Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 1934–1940
Marseille, 11–16 May 2020

Constructing a Public Meeting Corpus

Koji Tanaka[†], Chenhui Chu[‡], Haolin Ren[*], Benjamin Renoust[‡], Yuta Nakashima[‡], Noriko Takemura[‡], Hajime Nagahara[‡], Takao Fujikawa[**]
[†] Graduate School of Information Science, Osaka University, [‡] Institute for Datability Science, Osaka University,
[*] National Institute of Informatics, [**] Graduate School of Letters, Osaka University
tanaka.koji@ist.osaka-u.ac.jp,
{chu, renoust, n-yuta, takemura, nagahara}@ids.osaka-u.ac.jp,
haolinren@gmail.com
fuji@let.osaka-u.ac.jp

Abstract

In this paper, we propose a method for constructing a large corpus about a century of *public meetings* in historical Australian newspapers, and analyze the constructed corpus. The corpus construction method is based on image processing and Optical Character Recognition (OCR). We digitize and transcribe texts of the specific topic of public meeting. Experiments show that our proposed method achieves a F-score of 71.5% with a high recall of 97.5% for corpus construction. This allows us to feed a content search tool for temporal and semantic content analysis.

Keywords: corpus construction, public meeting, content delivery

1. Introduction

Large-scale text corpora are essential for natural language processing (NLP). Most existing corpora are created from text that has already been digitized. For instance, the benchmark syntactic parsing dataset *Penn Treebank* (Marcus et al., 1993) is created by labelling part-of-speech tags and syntactic information on the digitized text from the Wall Street Journal newspapers. The parallel corpus *Europarl* (Koehn, 2005) that has been used for the machine translation shared task workshop, Workshop on Statistical Machine Translation (WMT), is created by aligning parallel sentences from the digitized multilingual European Parliament data.

In various fields including literature and humanities, many materials to be studied are not digitized, which are stored in a physical medium such as paper or just scanned but not transcribed into text. By digitizing and transcribing such materials into text, and structuring them via extracting specific topics, we can apply many NLP techniques to analyzing them automatically. In research fields such as literature, digitization, text transcript and structure can significantly increase the value of the original materials.

In this paper, we work on the historical newspaper database Trove (Cassidy, 2016)[1] (Trove covers major Australian daily newspapers and local newspapers). We propose a corpus construction method based on image processing and Optical Character Recognition (OCR), which achieves a F-score of 71.5% with a high recall of 97.5%. We first identify the rule lines in newspaper images and trim the images into newspaper articles. Next, we apply OCR to the trimmed articles, and extract the articles with specific topic words. Evaluation conducted on manually annotated golden data indicates that the proposed method can extract 14.9% more articles without excess and deficiency, compared to a baseline that is based on linguistic features extracted from beginning and ending sentences of articles.

We extract articles about the specific topic of *public meeting* (Fujikawa, 1990),[2] which covers 120 years spanning from 19th to 20th centuries. 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. As a result, we develop a tool for content delivery so that we can search and visualize the content of the public meeting articles in semantics and time. Although the proposed method is focused on newspaper data, it is independent from periods and languages and thus can be applied to the corpus construction for historical newspaper data other than Trove.

2. Related Work

We first present previous studies that construct historical corpora, and then we describe about works that analyze the content of a corpus via search.

2.1. Historical Corpus Construction

Several studies on corpus construction for historical documents have been conducted. Davies (2012) built an American English historical corpus. They collected text from magazines, newspapers and books from 1810 to 2000. They further lemmatised and labeled part-of-speech (POS) tags on the corpus. Rögnvaldsson et al. (2012) built an Icelandic parsed historical document corpus. They collected text from the 12th to the 21st century and annotated them for parsing using the same schema as Penn Treebank (Marcus et al., 1993). Sánchez-Martínez et al. (2013) built a Spanish historical corpus. They collected text from prose, theatre, and verse from 1481 to 1748, lemmatised them, and labeled POS tags. Neudecker (2016) built a corpus for named entity recognition from historical newspapers in

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

[2] Public meetings is the main pillar of public opinion formation for Western Europe in the 19th century.

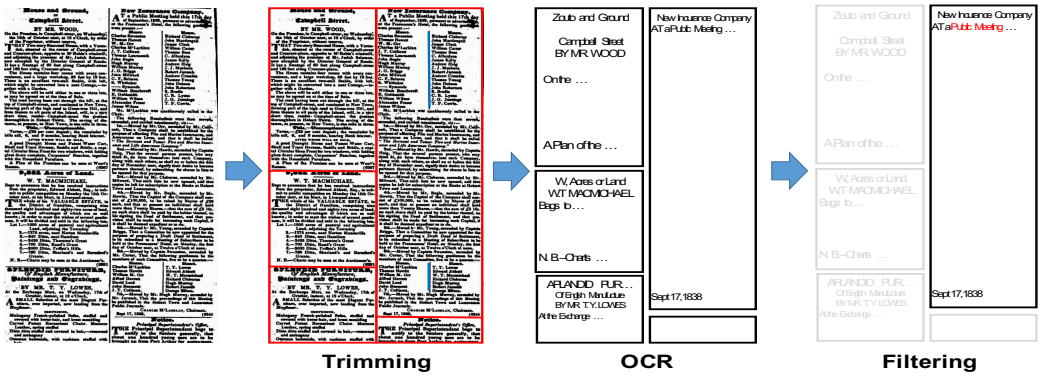

Figure 1: Overview of the corpus construction method.

French, Dutch, and German. They annotated named entity tags for the *Europeana Newspaper* from the 17th to the 20th century using the INL Attestation Tool.[3] Cassidy (2016) built an Australian historical newspaper corpus and published it on a website called Trove. They converted newspapers from the 19th century to the 21st century into text data using OCR. We construct our corpus based on Trove. Different from (Cassidy, 2016) and other previous studies, we propose a method to extract the specific topic of "public meeting" and analyze the temporal and semantic contents.

2.2. Search-based Content Analysis

Search result clustering often ends with *visualization* (Carpineto et al., 2009) to help users target their own search (Käki, 2005). A few search engines visually provide their results (Selberg and Etzioni, 1995; Ferragina and Gullì, 2004; Zhang and Dong, 2004; Koshman et al., 2006) to represent meaningful topics, but most of them focused at best onto cognitively costing file-hierarchy type of visualization. Studies have also tried to organize results spatially (Gomez-Nieto et al., 2014), or even by treemaps (Nocaj and Brandes, 2012). While they are good for coordination and contextualization, they do not provide space efficiency. Tag clouds also provide a good overview of the semantic space (Sinclair and Cardew-Hall, 2008) sometimes with hierarchy (Xu et al., 2016) and interaction (Renoust et al., 2015). Ren et al. (2018) use tag clouds to propose an integrated methodology for interactive exploration and search refinement. Although it is designed to support all different search tasks (Brehmer and Munzner, 2013), it still is limited in the amount of data it can handle. We extended this approach to handle a large quantity of data.

3. Corpus Construction

The overview of our proposed corpus construction method is shown in Figure 1. Because the OCR text provided by Trove lacks the rule line information, it is difficult to extract only "public meeting" article accurately. Therefore, we propose a method to address this problem by detecting rule lines from the image. 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 for the articles. Finally, we filter the articles with a query phrase to filter the articles and thus extract only the target articles that we are interested.

3.1. Trimming

We use OpenCV[4] to identify the rule lines in newspaper images and for trimming. First, we binarize the newspaper images using the method proposed by Ohtsu (Otsu, 1979). The binarization method transfers grayscale images to white-black 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 (Suzuki and Abe, 1985) 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 with a width below a threshold are identified as a column. Areas with a width above a threshold and with a height below a threshold are identified as an article split in the newspaper image. The thresholds are tuned manually. After that, we can finally trim the article images accordingly.

There are small columns in articles as the blue lines shown in the sub-figure "trimming" of Figure 1. To deal with this, we propose the following method to determine the vertically split column. Firstly, we trim the column with the x coordinate (horizontal direction) value. We then compare the minimum and maximum y coordinate (vertical direction) values with the newspaper coordinate value. If the difference is above a predefined threshold, we determine it as a small column and do not use it for trimming.

3.2. OCR

OCR is generally performed following the procedures of character delimiter recognition, size normalization, feature extraction, and classification. Google open-sources the OCR method Tesseract (Smith, 2007), which achieves 98.4% and 97.4% on newspaper articles in character and word level, respectively. However, when comparing OCR accuracy from Google Drive[5] to Tesseract, Google Drive

[3] https://github.com/INL/AttestationTool

[4] https://opencv.org/

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

performs best. Therefore, we use the OCR function of Google Drive for extracting text from the article images.

3.3. Filtering

We filter the OCRed articles that are not our target with a query phrase, leaving the target articles to be extracted. In order to allow the error of character recognition by OCR, we define similarities in character level. We use the Python *difflib* module SequenceMatcher[6] for calculating similarities. In SequenceMatcher, the similarities between a character string pair is defined as:

$$Similarity = \frac{2.0 \times M}{T}, \quad (1)$$

where M is the number of matched characters and T is the sum of character numbers in the character string pair.
We get 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 as target articles with the highest similarity above a threshold.The threshold is tuned on a development set, which shows the highest F-score.

4. Experiments

4.1. Data

We manually created the ground-truth data for "public meeting" articles, in order to evaluate the accuracy of article extraction. We used the newspaper image data crawled from Trove. Trove is an online library database service maintained by the Australian government. We searched the key phrase "public meeting" on Trove to get the newspaper IDs of our targetted articles. Next we obtained newspaper *pdf* data through the API provided by Trove from the newspaper IDs. Newspaper *pdf* files are converted to *PNG* images using ImageMagick[7].
In our experiments, we manually extracted 307 articles about "public meeting" spanning from 1838 to 1954, and split them into 149 and 158 articles for development and testing, respectively. Figure 2 shows the line number distribution of the golden data used for our evaluation. We can see that most golden articles contain less than 60 lines, but there are also exceptions.

4.2. Comparison

We compared the following methods in our experiments:

- **Baseline**: We compared a baseline method, which is based on text features to identify articles directly from the OCRed text provided by the Trove website. The baseline method extracts features from the beginning and ending sentences of articles for article identification. The features are as follows:
  - Beginning sentence: Take 2 sentences before the sentence that contains "public meeting."
  - Ending sentence: We first apply named entity recognition using the Stanford parser.[8] Then we take the sentence containing LOCATION, DATE, PERSON tags, but the following sentence that does not contain these tags as the ending sentence.
- Proposed: This is our proposed method presented in Section 3.
- Baseline+Proposed: Use our proposed method for articles which the baseline model fail to extract because the ending sentence corresponding to the beginning sentence is not found.

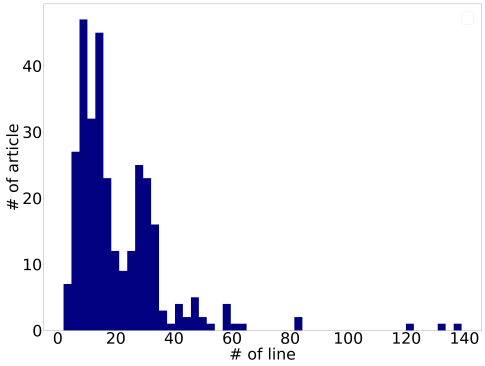

Figure 2: Line number distribution of the golden article data.

4.3. Parameter Tuning

To tune the thresholds for the rule line and small column identification described in Section 3.1., we used the newspaper data spanning in one month and determined the thresholds empirically. For the threshold used for filtering as described in Section 3.3., we tuned it on the development data and chose the one achieving the highest F-score. We tuned the threshold from 0 to 1 with an increment of 0.05, and it turned out that 0.8 was the best and thus we used the threshold of 0.8 for filtering.

4.4. Evaluation Methods

In our experiments, we conducted article level evaluation to evaluate if the articles are successfully extracted. In addition, we also conducted line level evaluation to evaluate the accuracy for article extraction. These two evaluation methods are described as follows:

Article level evaluation method

We calculated the similarity following Equation (1) between the sentences containing the keyword "public meeting" in the extracted article and golden article, respectively. If the similarity is higher than a threshold then the extraction is evaluated as success, otherwise it is failure. The threshold was empirically determined to be 0.6. Then we calculated the precision, recall, and F-score for the baseline and proposed method.

[6] https://docs.python.jp/3/library/difflib.html
[7] https://www.imagemagick.org/
[8] https://nlp.stanford.edu/software/lex-parser.shtml

Method	Precision	Recall	F-score
Baseline	**76.1**	56.3	64.7
Proposed	59.4	51.9	55.4
Baseline+Proposed	56.4	**97.5**	**71.5**

Table 1: Article extraction evaluation results.

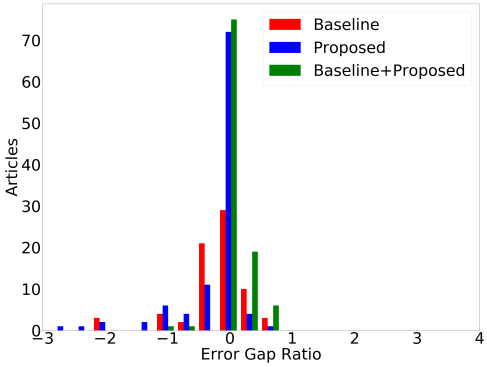

Figure 3: Line level evaluation results (beginning line).

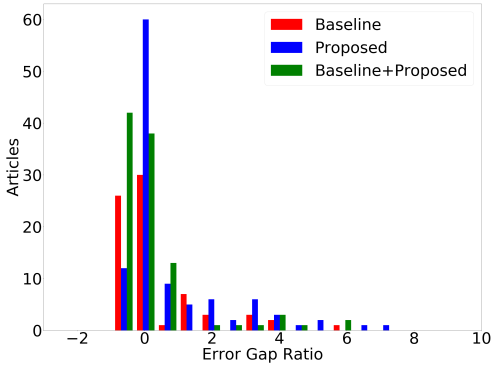

Figure 4: Line level evaluation results (ending line).

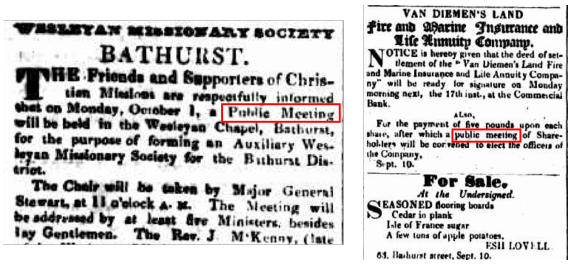

Figure 5: Article extraction examples

Line level evaluation method

We compared the beginning and ending lines of the extracted articles to the golden articles to investigate the difference. Then we calculated the ratio of excess and deficiency lines between the extracted and golden articles.

4.5. Results

Article level evaluation

Table 1 shows the results for article level evaluation. We can see that Baseline has a higher F-score than Proposed. The reason for this is that Baseline uses the feature of a sentence that includes "public meeting" when getting the first lines of the article, and thus the sentence used for article level evaluation is extracted. However, there are still some failures in the extraction of Baseline. This is because firstly there can be multiple public meeting articles in a newspaper image, secondly there are OCR errors about the keyword "public meeting." We can also see that Proposed has low precision considering that filtering component in Proposed is based on similarity with the keyword "public meeting." This is because the extracted text by Proposed is OCRed by Google Drive, however the gold text is OCRed by Trove. It is considered that this difference causes that the similarity between extracted text by Proposed and gold text is lower, and precision is decreased. After combing our proposed method with the baseline method, the article extraction results are improved significantly.

Line level evaluation

Figures 3 and 4 show the line level evaluation results for the beginning and ending lines, respectively. The horizontal axis represents the gap ratio of the number of the excess and deficiency lines against the entire number of lines in an article. The vertical axis represents the number of articles. We can see that on both the beginning and ending lines, Proposed extracted significantly more articles without excess and deficiency than Baseline. In addition, for the case of articles without excess and deficiency in both the beginning and ending lines, Baseline only successfully extracted 5 (3.1%) articles but Proposed extracted 19 (18.0%) articles. Therefore, we can say that the proposed method that uses visual features to identify the article split, is more effective for extraction articles with specific topics.

4.6. Discussion

Figure 5 (left) shows an example of an article that failed to be extracted. The "public meeting" area surrounded by the red rectangle in the image was incorrectly recognized as "Pohlle Meeling" by OCR. Therefore this article was not identified as a target article during filtering and thus failed to be extracted. This happens because the resolution of this newspaper article was lower than others, and thus OCR failed. Among all the test data, 8.1% of the articles were failed to be extracted due to OCR errors.

Figure 5 (right) shows an example of an article that was successfully extracted but with some excess. We can see that the trimmed newspaper image contains not only a target article but also an article that is not our target. The reason for this is that the ruled line has been cut off in the middle, which makes the identification of the article split fail, leading to the improper trimming. Among all the test data, 26.7% of the articles were successfully extracted but with some excess like this example.

There were also extracted articles with some deficiency compared to the golden articles. One reason for this is that the trimming algorithm treated some paragraph splits as article splits. There is 8.1% of this type of error among the test data. In addition, the proposed method cannot deal with an article spanning multiple columns, making this type of

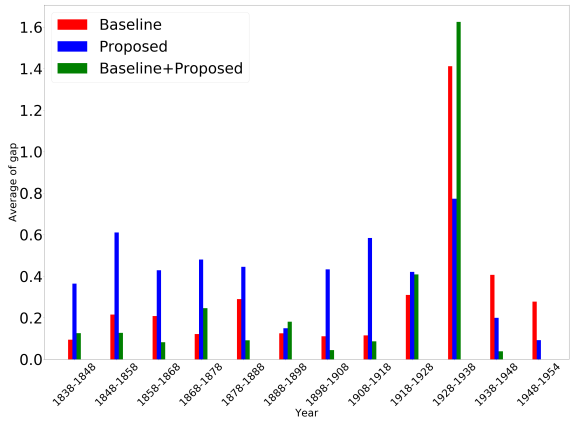

Figure 6: The average of the line level evaluation results in each year.

article being split into multiple articles. There is 1.9% of this type of error among the test data.

Figure 6 shows the average of the line level evaluation results (we merged the gaps in the beginning and ending lines) in each year (divided from 1838 to 1954 every 10 years). We can see that the average of gap ratio is large from 1928 to 1938. This is because, the variation of newspaper design increases in that period and the accuracy of our trimming method decreases.

5. Content Delivery

Using the "Baseline+Proposed" method, we extracted $269,044$ public meeting articles spanning from 1838 to 1954 from Trove. To provide exploration of the content, we extended the search engine Visual Cloud (Ren et al., 2018) so it may support a much larger number of documents to search. This was done by moving indexing on the server side rather than client, the server is freely available online.

We extract named entities from each article using Stanford CoreNLP.[9] Each article represents then a document, which is indexed by its list of named entities. Still a few noisy entities remained as a result of the OCR mistakes, as a result, many looked like acronyms, or very long expressions. Hence we removed all acronyms, two-letter words, and expressions over 50 characters. To optimize the visualization process, we removed the entities occurring less than 5 times in the dataset, and all documents annotated with only one word. As a result, we obtained a subset of 171,319 documents, annotated with 11,589 words of which the most occurring words are Melbourne and Sydney with both 13,815 and 12,384 occurrences, respectively. Although we loose a fair share of OCRed documents, these were noisy enough to be unexploitable from a NER standpoint, and the distributions remain roughly similar as illustrated in Figure 7.

The Visual Cloud provides a full search engine, with an interactive timeline and tag cloud, as illustrated in Figure 8. Upon a query, here "*horses*," the results are placed on a timeline, a word cloud describes the semantic content of the search results. Individual access to each result below, the result list may be sorted according to a given criteria. The Visual Cloud is built on top of a keyword co-occurrence graph of the search results, which enables the computation of hierarchical clusters of keywords (Ren et al., 2018) (in color in Figure 8). It is enriched with an optional heatmap (see Figure 9) that highlights neighbors and their influence based on their mutual co-occurrences (Ren et al., 2018). A click on a keyword may filter the result list. The timeline is also interactive to show the relative occurrence of the keywords over time (see Figure 10).

In our example, we searched *"horses"* between 1838 and 1950. It suggests that they were an important topic until the early 1880's (although we should remember it is influenced by the limited number of keywords per article after this date). Six clusters are presented, and most of the keywords are locations in Australia (Figure 8). From the heatmaps in Figure 9, it seems that *"sweepstakes"* taking place in *"Muswellbrook"* have been discussed, and *"London"* has been associated with a lot of places, and *"jockey club,"* probably in the context of racing. We may further notice from the timeline in Figure 10 that, although part of the early 1840-1860 period, the *"jockey club"* disappears from the discussion in the late 1930-1950 period.

6. Conclusion

In this paper, we constructed a corpus of public meeting articles via image processing and OCR. Experiments conducted on the newspaper data from Trove indicated that we can successfully extract 97.5% of the targeted articles and 18.0% of the extracted articles are without excess and deficiency. We further enabled content delivery of the public meeting articles through search and visualization. The visualization clearly illustrates the gaps from data and processing, suggesting us to focus our next efforts in the keyword extraction, beyond NER processing, and include topic modelling. In the long run, we wish to apply and verify our methods to historical materials in other fields.

7. Acknowledgement

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

8. Bibliographical References

Brehmer, M. and Munzner, T. (2013). A multi-level typology of abstract visualization tasks. *IEEE TVCG*, 19(12):2376–2385.

Carpineto, C., Osiński, S., Romano, G., and Weiss, D. (2009). A survey of web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):17.

Cassidy, S. (2016). Publishing the trove newspaper corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4520–4525, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Davies, M. (2012). Expanding horizons in historical linguistics with the 400-million word corpus of historical american english. *Corpora*, 7:121–157.

Ferragina, P. and Gullì, A., (2004). *Knowledge Discovery in Databases: PKDD 2004*, chapter The Anatomy of SnakeT: A Hierarchical Clustering Engine for Web-Page

[9] https://stanfordnlp.github.io/CoreNLP/index.html

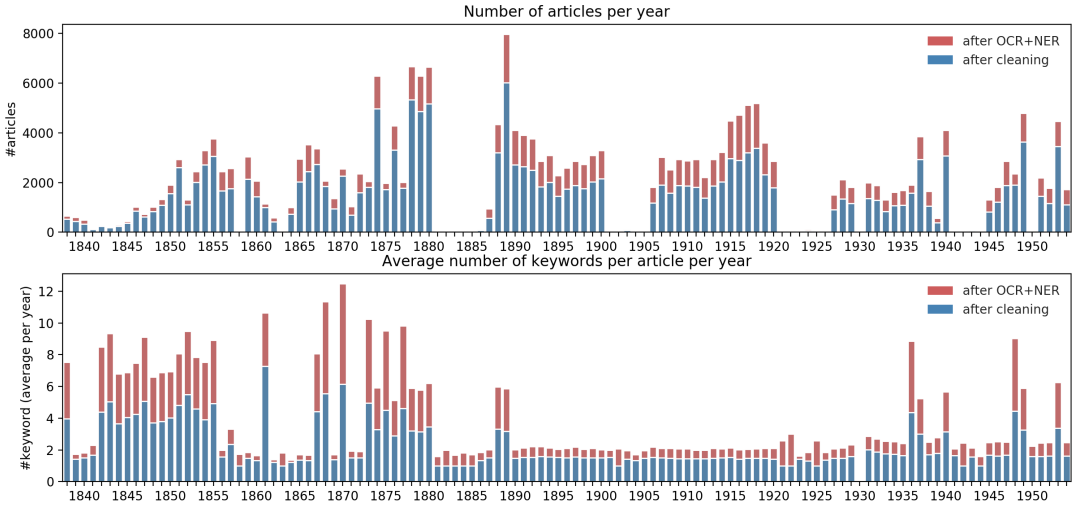

Figure 7: Distribution of the number of articles and of the average number of keywords (*i.e.,* entities detected by NER) per article, before and after cleaning the keywords for visualization.

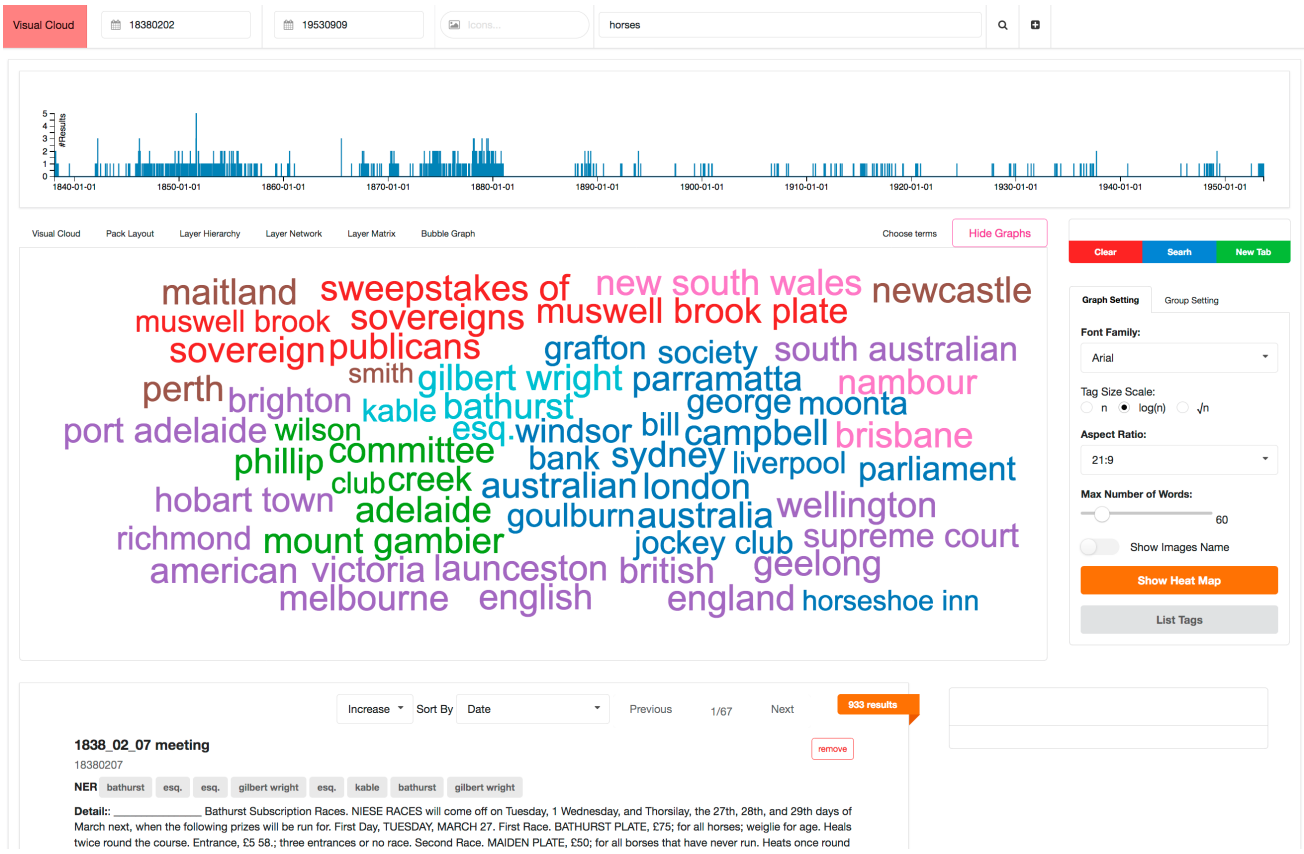

Figure 8: Search and visualization of the public meeting corpus, with the keyword “horses.”

Snippets, pages 506–508. Springer Berlin Heidelberg, Berlin, Heidelberg.

Fujikawa, T. (1990). Public meetings in new south wales: 1871-1901. *Journal of the Royal Australian Historical Society*, 76:45–61.

Gomez-Nieto, E., San Roman, F., and et al. (2014). Similarity preserving snippet-based visualization of web search results. *IEEE TVCG*, 20(3):457–470.

Käki, M. (2005). Findex: Search result categories help users when document ranking fails. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 131–140.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of Machine Transla-*

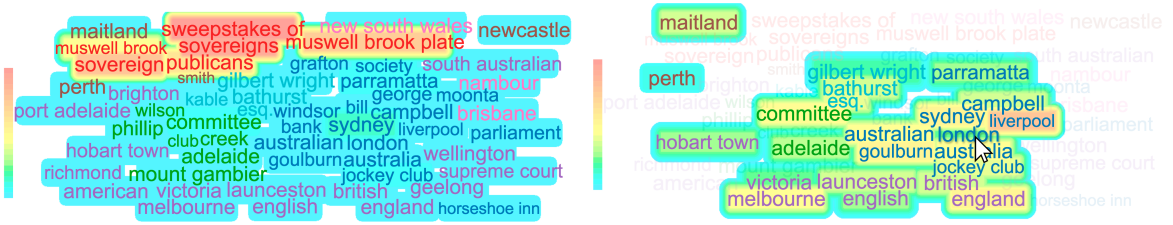

Figure 9: The interactive heatmap showing the relative degree of nodes in the co-occurrence graph, *i.e.,* influence of keywords in relation to the search (left) / other keywords (right).

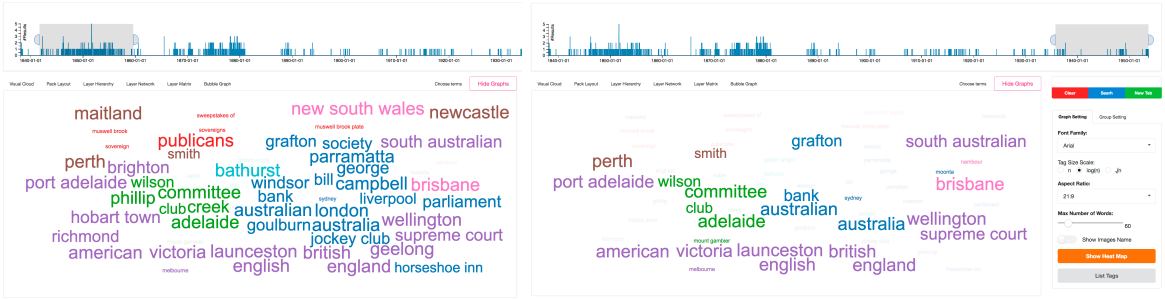

Figure 10: The interactive timeline showing the occurrence of the keywords in the subset (*e.g.,* left: first 20 years, right: last 20 years).

tion Summit, pages 79–86.

Koshman, S., Spink, A., and Jansen, B. J. (2006). Web searching on the vivisimo search engine. *Journal of the American Society for Information Science and Technology*, 57(14):1875–1887.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, Jun.

Neudecker, C. (2016). An open corpus for named entity recognition in historic newspapers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4348–4352, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Nocaj, A. and Brandes, U. (2012). Organizing search results with a reference map. *IEEE TVCG*, 18(12):2546–2555.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan.

Ren, H., Renoust, B., Viaud, M.-L., Melançon, G., and Satoh, S. (2018). Generating âvisual cloudsâ from multiplex networks for tv news archive query visualization. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE.

Renoust, B., Melancon, G., and Munzner, T. (2015). Detangler: Visual analytics for multiplex networks. In *Computer Graphics Forum*, volume 34-3, pages 321–330. Wiley Online Library.

Rögnvaldsson, E., Ingason, A. K., Sigursson, E. F., and Wallenberg, J. (2012). The Icelandic parsed historical corpus (IcePaHC). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1977–1984, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Sánchez-Martínez, F., Martínez-Sempere, I., Ivars-Ribes, X., and Carrasco, R. (2013). An open diachronic corpus of historical spanish. *Language Resources and Evaluation*, 47, 06.

Selberg, E. and Etzioni, O. (1995). Multi-service search and comparison using the metacrawler. In *In Proceedings of the 4th International World Wide Web Conference*, pages 195–208.

Sinclair, J. and Cardew-Hall, M. (2008). The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15–29.

Smith, R. (2007). An overview of the tesseract ocr engine. In *Proc. of International Conference on Document Analysis and Recognition*, volume 2, pages 629–633, Sep.

Suzuki, S. and Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46.

Xu, J., Tao, Y., and Lin, H. (2016). Semantic word cloud generation based on word embeddings. In *PacifVis,*, pages 239–243.

Zhang, D. and Dong, Y., (2004). *Advanced Web Technologies and Applications: 6th Asia-Pacific Web Conference, APWeb 2004*, chapter Semantic, Hierarchical, Online Clustering of Web Search Results, pages 69–78. Springer Berlin Heidelberg, Berlin, Heidelberg.