Noname manuscript No.
(will be inserted by the editor)

Information Extraction from Public Meeting Articles

Felix Giovanni Virgo* · Chenhui Chu · Takaya Ogawa · Koji Tanaka · Kazuki Ashihara · Yuta Nakashima · Noriko Takemura · Hajime Nagahara · Takao Fujikawa

Received: date / Accepted: date

Abstract Public meeting articles are the key to understanding the history of public opinion and public sphere in Australia. Information extraction from public meeting articles can obtain new insights into Australian history. In this paper, we create an information extraction dataset in the public meeting domain. We manually annotate the date and time, place, purpose, people who requested the meeting, people who convened the meeting, and people who were convened of $1,258$ public meeting articles. We further present an information extraction system, which formulates information extraction from public meeting articles as a machine reading comprehension task. Experiments indicate that our system can achieve an F1 score of 74.98% for information extraction from public meeting articles.

Keywords Information extraction · historical newspaper · corpus construction · public meeting

1 Introduction

Historical documents are important materials for understanding history, and many efforts have been made to convert them into texts for the purpose of analysis and preservation [2,13,1]. However, the documents are only digitized but not structured, making it difficult to extract the necessary information from a large amount of text.

Felix Giovanni Virgo* · Chenhui Chu
Graduate School of Informatics, Kyoto University
E-mail: felix@nlp.ist.i.kyoto-u.ac.jp

Koji Tanaka · Kazuki Ashihara
Graduate School of Information Technology, Osaka University

Yuta Nakashima · Noriko Takemura · Hajime Nagahara
Institute for Datability Science, Osaka University

Takao Fujikawa
Graduate School of Letters, Osaka University

*Corresponding author.

Extracting specific information from a text is called information extraction [20], which facilitates the analysis of unstructured text data by structuring it. Extracting and analyzing information from historical documents is expected to provide important historical insights.

The purpose of this study is to extract information about six key items: the date and time, place, purpose, people who requested the meeting, people who convened the meeting, and people who were convened, from *public meeting* articles in historical Australian newspapers. Public meetings were the main pillar of public opinion formation for Western Europe, spanning 120 years from the 19th to 20th century [5]. The knowledge obtained from public meeting articles is important for understanding Australian history, and it is expected that analysis of long-running public meeting articles will provide new insights in Australian history. However, public meetings articles are not structured.

In this study, we formulate information extraction from public meeting articles as a machine reading comprehension task. Namely, given a public meeting article, a machine reading model should answer the questions about the six items described in the previous paragraph by identifying corresponding spans from the article. However, there is no dataset available for the machine reading comprehension task from public meeting articles. To this end, we construct a dataset for extracting information in the public meeting domain based on the information contained in the public meeting articles annotated by an Australian history expert. In total, $1,258$ public meeting articles are annotated with the six items.

For the machine reading model, we use the ALBERT model [12], which achieves state-of-the-art performance in the machine reading comprehension benchmark of SQuAD [18].[1] We propose two fine-tuning methods using both the SQuAD dataset and our constructed public meeting dataset. Experiments indicate that our model achieves an F1 score of 74.98%. The main contributions of this paper are two-folds:

- We construct a dataset for information extraction from public meeting articles.[2]
- We formulate information extraction from public meeting articles in a novel way, i.e., as a machine reading comprehension task and present a system to extract this public meeting information.
- We propose a pre-training-based method with fine-tuning on our dataset to address the machine reading comprehension task for public meeting information extraction.

2 Related Work

2.1 Historical Corpus Construction

Several studies have been conducted on corpus construction of historical documents. Davies [2] constructed an American English historical corpus. They collected texts from books, newspapers, and magazines ranging from year 1810 to 2000. They further lemmatised and labeled part-of-speech (POS) tags on the corpus. A corpus from historical newspapers in French, Dutch, and German for the

[1] https://rajpurkar.github.io/SQuAD-explorer/

[2] The dataset is available at: https://github.com/felixgiov/public-meeting/

purpose of named entity recognition was built by Neudecker [13]. They annotated named entity tags using the INL Attestation Tool from the 17th to the 20th century *Europeana Newspaper*. Cassidy [1] constructed an Australian historical newspaper corpus. They converted newspapers from the 19th century to the 21st century into text data using OCR. The corpus is published on a website called Trove. Our corpus is also based on Trove. Different from Cassidy [1] and other previous studies, our corpus consists of extracted articles using OCR error correction with the specific topic of *public meeting*. More importantly, we annotated information extraction on the extracted public meeting articles.

2.2 Reading Comprehension Datasets and Models

There are many works on constructing machine reading comprehension datasets with various task formulations. In multiple-choice reading comprehension tasks, given a passage, a question, and multiple answer choices, the task is to select the correct answer from the given choices. MCTest [19] is one of multiple-choice reading comprehension datasets. It contains 550 passages from fictional stories created by crowdworkers, with 4 questions per passage, 4 answer choices per question, and 1 correct answer. Similar to MCTest, RACE [11] also consists of 4 questions per passage and 4 answer choices per question, but RACE is substantially larger than MCTest with almost $28k$ passages. This dataset is collected from the English exams for middle and high school Chinese students generated by human experts. In cloze-style reading comprehension tasks, the objective is to predict the missing word, usually a named entity, from a given passage. The Children's Book Test [7] consists of sentences from children's books with the goal to predict a blanked-out word of a sentence given the 20 previous sentences. The CNN/Daily Mail corpus [6] consists of news articles from CNN and Daily Mail with the goal to predict blanked-out entities from these articles. In extractive reading comprehension tasks, the goal is to extract the correct answer to a question from a given context paragraph. NewsQA [26] is an extractive reading comprehension dataset consisting of $100k$ question-answer pairs created by crowdworkers based on $10k$ CNN news articles. TriviaQA [8] consists of $95k$ question-answer pairs created by trivia enthusiasts with around 6 independently gathered evidence documents per question as context paragraphs. SQuAD 1.0 [18] and SQuAD 2.0 [17] are two famous extractive reading comprehension datasets. Questions and answer spans are annotated by crowdworkers on Wikipedia articles in a large scale. SQuAD 1.0 has around $100k$ question-answer pairs, and SQuAD 2.0 has around $150k$ question-answer pairs. Built upon the previous version of SQuAD 1.0, SQuAD 2.0 further introduced the problem of unanswerable questions and made annotations for it. Our dataset follows the same format as SQuAD 2.0. Although our dataset is small compared to SQuAD, we annotate on automatically extracted historical articles. We also use SQuAD to fine-tune our model.

The task of reading comprehension has been studied extensively using non-neural and neural-based models. For non-neural models, sliding window [19] measures the similarity of the bag-of-words representation of the question and candidate answer to the sliding window in the text. It uses inverse word count as the weight of each word. Logistic regression has also been used for the reading comprehension task [14,18]. For each candidate answer, several types of features are

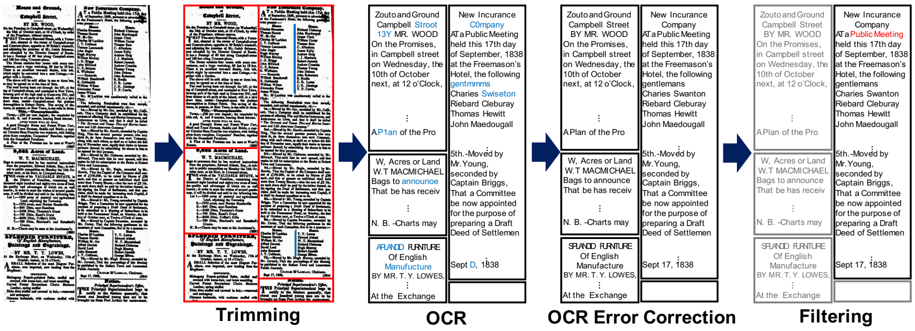


Fig. 1 Overview of the public meeting article extraction method (Small columns in articles are shown as blue lines in the "trimming" sub-figure, and OCR errors are shown in blue fonts in the "OCR" sub-figure).

extracted. These features include word frequencies, bi-gram frequencies, lengths, span word frequencies, span POS tags, lexical features, and dependency tree paths. Logistic regression is then used to predict whether a text span is the answer based on those features. The sliding window and logistic regression models are used in the SQuAD [18] paper.

For neural-based models, BiDAF [21] uses bi-directional attention flow networks, which considers attention in two directions: query-to-context and context-to-query. The two attention vectors are then combined with the original contextual embeddings. The combined results are used for span prediction. Pre-trained Transformer-based models, such as GPT [16] and BERT [3], can also be fine-tuned for reading comprehension tasks. While GPT only uses unidirectional language models to learn the representation of tokens, BERT is based on bidirectional representations. In this paper, we use the ALBERT model for our task.

3 Public Meeting Information Extraction Dataset

We first present how we extract public meeting articles and then describe the information extraction annotation process.

3.1 Article Extraction

Public meeting articles are extracted from the historical newspaper database Trove [1][3]. The overview of the public meeting article extraction method is shown in Figure 1. Note that the same method of [25] has been applied for article extraction, and we did not consider this section as a contribution of this paper. As the focus of this paper is about how to extract the information given the articles, the article extraction method of [25] can be considered as preparation for this work.

Trove covers major Australian daily newspapers and local newspapers. We target public meeting articles in Australian historical newspapers. As the OCR text provided by Trove lacks the rule lines information, it is difficult to extract

[3] https://trove.nla.gov.au

only public meeting articles accurately. Therefore, we address this problem by detecting rule lines from the images with the same method of [25]. We first identify the rule lines in newspaper images and then trim the rule lines to extract images for articles. Next, we apply OCR to the extracted article images to extract text from the articles and apply OCR error correction to OCRed text. Finally, we filter the articles with a query phrase to filter the articles and thus extract only the target articles that we are interested in.

3.1.1 Trimming

We use OpenCV[4] for identifying the rule lines in newspaper images and trimming. First, we binarize the newspaper images using the method proposed by Ohtsu [15]. The binarization method transfers grayscale images to binary images by calculating the threshold that maximizes the separation degree from the histogram of picture element numbers. Next, we apply the contour tracking processing algorithm of [23] to extract the contours from the binarized images. In order to identify the contours, this algorithm calculates the boundary of the binarized images and sequentially detects the pixels that are the contour counterclockwise. Areas with a height above a threshold and a width below a threshold are identified as a column, and areas with a width above a threshold and a height below a threshold are identified as an article in the newspaper image. The thresholds are tuned manually. After that, we can finally trim the article images accordingly.

3.1.2 OCR

OCR is generally performed following the procedures of character delimiter recognition, size normalization, feature extraction, and classification. An open-source OCR method, Tesseract [22], achieves 98.4% and 97.4% on newspaper articles in character and word level, respectively. However, after comparing the OCR accuracy of Google Drive[5] to Tesseract, we find that Google Drive works better by manually checking the results of some randomly sampled ORCed articles. Therefore, we use the OCR function of Google Drive for extracting text from the article images.

3.1.3 OCR Error Correction

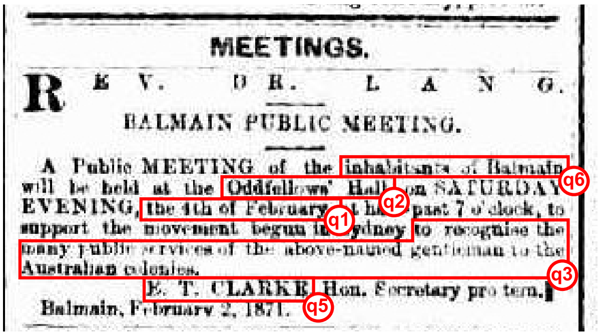
As the OCRed text has errors (as shown in blue fonts in the sub-figure “OCR” of Figure 1), we apply OCR error correction to the OCRed text. We use a statistical machine translation based model [10] for OCR error correction as it shows the best performance in our experiments [24].

3.1.4 Filtering

We filter the OCRed articles that are not our target with a query phrase “public meeting,” leaving the target articles to be extracted. In order to allow the error of character recognition by OCR, we define similarities in the character level. We

[4] https://opencv.org/

[5] https://www.google.com/drive/



MEETINGS.

REV. DR. LANG.

BALMAIN PUBLIC MEETING.

A Public MEETING of the inhabitants of Balmain will be held at the Oddfellows' Hall on SATURDAY EVENING, the 4th of February at half past 7 o'clock, to support the movement begun in Sydney to recognise the many public services of the above-named gentleman to the Australian colonies.

E. T. CLARKE, Hon. Secretary pro tem.

Balmain, February 2, 1871.

Fig. 2 Example of a public meeting article. Information corresponding to each question is shown in the red boxes (information corresponding to question number 1 is shown in box q1 and so on).

use the Python *difflib* module SequenceMatcher[6] for calculating similarities. In SequenceMatcher, the similarities between a character string pair is defined as:

$$Similarity = \frac{2M}{T}, \tag{1}$$

where M is the number of matched characters, and T is the sum of character numbers in the character string pair.

We get word n-grams from the articles according to the number of words in the query character string. We then calculate the similarity between the n-gram and query character string and take the articles with the highest similarity above a threshold as the target article. The threshold is tuned on a manually extracted validation set of public meeting articles, which shows the highest F-score for article extraction evaluation. For the details, please refer to [25].

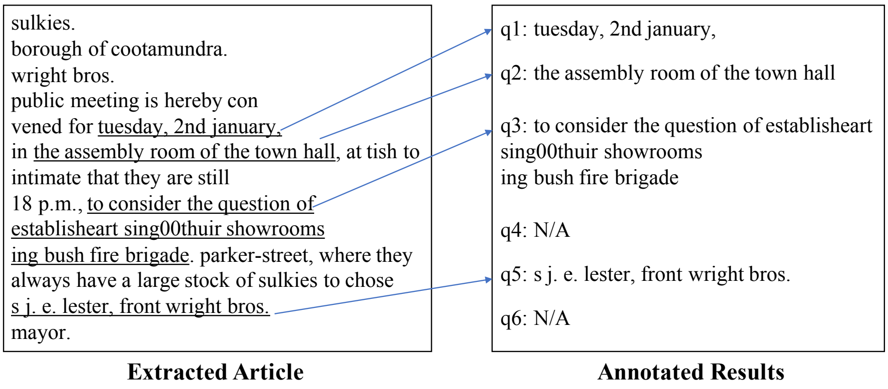
3.2 Annotation Process

After extracting public meeting articles, we annotate the six items, i.e., 1) the date and time, 2) place, 3) purpose, 4) people who requested the meeting, 5) people who convened the meeting, and 6) people who were convened in the articles, which are the key elements of public meetings. As we formulate it as a machine reading comprehension task, the six items are named as questions from q1 to q6, which correspond to:

q1 When was the public meeting being held?
q2 Where was the place of the public meeting being held?
q3 What was the purpose of the public meeting?
q4 Who requested the convening of the public meeting?
q5 Who was asked to convene the public meeting?
q6 Who were convened to attend the public meeting?

Answer spans to these six questions of a public meeting article example are shown in Figure 2.

[6] https://docs.python.jp/3/library/difflib.html



sulkies.
borough of cootamundra.
wright bros.
public meeting is hereby con
vened for tuesday, 2nd january,
in the assembly room of the town hall, at tish to
intimate that they are still
18 p.m., to consider the question of
establisheart sing00thuir showrooms
ing bush fire brigade. parker-street, where they
always have a large stock of sulkies to chose
s j. e. lester, front wright bros.
mayor.

Extracted Article

q1: tuesday, 2nd january,

q2: the assembly room of the town hall

q3: to consider the question of establisheart
sing00thuir showrooms
ing bush fire brigade

q4: N/A

q5: s j. e. lester, front wright bros.

q6: N/A

Annotated Results

Fig. 3 Example of annotation information from an extracted public meeting article.

Table 1 Public meeting information extraction dataset statistics.

Question type	Number of questions with answers	Average number of words in each answer
q1	819	5.58
q2	744	4.41
q3	531	14.79
q4	43	7.28
q5	105	4.72
q6	429	5.22

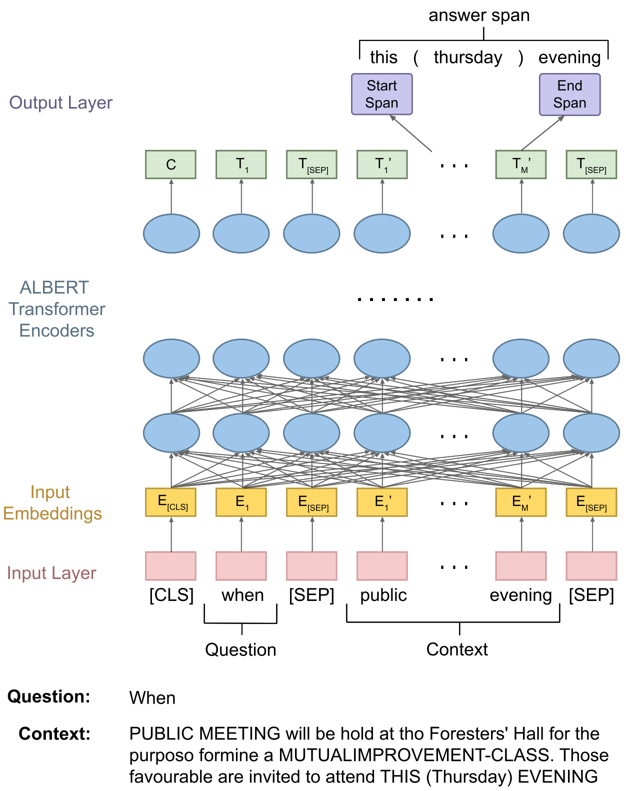
Answer spans of q1-q6 shown in Figure 2 were manually annotated by an expert in Australian history. The expert was asked to annotate these spans based on the automatically extracted articles. Figure 3 shows an example of annotation. The annotator was asked to annotate the answer spans[7] to each question with automatic OCR error correction to be consistent with our article and information extraction pipeline. For questions that answers are unavailable, a "N/A" tag was used. In total, 1,258 articles were annotated. Table 1 shows the statistics of our annotated dataset.

4 Information Extraction Method

In this study, we formulate information extraction from public meeting articles as a machine reading comprehension task. A machine reading comprehension task is a task in which a paragraph of text and a question are given, and the appropriate phrase span is identified from the paragraph as an answer to the question [18]. Therefore, it is possible to extract the six items of a public meeting by answering their corresponding questions listed in Section 3.2.

For the extraction method, we use ALBERT [12], which achieves the highest accuracy in the machine reading comprehension task for a single model and state-of-the-art accuracy for an ensemble model. Compared to BERT [3], ALBERT improves the performance by using sentence order prediction for pre-training tasks

[7] Note that in our annotation, answers always take a continuous span and do not intersect.



Question: When

Context: PUBLIC MEETING will be hold at tho Foresters' Hall for the purposo formine a MUTUALIMPROVEMENT-CLASS. Those favourable are invited to attend THIS (Thursday) EVENING

Fig. 4 Illustration of fine-tuning ALBERT on our public meeting information extraction task.

while reducing parameters through matrix factorization and cross-layer parameter sharing. We use version two of ALBERT, which is better than the first version. Compared to ALBERT v1, ALBERT v2 differs by using additional training data, no dropout, and longer training. ALBERT is then fine-tuned using our constructed dataset and a benchmark machine reading comprehension task dataset SQuAD 2.0 [17] for information extraction task. SQuAD 2.0 differs from SQuAD 1.0 [18] in that the answers to the questions may not be included in the text. It is considered to be appropriate in our task, which does not include answer spans in the articles for all questions.

Figure 4 illustrates the process of fine-tuning ALBERT on our information extraction task using public meeting data. The input consists of a question, a context, and special tokens, [CLS] and [SEP]. [CLS] token is added in front of every input, and [SEP] token is inserted at the end of both the question and context. Note that context corresponds to a public meeting article here. If the question can be answered, the model will output the continuous span of context that contains the answer.

The base version of ALBERT uses 12 stacked layers of bidirectional Transformer [27] encoder blocks as encoders. Different from BERT, instead of learning unique parameters for each of the 12 layers, ALBERT only needs to learn the parameters for the first block and reuse the block for the remaining 11 layers.

A Transformer encoder block consists of a multi-head self-attention layer and a fully connected feed-forward network. The multi-head self-attention mechanism performs the scaled dot-product attention multiple times in parallel. The outputs of the multi-head mechanism layer are then fed to the feed-forward network.

During fine-tuning, ALBERT introduces a start vector $S \in \mathbb{R}^H$ and an end vector $E \in \mathbb{R}^H$, where H is the hidden size of 768. The probability of word i being the start of the answer span is calculated by taking a dot product between S and the final hidden vector of ALBERT for i^{th} input token $T_i \in \mathbb{R}^H$, followed by a softmax function over all the words in the context:

$$P_i^S = \frac{e^{S.T_i}}{\sum_k e^{S.T_k}} \tag{2}$$

The probability of end of the answer span P_i^E is also calculated similarly by computing dot product between E and T_i followed by a softmax. The maximum scoring span (P_i^S, P_j^E) where $i \leq j$ is then used as the prediction. For questions that do not have an answer, we treat those questions as having an answer span with start and end at the [CLS] token. We predict the answer is null if the score of no-answer span is higher than the score of the best non-null span. Otherwise, the best non-null span will be predicted as the answer span. The model is trained with the objective to minimize the cross-entropy loss of the start and end span positions.

5 Experiments

5.1 Experimental Settings

In our experiment, we used both of the SQuAD 2.0 datasets (consisting of two sets of data: $130,319$ question-answer pairs for training and $11,873$ question-answer pairs for evaluation) and our public meeting dataset (Table 1). As the amount of our dataset is small and thus the change in accuracy due to data splitting is considered to be large. Therefore, the evaluation was conducted by using 5-fold cross-validation on our dataset.

We used the base model of ALBERT. The number of ALBERT dimensions was 768, and the number of layers was 12. For optimization, we used Adam [9] with a learning rate of $3e$-5 and batch size of 8 over 3 epochs.

We used two different metrics to evaluate the model performance similar to SQuAD 2.0:

- Exact Match, which indicates the ratio of cases where the extracted phrase is perfectly matched with the ground truth.
- F1, which indicates the ratio of matched spans between the prediction and ground truth. F1 can be defined as:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

  Precision is the ratio of correctly predicted spans to the whole predicted spans. Recall is the ratio of correctly predicted spans to the gold answer spans. Furthermore, we also use F1 (Has Ans) when the text contains the extraction target and F1 (No Ans) when the text does not contain the extraction target.

Table 2 Information extraction experiment results. * indicates that the difference compared to ALBERT+SQuAD is statistically significant ($p < 0.05$) using the k-fold cross-validated t test [4]. ** indicates that the difference compared to both ALBERT+SQuAD and ALBERT+Merge is statistically significant.

	Exact Match	F1	F1 (Has Ans)	F1 (No Ans)
ALBERT+SQuAD	42.18	48.82	24.11	62.78
ALBERT+Merge	80.90	86.71	73.80	93.84
ALBERT+Series	**81.61***	**87.47****	**74.98***	**94.41***

Table 3 Information extraction results for each question.

Setting	Question	Exact Match	F1	F1 (Has Ans)	F1 (No Ans)
ALBERT+SQuAD	q1	28.92	45.90	35.59	63.31
	q2	16.10	27.13	18.81	38.82
	q3	40.08	47.00	25.38	65.36
	q4	41.91	42.35	19.97	43.39
	q5	81.20	81.33	2.77	88.65
	q6	44.86	49.19	14.38	67.23
ALBERT+Merge	q1	69.56	79.95	80.67	78.48
	q2	75.30	84.28	81.57	87.93
	q3	67.49	79.08	63.37	90.58
	q4	96.97	97.18	31.87	99.34
	q5	91.63	92.27	20.19	98.82
	q6	84.46	87.51	75.51	92.95
ALBERT+Series	q1	70.60	81.34	81.30	81.02
	q2	77.05	84.79	82.72	87.16
	q3	67.01	79.04	62.34	91.49
	q4	97.77	97.95	39.45	99.68
	q5	91.71	92.72	26.19	98.79
	q6	85.50	88.96	77.56	94.21

In our experiments, we compared the following three settings:

- ALBERT+SQuAD: an ALBERT model that was fine-tuned only with SQuAD 2.0;
- ALBERT+Merge: an ALBERT model that was fine-tuned on the data that merge the SQuAD 2.0 dataset and our dataset;
- ALBERT+Series: an ALBERT model that was first fine-tuned on the SQuAD 2.0 dataset and then further fine-tuned on our dataset.

6 Results

Table 2 shows the experiment results of information extraction. We can see that ALBERT+SQuAD does not perform well due to the domain difference between SQuAD and our dataset, which are Wikipedia articles. ALBERT+Merge and ALBERT+Series significantly outperform ALBERT+SQuAD, indicating the effectiveness of fine-tuning on the in-domain data. ALBERT+Series performs better than ALBERT+Merge on all of the evaluation metrics. We think the reason for this is that the last fine-tuning stage focuses on the in-domain task.

Table 3 shows the results of information extraction for each question for the three different settings. ALBERT+SQuAD generally achieved low F1 (Has Ans) scores in all of the questions, especially in q5. In the ALBERT+Merge and ALBERT+Series settings, q4 and q5 have lower F1 (Has Ans) compared to the other

Table 4 Examples of extracted information.

Context	I hereby convene a PUBLIC MEETING oHhe Landholders and Householders in the Road District of Pembroke to be holden for the purposes stated in the said Requisition at the Molding Star Inn the llth day of October now next ensuing at the hour of 11 o'clock in the forenoon-T. l Assistant Police Magistrate of ' Spring Bay in the Road 4480 District of Pembroke.			
Question	ALBERT + SQuAD Answer	ALBERT + Merge Answer	ALBERT + Series Answer	Gold Answer
q1	N/A	the llth day of October now next ensuing at the hour of 11 o'clock in the forenoon-T.	the the llth day of October now next ensuing at the hour of 11 o'clock in the forenoon	the the llth day of October now next ensuing at the hour of 11 o'clock in the forenoon
q2	in the Road District of Pembroke	at the Molding Star Inn	at the Molding Star Inn	at the Molding Star Inn
q3	N/A	for the purposes stated in the said Requisition	for for the purposes stated in the said Requisition	for the purposes stated in the said Requisition
q4	I hereby convene a PUBLIC MEETING oHhe Landholders and Householders in the Road District of Pembroke	N/A	N/A	N/A
q5	N/A	N/A	N/A	Assistant Police Magistrate of ' Spring Bay
q6	N/A	N/A	Landholders and Householders in the Road District of Pembroke	Landholders and Householders in the Road District of Pembroke

questions but have higher F1 (No Ans). This might have occurred due to the data sparseness of q4 and q5. Table 1 shows both q4 and q5 have fewer data compared to the other questions. This might lead to a lot of questions that have answers predicted as no answer, thus lowering the F1 (Has Ans) while keeping the F1 (No Ans) high.

Table 4 shows an example of extracted information for the three different settings. In the ALBERT+SQuAD setting, the model incorrectly predicted all of the questions. The model predicted the wrong location for q2 and incorrectly predicted q4 as having an answer. In the ALBERT+Merge setting, the model predicted q5 and q6 incorrectly. The model predicted q5 and q6 as having no answer while there should be one. The predicted q1 does not exactly match the gold answer since it extracted a slightly longer span compared to the gold answer. In the ALBERT+Series setting, the model correctly predicted all of the questions except for q5, which is predicted as having no answer. In all of the settings, the models failed to predict q5. The answer to q5 is usually a person's name or organization name, which is more challenging to answer, especially with the low number of training examples.

7 Conclusion

In this paper, we proposed a method for extracting information about the date and time, place, purpose, people who requested the meeting, people who convened the meeting, and people who were convened from public meetings in historical Australian newspapers. We constructed a dataset for information extraction in the public meeting domain and trained a machine reading comprehension model to extract the information. Experiments showed that our model achieves an F1 score of 74.98% with the proposed ALBERT+Series method for information extraction from public meeting articles. In the future, we plan to analyze the extracted information in a long time span to obtain new insights into Australian history.

Acknowledgements This work was supported by Grant-in-Aid for Scientific Research (B) #19H01330, JSPS.

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Cassidy, S.: Publishing the trove newspaper corpus. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 4520–4525. European Language Resources Association (ELRA), Portorož, Slovenia (2016). URL https://www.aclweb.org/anthology/L16-1715
2. Davies, M.: Expanding horizons in historical linguistics with the 400-million word corpus of historical american english. Corpora **7**, 121–157 (2012)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
4. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. **10**(7), 1895–1923 (1998). DOI 10.1162/089976698300017197. URL https://doi.org/10.1162/089976698300017197
5. Fujikawa, T.: Public meetings in new south wales: 1871-1901. Journal of the Royal Australian Historical Society **76**, 45–61 (1990)
6. Hermann, K.M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, p. 1693–1701. MIT Press, Cambridge, MA, USA (2015)
7. Hill, F., Bordes, A., Chopra, S., Weston, J.: The goldilocks principle: Reading children's books with explicit memory representations. In: Y. Bengio, Y. LeCun (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (2016). URL http://arxiv.org/abs/1511.02301
8. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Vancouver, Canada (2017)
9. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations p. 13 (2014)

10. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL '07, pp. 177–180 (2007). URL http://dl.acm.org/citation.cfm?id=1557769.1557821
11. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: RACE: Large-scale ReAding comprehension dataset from examinations. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 785–794. Association for Computational Linguistics, Copenhagen, Denmark (2017). DOI 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082
12. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
13. Neudecker, C.: An open corpus for named entity recognition in historic newspapers. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 4348–4352. European Language Resources Association (ELRA), Portorož, Slovenia (2016). URL https://www.aclweb.org/anthology/L16-1689
14. Ng, H.T., Teo, L.H., Kwan, J.L.P.: A machine learning approach to answering questions for reading comprehension tests. In: 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 124–132. Association for Computational Linguistics, Hong Kong, China (2000). DOI 10.3115/1117794.1117810. URL https://www.aclweb.org/anthology/W00-1316
15. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics **9**(1), 62–66 (1979). DOI 10.1109/TSMC.1979.4310076
16. Radford, A.: Improving language understanding by generative pre-training (2018)
17. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for SQuAD. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 784–789. Association for Computational Linguistics, Melbourne, Australia (2018). DOI 10.18653/v1/P18-2124. URL https://www.aclweb.org/anthology/P18-2124
18. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2383–2392. Association for Computational Linguistics, Austin, Texas (2016). DOI 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264
19. Richardson, M., Burges, C.J., Renshaw, E.: MCTest: A challenge dataset for the open-domain machine comprehension of text. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 193–203. Association for Computational Linguistics, Seattle, Washington, USA (2013). URL https://www.aclweb.org/anthology/D13-1020
20. Sekine, S.: Information extraction from texts (new fields of natural language processing techniques). Information Processing **40**(4), 370–373 (1999). URL https://ci.nii.ac.jp/naid/110002763931/
21. Seo, M.J., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. In: Proceedings of the International Conference on Learning Representations (2017)
22. Smith, R.: An overview of the tesseract ocr engine. In: Ninth International Conference on Document Analysis and Recognition, vol. 2, pp. 629–633 (2007). DOI 10.1109/ICDAR.2007.4376991
23. Suzuki, S., Abe, K.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing **30**(1), 32–46 (1985). DOI https://doi.org/10.1016/0734-189X(85)90016-7
24. Tanaka, K., Chu, C., Kajiwara, T., Nakashima, Y., Takemura, N., Nagahara, H., Fujikawa, T.: Corpus construction from historical newspapers with OCR error correction. In: Proceedings of the 26th Annual Meeting of the Association for Natural Language Processing, pp. 653–656 (2020)
25. Tanaka, K., Chu, C., Ren, H., Renoust, B., Nakashima, Y., Takemura, N., Nagahara, H., Fujikawa, T.: Constructing a public meeting corpus. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 1934–1940. European Language Resources Association, Marseille, France (2020). URL https://www.aclweb.org/anthology/2020.lrec-1.238

26. Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., Suleman, K.: NewsQA: A machine comprehension dataset. In: Proceedings of the 2nd Workshop on Representation Learning for NLP, pp. 191–200. Association for Computational Linguistics, Vancouver, Canada (2017). DOI 10.18653/v1/W17-2623. URL https://aclanthology.org/W17-2623
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008. Curran Associates, Inc. (2017)