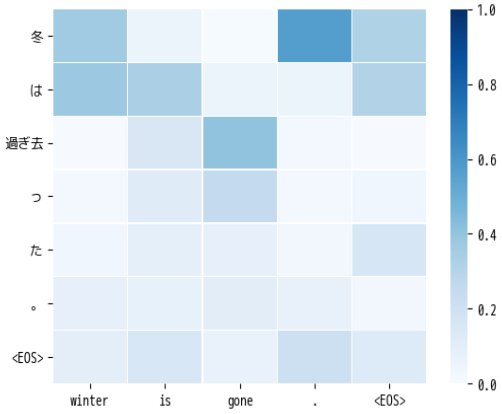


Figure 14: Source and target words alignment - Baseline Model

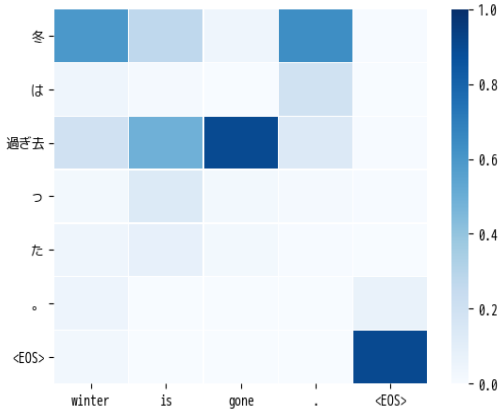


Figure 15: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | 冬は過ぎ去った。 | 冬は過ぎ去った。 |
| **Target Sentence** | winter is gone. | winter is gone. |
| **Translation** | we had a lot of finished in the winter | the winter was gone |

Table 7: Translations of the baseline and lexical model

**Sentence 5**

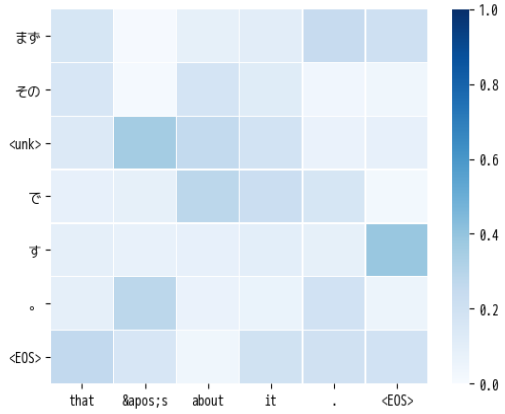


Figure 16: Source and target words alignment - Baseline Model

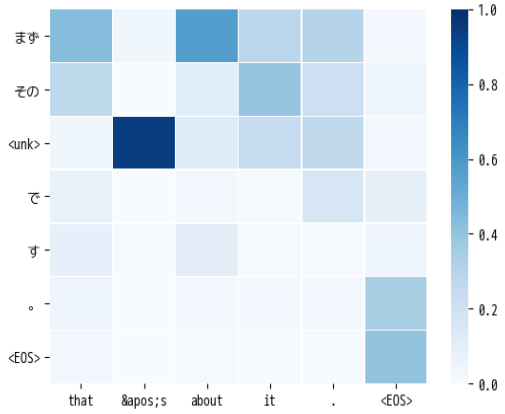


Figure 17: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | まずその🄵です。 | まずその🄵です。 |
| **Target Sentence** | that 's about it. | that 's about it. |
| **Translation** | The problem is the best of the way. | It is the same in the evening. |

Table 8: Translations of the baseline and lexical model

**Sentence 6**

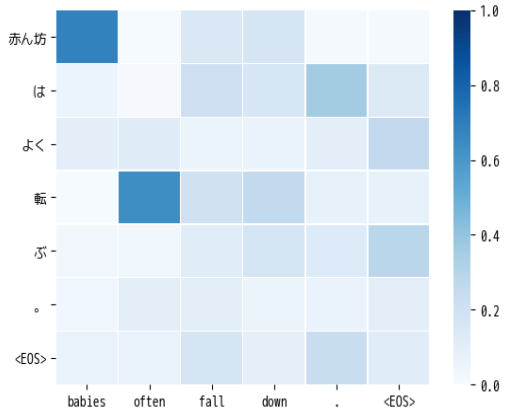


Figure 18: Source and target words alignment - Baseline Model

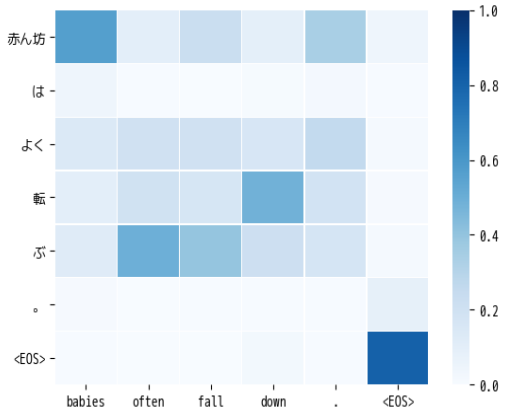


Figure 19: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | 赤ん坊はよく🄵ぶ。 | 赤ん坊はよく🄵ぶ。 |
| **Target Sentence** | babies often fall down. | babies often fall down. |
| **Translation** | The baby is familiar with a lot of | Baby is as well as well |

Table 9: Translations of the baseline and lexical model

**Sentence 7**

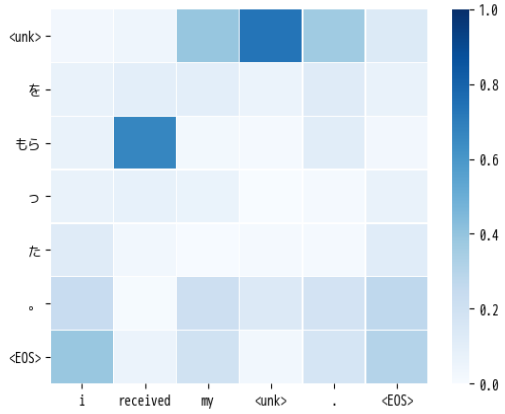


Figure 20: Source and target words alignment - Baseline Model

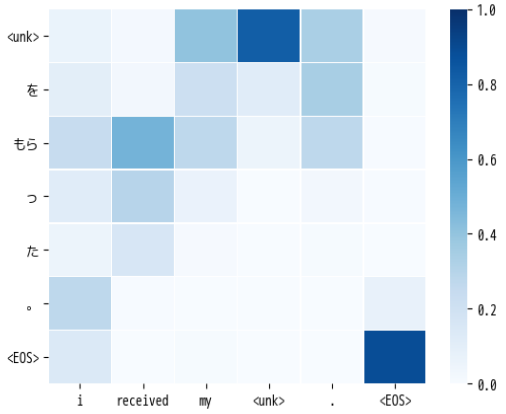


Figure 21: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | ボ🄵ナスをもらった。 | ボ🄵ナスをもらった。 |
| **Target Sentence** | i received my bonus. | i received my bonus. |
| **Translation** | I had a good time | I had a big pay |

Table 10: Translations of the baseline and lexical model

**Sentence 8**

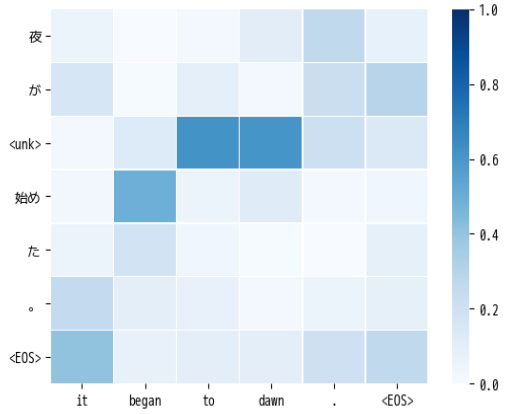


Figure 22: Source and target words alignment - Baseline Model

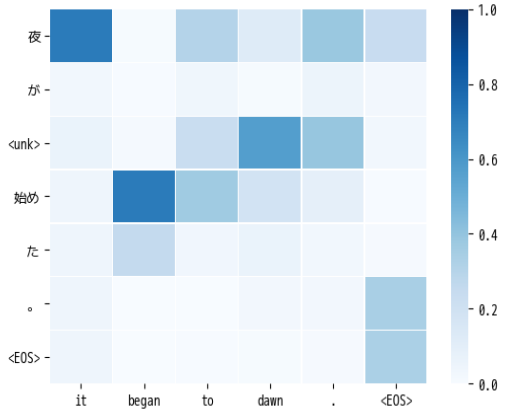


Figure 23: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | 夜が明け始めた。 | 夜が明け始めた。 |
| **Target Sentence** | it began to dawn. | it began to dawn. |
| **Translation** | The night began to the last night | The night began to the night |

Table 11: Translations of the baseline and lexical model

**Sentence 9**

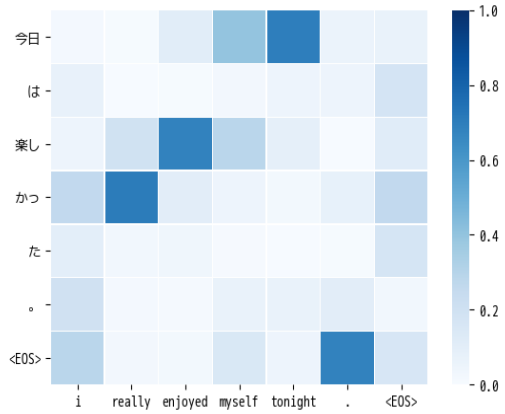


Figure 24: Source and target words alignment - Baseline Model

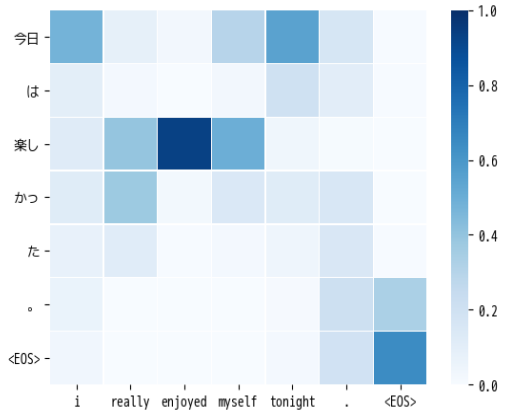


Figure 25: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | 今日は🄵しかった。 | 今日は🄵しかった。 |
| **Target Sentence** | i really enjoyed myself tonight. | i really enjoyed myself tonight. |
| **Translation** | i had a good time today. | it was good today. |

Table 12: Translations of the baseline and lexical model

**Sentence 10**

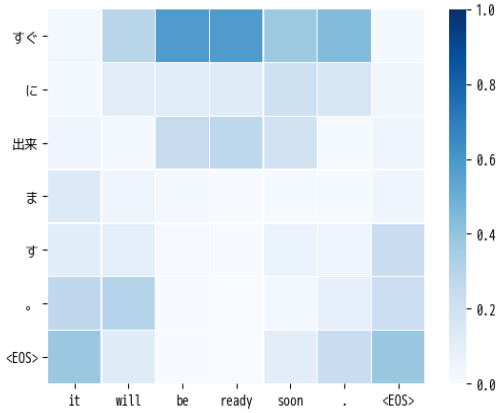


Figure 26: Source and target words alignment - Baseline Model

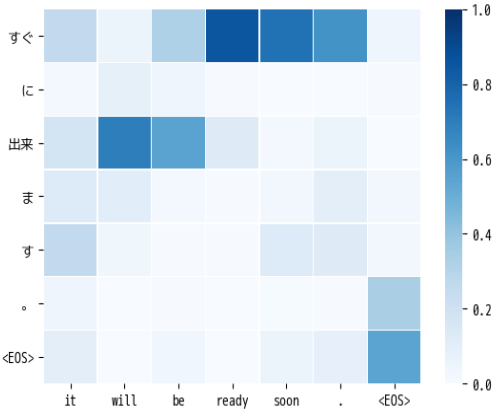


Figure 27: Source and target words alignment - Lexical Model

| | Baseline | Lexical Model |
|---|---|---|
| **Source Sentence** | すぐに出来ます。 | すぐに出来ます。 |
| **Target Sentence** | it will be ready soon. | it will be ready soon. |
| **Translation** | I can get soon soon. | I can get once once. |

Table 13: Translations of the baseline and lexical model