AKBL at the NTCIR-16 QA Lab-PoliInfo-3 Task

Ryoto Ohsugi
Toyohashi University of Technology
Japan
ohsugi.ryoto.dv@tut.jp

Teruya Kawai
Toyohashi University of Technology
Japan
kawai.teruya.ic@tut.jp

Yuki Gato
Toyohashi University of Technology
Japan
gato.yuki.am@tut.jp

Tomoyosi Akiba
Toyohashi University of Technology
Japan
akiba@cs.tut.ac.jp

Shigeru Masuyama
Tokyo University of Science
Japan
masuyama@rs.tus.ac.jp

ABSTRACT

AKBL team participated in the QA alignment, the Question Answering, and the Fact Verification subtasks. For the QA alignment subtask, our method firstly divides given question and answer texts into semantically consistent segments, then apply the Hungarian algorithm with the BM25 similarity metric to align those segments. For the Question Answering subtask, our system firstly selects a short segment relevant to a given question summary from the answer text, then converts it into the answer summary by using the abstractive summarizer based on the pre-trained BART. For the Fact Verification subtask, our best system firstly retrieves a passage relevant to a given claim from the assembly minutes, then checks if the passage entails the claim or not by using a BERT-based textual entailment classifier.

KEYWORDS

QA Alignment, Question Answering, Fact Verification, text segmentation, matching algorithm, passage retrieval, textual entailment, Okapi BM25, BERT, BART

TEAM NAME

AKBL

SUBTASKS

QA Alignment (Japanese)
Question Answering (Japanese)
Fact Verification (Japanese)

1 INTRODUCTION

NTCIR-16 QA Lab-PoliInfo-3 is a project aimed at presenting appropriate information for solving political issues. We participated in three of the sub tasks (QA Alignment, Question Answering and Fact Verification). QA Alignment aims to find the answer corresponding to a question in the form of a batch question and answer when given a question and answer in the form of a batch question and answer, for the Tokyo Metropolitan Assembly. Question Answering aims to find the answer corresponding to a question in the meeting minutes when given a summary of the question in the meeting minutes and to return the summarized result. Fact Verification, given a summary and the meeting minutes, aims to determine whether the content of the summary exists in the meeting minutes and, if so, to identify its scope. We proposed several methods for each of these tasks.

2 QA ALIGNMENT

2.1 Overview

The QA Alignment task aims to match the "member's question" with the corresponding "governor's answer"[1]. In the question-and-answer session, the questioner asks multiple questions at once, and the respondents answer the questions that can be answered at once, so the corresponding questions and answers are not directly matched. Techniques in this task act as pre-processing for other tasks such as summarization and topic detection.

2.2 Methods

To solve the problem in this task, it is necessary to predict the range of questions and answers and find the correspondence between the questions and answers. Therefore, our proposed method is divided into two steps. The outline of our proposed method is shown in the figure 1. The first step is to segment the minutes text. In this step, the question and answer range is predicted. The next step is to match the segments. In this step, find the corresponding question and answer.

2.2.1 Segmentation. First, split the text of the minutes. In the question-and-answer session, the member or governor speaks questions and answers at once. Therefore, we need to predict the boundaries between questions and answers, and we segment the utterances.

We use a rule-based approach to segmentation. This is because explicit delimiter phrases appear in question sentences and answer sentences. For example, on the question side, "I will ask the governor's opinion," and on the answer side, "I will answer about." Therefore, we created a rule (regular expression) that matches such a phrase. The rules used are shown in the table 1.

We use different rules for questions and answers. Question rules match phrases that ask for opinions or questions, such as "what is your opinion," "ask," or "how about." Answer rules match phrases such as "I will answer" and "For questions", as well as conjunctions such as "First," and "Next." The match position is also different: the question matches the last sentence of the segment, and the answer matches the first sentence of the segment. This is because the characteristic phrases frequently appear in such positions. Therefore, the boundary is added after the question is matched and the answer is before the matched sentence.

In post-processing, a segment with only one sentence is merged with the next segment. This is a heuristic process to reduce segmentation errors.

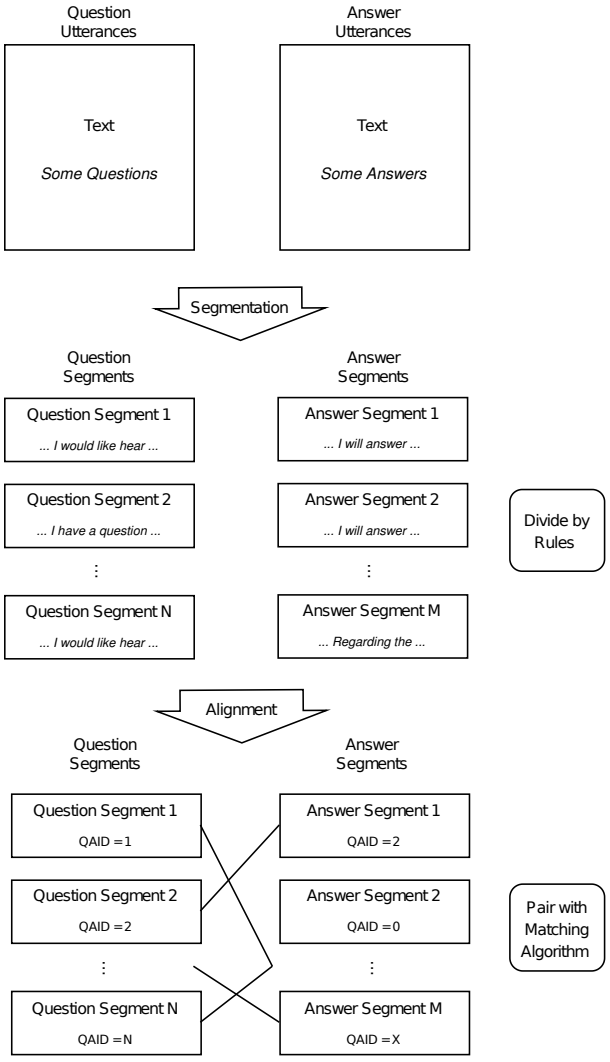

Figure 1: Outline of our proposed method in QA Alignment

Alignment. Then find the corresponding question and answer. Specifically, create a question/answer pair for the segment in the previous step. The processing procedure is as follows: (1) Vectorize the segmentation and (2) Match based on the similarity between the vectors.

Vectorization. A segment is a set of text and cannot be compared directly. So we transform the segment into a vector. We use Okapi BM25[2] for vectorization. BM25 is an extension of TF-IDF, which incorporates an average word count. Each vector is represented by a sequence of weights calculated by BM25 for each word.

Since it is necessary to divide into query units before applying, MeCab [3] is used for this process. We use only words with the following part of speech as queries: nouns, verbs, adjectives, adverbs, adnominal adjectives, and interjections.

Matching. By calculating the similarity between the vectorized segments, the corresponding questions and answers are found. For similarity, we use cosine similarity.

Table 1: Rules for segmentation (regular expressions)

Question	お?(伺い|尋ね) を?(いた)?し? (させていただき)?(ます|たい) |(見解|答弁|所見|課題|認識|考え|説明) を (お)?(求め|伺い|聞かせ|尋ね) |お?(答え|聞かせ)(て|を)?ください |ありがとうございました |いかがですか|どうですか |ではありませんか|るものです |(どのように|どう)(考えて|認識して|取り組む) |のですか|のでしょうか
Answer	お?答え (を)?(いた)?(|し|申し上げ) ます |初めに、|次 (いで|に|は)、|まず、 |他方で、|最後に、|続きまして、 |について (です|であります|でございます) |の (お話|お尋ね)(がございました|でございます) |(の|に関する)(ご)?質問で (ございま)?す |(質問|指摘|言及|お尋ね) が?ございました |(質問|指摘) を?いただきました

Due to the structure of the question-and-answer session, the question-and-answer pair is always one-to-one. Therefore, it is necessary to assign questions and answers without duplication while keeping high similarity. To solve such a problem (called an allocation problem), we used the Hungarian algorithm. By using this algorithm, the problem can be solved efficiently.

In most cases, a segmentation error will increase or decrease the number of segments. The Hungarian algorithm tries to make a one-to-one pair, but some segments are not assigned. However, in our proposed method, unassigned segments are output without being assigned. This is because forcibly creating pairs can lead to incorrect alignment and lower scores.

2.2.2 Additional processing. In this task, it is necessary to exclude sentences that do not correspond to the question and answer. For example, a sentence such as "I will answer n questions" in the answer has no corresponding question. Therefore, we add more rules to exclude such sentences. This rule is the first sentence of the answer segment and the sentence containing the word "お答え (answer)".

2.3 Experiments and Results

The minutes dataset is 2019 and 2020 of the Tokyo Metropolitan Assembly meeting. The reference dataset is the Tokyo Metropolitan Assembly Net Report (manual summary).

In this task, the QAID given to each sentence is used for evaluation. The same QAID in a question represents the scope of the question, as is the answer. The same QAID between a question and an answer indicates that the question and answer correspond. Count the number of matches between the prediction and the reference QAID and calculate the Precision, Recall, and F-measure. Please refer to the overview paper for the detailed evaluation method[1].

Table ref shows the evaluation results of our proposed method in Dry Run and Formal Run. Our proposed method is simple and uses only classical information retrieval methods and algorithms. Nevertheless, all F-scores achieved about 80% performance. There are two main reasons for this.

Table 2: Results of QA Alignment

version	Precision	Recall	F-measure
Dry Run	0.8268	0.8186	0.8202
Formal Run	0.8000	0.8311	0.8098

The first is the good performance of rule-based segmentation. In fact, by using only the rules, the correct segment boundary can be predicted with an accuracy of 90% or more. This is because the question answering is formalized. Frequent phrases in question answering greatly help segmentation.

Second, we could use existing methods for assigning questions and answers. The number of questions and answers is not large, and there is a one-to-one constraint. Therefore, a good score was achieved only by using the existing algorithm.

3 QUESTION ANSWERING

3.1 Overview

We used the following four steps in the Question Answering Task to output the answers.

Step 1 Step1 Extract all utterances of the respondent to a question
Step 2 Segmenting Utterances
Step 3 Find the segment of the answer that corresponds to the question
Step 4 Summarize the response segments

Based on the quantitative and qualitative evaluation of the output data, we discussed the problems of the system.

3.2 Method

The flow of the whole system is shown in the figure 2.

Step1 Extract all utterances of the respondent to a question

Since we are given who is answering the question, we extract all the utterances of the respondent.

Step2 Segmenting Utterances

Divide the utterance into topic-specific segments. An agenda utterance is one speaker talking about multiple topics. Using the expression when a speaker switches topics, we realized the division into segments. The splitting was done by modifying the regular expression of Kanasaki K et al. [4]. This is a collection of topic-changing expressions that are often used in conference proceedings. The regular expressions for splitting by topic are shown in Table 3.

Step3 Find the segment of the answer that corresponds to the question.

Select the appropriate segment to answer the question. Segments are divided into topics, so search for a segment that matches the topic of your question. The search uses the QuestionSummary and SubTopics as the query. We used Okapi BM25 [2], an algorithm used by search engines to rank documents according to their relevance to the query. was used. The weighting was made possible by scoring the segments by the two queries respectively. In other words, we take into account how much importance is given to either the question or the subtopic in the segment's search. The evaluation scores of the segment selection for each ratio of questions and subtopics are shown in Table 4. The evaluation data set used

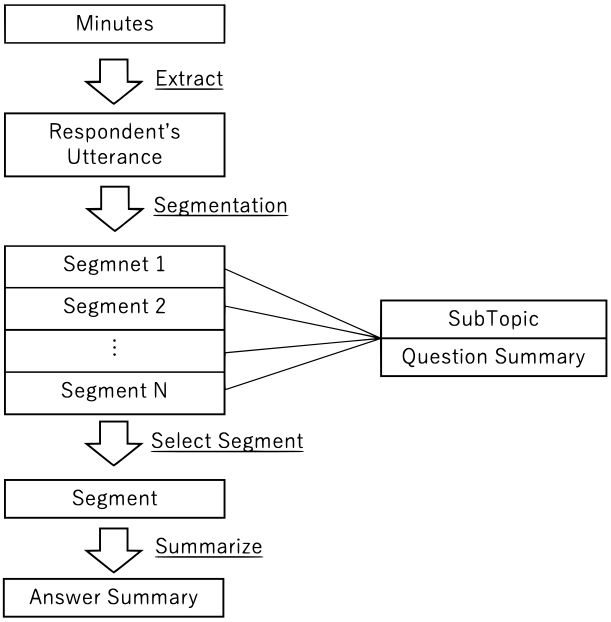

Figure 2: Outline of our proposed method in Question Answering

Table 3: Regular expressions to split by topic.

^まず (FIrst)|^最初に (At first)|^初めに (At first)|
^次いで (Next)|^最後に (Finally) |^終わりに (At the end) |
^[一二三四五六七八九十]+点目 (N point)|
^[^、]+についてで (about)
(す|あります|ございます) (が|けれど)|
^終わり (ま|で) す。(It's over) |
^以上で (that's all) |^ありがとうございま (Thank you) |
他の質問に (ついて |つきまして) は (For other questions)
^そこで (Therefore)|

Table 4: Evaluation of segment estimation for each BM25 score ratio

Ratio of Subtopic	Content rate	F-measure
0.20	0.7987	0.7954
0.30	0.8053	0.8069
0.40	0.8251	0.8287
0.50	0.8218	0.8245
0.60	0.8152	0.8182
0.70	0.8119	0.8172
0.80	0.802	0.8057
0.90	0.7624	0.7666
1.00	0.6865	0.6958

is the Segmented data from Polinfo2(Gold Segment). First, the segments were scored from the questions and subtopics respectively and normalized to a range of 0 to 1. The scores were then weighted with arbitrary ratios, and the sum of the question query score and subtopic score was used as the final score for the two queries. The results in the table show that a question to subtopic ratio of 0.6:0.4 is the most correct segment selection possible.

Step4 Summarize the response segments.
For the summarizer, we adopted an abstract summarization system based on the sequence to sequence model. We used Transformer [5] for the sequence-to-sequence model, which was implemented using OpenNMT [6]. The Transformer computes an embedded representation of a sentence from an input word sequence (encoding) and transforms it into another word sequence as an output (decoding). the Transformer differs from the Recurrent Neural Network (RNN) in that it relies only on self-attention it relies only on the mechanism. The decoder also employs a "cross-attention" mechanism that automatically pays attention to the relevant portion of the encoder's output. The decoder also employs a cross-attention mechanism, which automatically pays attention to the relevant portion of the encoder's output. The encoder layer consists of a stack of self-attention modules and feed-forward networks of skip connections and layer normalization, while the decoder layer consists of a stack of self-attention modules, cross-attention modules, and feed-forward networks of skip connections and layer normalization. The decoder layer consists of a stack of self-attention modules, cross-attention modules, skip connections, feed-forward networks of layer normalization. These layers are stacked several more times in both the encoder and the decoder.

3.3 Results

Table 5: Quantitative evaluation.

	ROUGE-1 F-Score	Baseline
DryRun	0.2416	0.0879
FormalRun	0.2306	0.0767

Table 6: Qualitative evaluation.

	Correspondece	Content	WellFormed	Total
○	101	43	137	49
△	17	52	8	47
×	32	55	5	54

The evaluation results of the output when the DryRun and FormalRun data are input are shown in Table 5. For Baseline, we adopted an algorithm that extracts the last 40 characters of an utterance. The results of manual evaluation of the output of NormalRun are shown in Table 6 shows the results of manual evaluation of the output of FormalRun. The evaluation was based on (1) Correspondence: "appropriateness as a response expression". (2) Content: Evaluation of the comprehensiveness of the important points in the answer in the meeting minutes. (3)Well-formed: Evaluation of "Naturalness of expression and grammar in Japanese". (4) Total: "Overall quality of the answer" was evaluated on a scale of ○, △, and ×.

3.4 Discussion

Table 5 shows that the baseline for the tail extraction type is significantly higher. However, the results of the manual evaluation in Table 6 show that the "Content" evaluation is low. We believe that this is due to the fact that the wrong segment is selected at the segment selection stage before inputting into the summarizer. Looking at "WellFormed", we can see that there are a few errors with incorrect grammar. In order to improve the performance of the system in the future, the segment selection needs to be better.

4 FACT VERIFICATION

4.1 Overview

The systems developed for the Fact Verification task are firstly retrieve a passage relevant to a given summary from the assembly minutes, then check if the passage entails the summary or not. For the latter process, we developed rule-based classifiers and ML-based classifiers. For the machine learning for the textual entailment, the classifiers based on the pre-trained BERT were employed.

4.2 Passage Retrieval

The assembly minutes have large amounts of text, so it is necessary to retrieve passages relevant to the given summary in order to classify the summary. For the passage retrieval, we investigated several types of passage and a IR metrics. The investigated types of passage are:

- pre-processed segment
- separate N sentences

For the former, we use the regular expressions shown in Table 7 to identify where a topic switches to another and divide the assembly minutes into segments. The positions after the sentences that match the start expression and the positons before the sentences that match the end expression are used to divide the assembly minutes. Other divisions are also made where the Speaker or Date changes. For the latter, instead of dividing the assembly minutes into segments, the passage is constructed from N sentences related to the given summary. They are ranked using the IR metrics described below with the given summary as the query.

The employed IR metrics is BM25+[7]. Using the summary as a query, scores are calculated and used to rank either segments or sentences. The formula for BM25+ is shown below.

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \left[\frac{f(q_i,D)(k_1+1)}{f(q_i,D) + k_1(1 - b + b\frac{|D|}{avgdl})} + \delta \right] \quad (1)$$

$$IDF(q_i) = log\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \quad (2)$$

4.3 Textual Entailment

After obtaining the relevant passage, the system checked if it entails the given summary or not. The task can be considered same as textual entailment. We employed two types of classifiers for the task.

- Rule based classifier

Table 7: Regular expression for segmentation

Start	^まず|^最初に|^初めに|^次に|^次いで|^続いて|^最後に |^終わりに|^では|^[一二三四五六七八九十]+点目 |^[^、]+についてで (す|あります|ございます)(が|けれど) |^終わり (ま|で) す。|^以上で|^ありがとうございま |他の質問に (ついて|つきまして) は |質問いたします。|^一方
End	伺い [^、]*ます。|お尋ね [^、]*します。|お答えください。 |(見解|所見|答弁) を求め [^、]*ます。 |(いかがで|どうで)(しょうか|すか)。|ありませんか。 |.+質問を (終わります|終了します|いたします)。 |(お答え|回答)(を?いたします|を?申し上げます)。

- Machine Learning based classifier

The rule based classifier uses the number of common nouns between the summary and the assembly minutes. Specifically, nouns extracted from the UtteranceSummary, RelatedSummary, and ContextSummary are compared with ones in the assembly minutes limited by Date and Speaker. If there are more than two nouns that appear only in the summary, it is judged false. Otherwise, it is judged true. In this case, the assembly minutes are divided into segments using the regular expression in Table 7, and the segment with the highest score in BM25+ is extracted.

For the machine learning based classifier, we employed the pre-trained language model, BERT[8], for the classifier. Our proposed method is shown in Figure 3. The Fact Verification training data is used to fine-tune the BERT. Passages are retrieved for both true and false summaries, and the dataset is constructed from the resulting passages and the given summaries. Then, BERT is fine-tuned for the textual entailment task by using this dataset. If the summary is determined to be true, its corresponding passage is identified in the same process as in the rule-base.

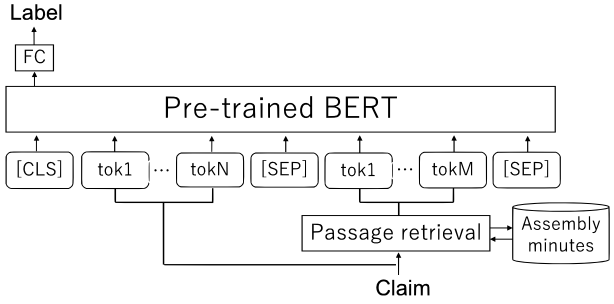

Figure 3: Outline of our proposed method for Fact Verification

4.4 Experiments

For separate N sentence, N = 7 is selected through our preliminary experiments shown in Figure 4. For BM25+, hyperparameters are set to $k_1 = 1.2, b = 0.75, \delta = 1.0$. For BERT, we used the BERT-base model and BERT-large model published by the Inui Laboratory at Tohoku University[1]. The number of training data for fine-tuning is 1024. The input for BERT is formulated as: [CLS] summary [SEP] passsage. The training epochs is set to 6 and the batch size is 16.

[1]https://huggingface.co/cl-tohoku

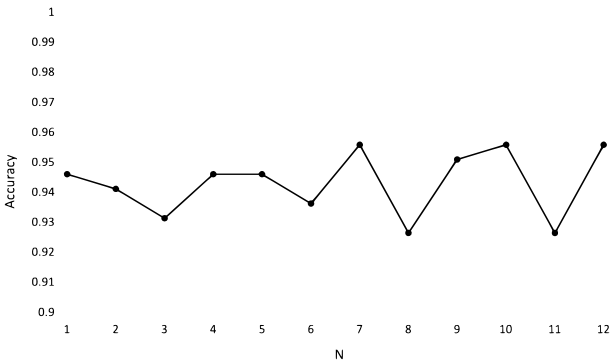

Figure 4: Preliminary experiments on optimal N

4.5 Results

The Formal Run results are shown in Table 8. The method using separate 7 sentences for passage type and BERT-large for TE method resulted in the highest F value.

Table 8: Results of Fact Verification Formal Run

Passage	IR metrics	TE method	recall	precision	F
segment	BM25+	Rule-based	0.8238	0.8139	0.8098
segment	BM25+	BERT-base	0.8610	0.8559	0.8506
segment	BM25+	BERT-large	0.8718	0.8668	0.8608
7 sentences	BM25+	BERT-large	0.9030	0.8951	0.8892

4.6 Discussion

For our examination, 20% of the 1024 training data were held out for test data, and BERT was newly fine-tuned with the remaining 80%. We examined samples that were incorrect in the rule-based classifier but correct in BERT-base. Table9 shows one of those samples. We found that the rule-based classifier did not work well on the summary that had an opposite polarity from the assembly minutes. Because BERT-base classified the example correctly, it can be said that the BERT-based method not only looks at the common nouns but also takes the meaning of the summary into account.

We also investigated the summaries that could not be correctly classified by the BERT-based method. It revealed that most of them had small number of words. Indeed, the average number of the words in those summaries was about 21, while that in all the summaries of the test data was about 32. We would like to improve the performance on those short summaries in our future work.

5 CONCLUSIONS

We took on the three tasks of NTCIR-16 QA Lab-Poli-Info-3 and proposed our own methodology. Future tasks for QA Alignment, Question Answering, and Fact Verification are to improve the splitting rules and process unassigned segments, improve segment selection and improve the correct response rate for short summaries, respectively.

Table 9: Typical example of an incorrect summary in a rule-based classifier

summary	今回の会議で知事が検討する議論は、今後行われる東京緊急対策二〇一の一つにあたり、東京都の**マンション耐震化促進については否定する**ものと考えられる。
assembly minutes	(省略) 都では、昨年六月に策定した東京緊急対策二〇一一の中で、マンション耐震化促進に向けた取り組みを緊急対策の一つとして取り上げ、 (省略)

REFERENCES

[1] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Madoka Ishioroshi, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Kazuma Kadowaki, Tatsunori Mori, Kenji Araki, Teruko Mitamura, and Satoshi Sekine. Overview of the ntcir-16 qa lab-poliinfo-3 task. *Proceedings of The 16th NTCIR Conference*, 6 2022.

[2] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pp. 109–126. Gaithersburg, MD: NIST, January 1995.

[3] T. KUDO. Mecab : Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.net/*, 2005.

[4] Katsumi Kanasaki, Jiawei Yong, Shintaro Kawamura, Shoichi Naitoh, and Kiyohiko Shinomiya. Cue-phrase-based text segmentation and optimal segment concatenation for the ntcir-14 qa lab-poliinfo task. pp. 85–96, 11 2019.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. Vol. 30, , 2017.

[6] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. *CoRR*, Vol. abs/1701.02810, , 2017.

[7] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. *In Proceedings of CIKM 2011*, pp. 7–16.

[8] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.