Relevance-specific clustering in predictive coding

William Webber, John Tredennick

Merlin Search Technologies, Denver USA

Abstract

Predictive coding is the most successful application of machine learning to e-discovery thus far. Static clustering methods are also widely implemented, but their usefulness in e-discovery practice is questionable. In this paper, we describe the use of relevance-specific clustering to present the results of predictive coding in a thematic way. We outline possible extensions, in which cluster similarity is directly informed by the predictive model. We also suggest that relevance-specific analytics should be more widely integrated with static analytics in e-discovery systems.

Keywords

TAR, predictive coding, clustering, e-discovery, relevance

1. Introduction

Interactive text classification using active learning, known as predictive coding, is to date the most successful application of machine learning and artificial intelligence technologies to e-discovery. In the early days of e-discovery, many other ML techniques were tried out. One of these, text clustering, is still widely deployed in e-discovery systems. Most commonly, this takes the form of a statically-generated hierarchical cluster over the full collection, often displayed and navigated graphically as an expandable cluster wheel.

Such hierarchical cluster UIs make for visually impressive demonstrations, particularly on suitably curated and ontologically decomposable collections (such as Wikipedia). Their usefulness in assisting the real tasks of investigation and review, however, is debatable. Heterogeneous enterprise document collections are difficult to ontologize, and even when meaningful clusters can be identified, they often will not divide documents along distinctions that are useful to the user's conception of relevance.

In this paper, we propose what we believe to be a novel application of clustering to predictive-coding–based technology-assisted review (TAR) and technology-assisted investigation (TAI): namely, relevance-specific clustering. Rather than statically clustering the whole collection, this approach dynamically clusters document sets that the predictive model identifies as likely relevant. Within an active learning process, the actively-selected suggested documents are presented to the user in clustered form, and the clustering dynamically updates with the predictive model.

ALTARS '22: 1st Workshop on Augmented Intelligence for Technology-Assisted Review Systems: Evaluation Metrics and Protocols for eDiscovery and Systematic Review Systems.

EMAIL: wwebber@merlin.tech (W. Webber); jt@merlin.tech (J. Tredennick)

URL: https://www.merlin.tech/about-us/team/ (J. Tredennick)

CEUR Workshop Proceedings (CEUR-WS.org)

The clustered presentation of predictive coding results enables several processing modes. First, the reviewer is frequently able to judge and code relevance for a whole cluster at a time, speeding up the review process. Second, similar documents can be reviewed together, improving the accuracy of the coding; and these clusters make a natural way of breaking up review work amongst multiple reviewers, with each reviewer focusing on a cluster of similar documents. Third, in investigations, iterative coding and clustering allows the investigator to pursue their line of inquiry from general to specific topical aspects.

We describe an initial implementation of relevance-specific clustering for predictive coding. We propose the extension of this approach to one in which not only are predicted-relevant documents clustered, but also the nature of the clustering (for instance, the similarity metric used) is influenced by the relevance model. Finally, we urge the broader use of relevance-aware analytics within e-discovery systems.

2. Previous work

[1] is to our knowledge the first researcher to propose that the results of an information retrieval system should be clustered for presentation to the user. [2] describe a "Scatter/Gather" method of cluster-based information retrieval, in which the collection is split into clusters (scatter); the user selects the clusters of interest to them (gather); the documents in these clusters are then re-clustered (scatter); and so on iteratively, until the desired documents are located. [3] employ Scatter/Gather on the results of a free-text search. [4] first referred to the clustering of search results as "query-specific clustering". [5] proposes query-sensitive clustering, and [6] implements and tests their own query-sensitive clustering method. [7] propose a repeated clustering method on search retrieval results to identify "dominant" documents for pseudo-relevance feedback.

3. Implementation

We describe the implementation of relevance-specific clustering within a TAR system that employs a continuous active learning (CAL) process (that is, an iterative active learning process in which the documents with the strongest prediction of relevance are the ones that are selected for review at each iteration) [8].

The interactive predictive coding process starts with one or more documents manually coded for relevance. These might be identified for instance, by a keyword or free-text search. At least one relevant document is required as a positive training example. If there are no or insufficient negative (irrelevant) training examples, we randomly sample pseudo-negative training examples from the collection, excluding documents that are similar to the positive examples [9]. Thus, the interactive process is able to begin with a single relevant example—a "more like this" query.

At each iteration, a predictive model is built using the current training examples. The feature set is of unigram terms, with normalized TF*IDF weights; the classifier is two-class (relevant versus irrelevant) logistic regression. Any classification technology

able to produce a ranking by predicted relevance could be used instead; we selected logistic regression due to the interpretability of feature weights under this model.

The predictive model is then used to assign a predictive relevance score to each unreviewed document in the collection, and the unreviewed documents are ranked by decreasing score. The top N documents are taken from the ranking (where, for instance, $N = 50$ might be suitable for an interactive investigation) as the set of suggested documents. The suggested documents are then clustered by similarity, again using the unigram TF*IDF feature set. While any non-hierarchical clustering method could be used, we use affinity propagation [10], as it is fast for a small number of documents—an important consideration for interactive use.[1] In addition, affinity propagation does not require the number of clusters to be specified in advance, but rather determines the number of "natural" clusters in the data.

Affinity propagation has an input parameter array (one value per document) known as the "preference" [10]. A document assigned a higher preferences is more likely to become a cluster exemplar. We, though, assign each document the same preference value. Additionally, a higher average preference value tends to increase the number of clusters created, even if each document receives the same value. In effect, this makes the clustering more sensitive to document differences. The default preference is the median of document similarities for the target set. Documents with a high relevance rank, though, can be very similar, which could lead to overly-sensitive clustering. For instance, near duplicate emails might be split into clusters based only on different recipients. To mitigate against over-sensitivity, we smooth the input preference between the median of the target set of high relevance-rank documents, and the background median similarity of the collection as a whole. It would also possible to allow the user manually to adjust cluster sensitivity, but we have not explored this option.

The clustered suggestions are then displayed to the user (Figure 1). Clusters are ranked by the decreasing relevance of their most relevant document, and documents within each cluster are ranked by decreasing relevance. Each cluster is labelled with the cluster's significant terms, along with snippets from the cluster's documents. These snippets are selected by taking the snippets from each document (generated by a standard snippet extraction method); clustering them; and taking the exemplar snippet for each cluster.

The user is then able to review the clusters and the documents within them. Clusters can be individually expanded (Figure 2) and collapsed. Documents within a cluster can be coded individually for relevance, or the cluster can be coded as a whole. The user is free to code as many or as few documents as they wish, in whichever clusters they choose. Their codings are then added to the training set, and the active learning process iterates.

4. Extensions

The clustering of suggestions described above uses the relevance model to determine which documents to cluster, and also how to rank the clusters and the documents within

[1] On the experimental machine, 50 email-length documents with an unfiltered unigram feature set are clustered in around 20ms.

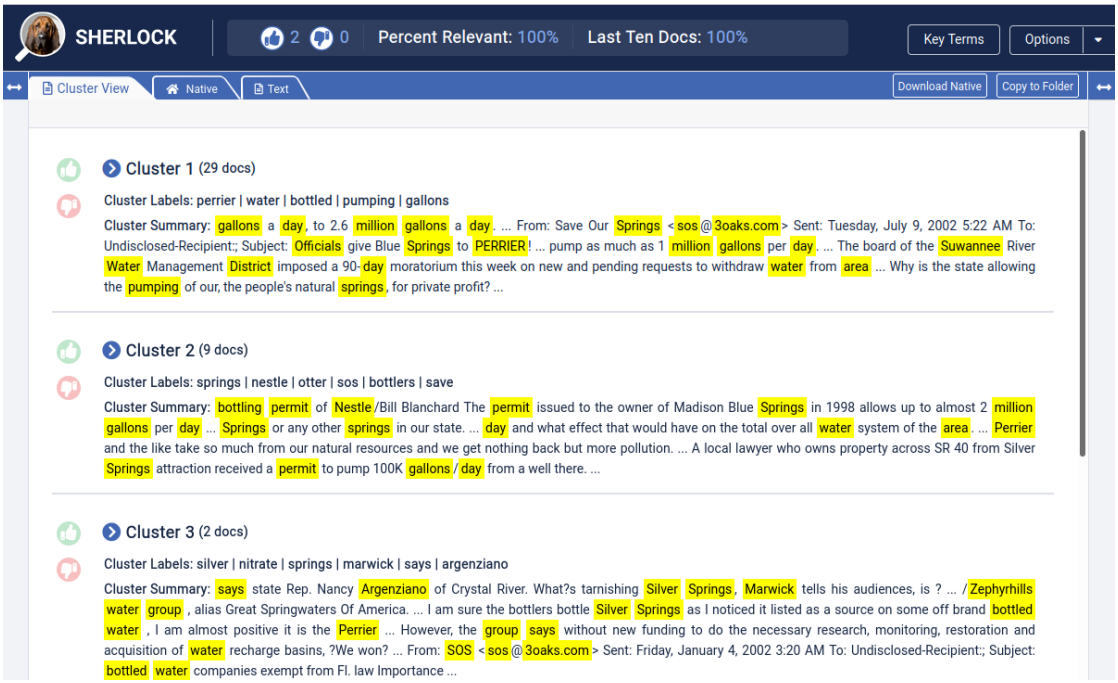

Figure 1: Cluster view, Jeb Bush collection, for "bottled water" investigation, with clusters unexpanded. Highlighted are terms that are significant for the cluster and/or have a high weight in the underlying predictive model.

each cluster. The focus on relevant documents helps ensure that the clustering is over documents that the user is interested in, and (we assert) makes it more likely that clusters will reveal distinctions that are more relevant to the user's review or investigation task. The predictive model, however, does not directly influence how clustering is performed. There has been previous work on query-sensitive clustering, in which the clustering is biased by the query [5, 6], for instance by giving more weight in the inter-document similarity metric to query terms (as opposed to query-specific clustering [4], where the query simply identifies the documents to be clustered). This could naturally be extended to the predictive model, which (under logistic regression at least) provides importance weights across the whole feature set. By reweighting the features in the similarity calculation using their importance in the predictive model, it may be that the resulting relevance-sensitive clustering will be more focused on distinctions that are meaningful to the user's conception of relevance.

More generally, it is our opinion that relevance sensitivity (whether based on a query or on a predictive model—or indeed on a combination of the two [11]) remains an under-utilized tool in e-discovery analytics. In this paper, we have presented an approach to relevance sensitivity for clustering, but similar approaches could be made to other forms of otherwise static analysis. For instance, relevance-specific or relevance-sensitive topic modelling could be helpful in identifying topical themes within the collection, as focused

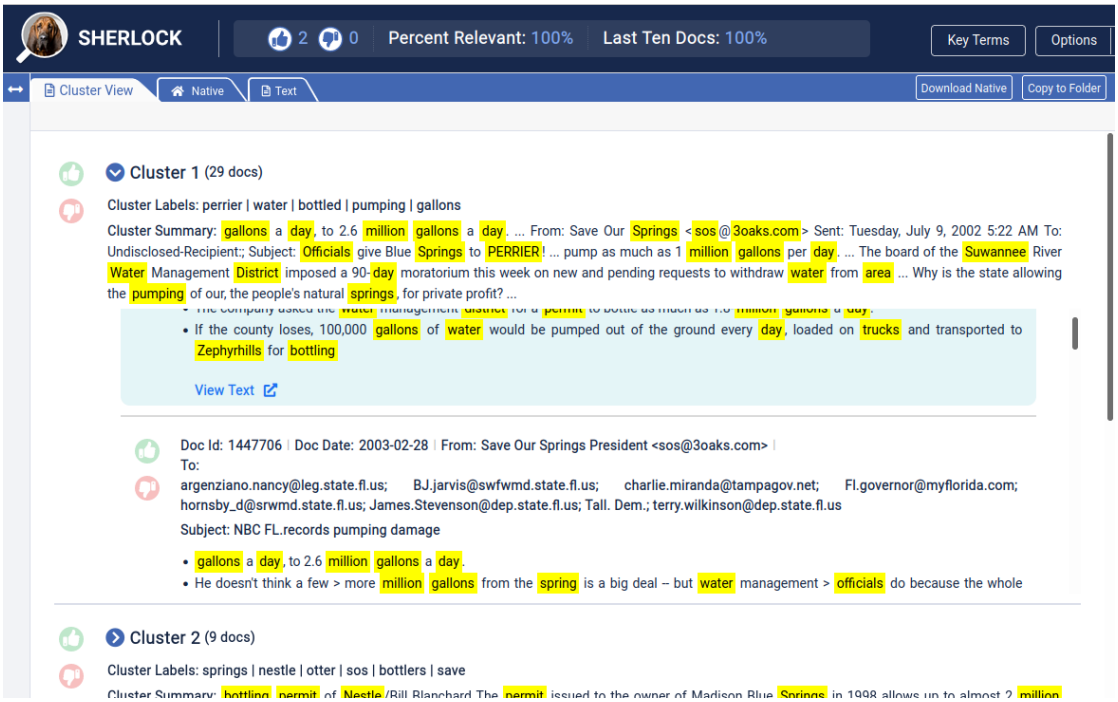

Figure 2: Cluster view, Jeb Bush collection, for "bottled water" investigation. Cluster 1 has been expanded for further investigation.

on the user's conception of relevance. Even simpler analytic tools, such as relevance ranking of custodians or date ranges, seem to us useful and under-explored in current e-discovery systems.

References

[1] S. E. Preece, Clustering as an output option, Proceedings of the American Society for Information Science 10 (1973) 189–190.

[2] D. R. Cutting, D. R. Karger, J. O. Pedersen, J. W. Tukey, Scatter/Gather: A cluster-based approach to browsing large document collections, in: N. J. Belkin, P. Ingwersen, A. M. Pejterson (Eds.), Proc. 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 1992, pp. 318–329.

[3] M. A. Hearst, J. O. Pedersen, Reexamining the cluster hypothesis: Scatter/Gather on retrieval results, in: and, and (Eds.), Proc. 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, 1996, pp. 76–84.

[4] A. Tombros, R. Villa, C. Van Rijsbergen, The effectiveness of query-specific hierarchic

clustering in information retrieval, Information Processing and Management 38 (2002) 559–582.

[5] M. Iwayama, Relevance feedbak with a small number of relevance judgments: Incremental relevance feedback vs. document clustering, in: E. Yannakoudis, N. J. Belkin, M.-K. Leong, P. Ingwersen (Eds.), Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 2000, pp. 10–16.

[6] A. Tombros, The effectiveness of query-based hierarchic clustering of documents for information retrieval, Ph.D. thesis, University of Glasgow, Glasgow, UK, 2002.

[7] K. S. Lee, W. B. Croft, J. Allan, A cluster-based resampling method for pseudo-relevance feedback, in: S.-H. Myaeng, D. W. Oard, F. Sebastiani, T.-S. Chua, M.-K. Leong (Eds.), Proc. 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, Singapore, 2008, pp. 235–242.

[8] G. V. Cormack, M. R. Grossman, Evaluation of machine-learning protocols for technology-assisted review in electronic discovery, in: P. Bruza, C. L. A. Clarke, K. Järvelin (Eds.), Proc. 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Gold Coast, Australia, 2014, p. 153–162.

[9] B. Liu, Y. Dai, X. Li, W. Lee, P. Yu, Building text classifiers using positive and unlabeled examples, in: 3rd IEEE International Conference on Data Mining, Melbourne, FL, USA, 2003, pp. 179–186.

[10] B. J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972–976.

[11] E. Yang, D. D. Lewis, O. Frieder, Text retrieval priors for bayesian logistic regression, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 2019, p. 1045–1048.