
Crowdsourcing Intelligence Analysis with Context Slices

Tianyi Li

Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
tianyili@vt.edu

Asmita Shah

Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
asmita21@vt.edu

Kurt Luther

Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
kluther@vt.edu

Chris North

Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
north@vt.edu

Abstract

Making sense of large text datasets is a difficult problem in many domains and does not scale well for individuals. Crowdsourcing presents new opportunities for large-scale sensemaking, but we must first overcome the challenge of enabling many distributed novice workers to contribute meaningfully. In this paper, we explore the use of non-expert crowds to support expert analysts in complex sensemaking, focusing on the task of finding connections between entities in documents with narrative textual information. We introduce a novel concept called *context slices* in which datasets are restructured to support large-scale text analysis by crowd workers. We implemented this concept in a web application, *Connect the Dots*, in which crowds build subgraphs of entity relationships that can be layered and visualized for expert analysts. Our results suggest that with context slices, crowds are able to find most of the connections that analysts need, along with accurate and meaningful description labels, and can be used to retrieve and schematize information from the source documents.

Author Keywords

Crowdsourcing; Information Visualization; Intelligence Analysis; Sensemaking

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Group and organization interfaces

Introduction

In the field of intelligence analysis, analysts must deal with a tremendous amount of rapidly changing data with diverse or unknown structures to look for clues of potential threats. Crowdsourcing presents new opportunities for complex sensemaking by augmenting the cognitive work of individual analysts, providing richer analysis than automated approaches and scaling better than traditional intelligence work [5]. However, leveraging crowds requires finding a way for many distributed novice workers to contribute meaningfully, through small and independent tasks, to the sensemaking process of experts [20].

To address this challenge, we introduce a novel concept called *context slices*. With context slices, we use semantic information about a large text corpus to intelligently “slice” document subsets into analysis micro-tasks suitable for crowdsourcing. The slices provide enough context for the worker to make connections and draw broader conclusions without being overwhelming.

We implement context slices in a web application, *Connect the Dots*, that allows crowds to create and visualize networks of entity relationships aggregated across many workers and documents, providing a potentially valuable resource for an analyst.

To evaluate our approach, we conducted a study with 275 crowd workers and a text corpus describing a fictional terrorist plot. We experiment with different slicing methods and explore to what extent crowds can find the connections needed by expert analysts, when only examining the data through context slices. Our results show that the crowds connected most of the connections needed by experts and can be visualized to show useful patterns of the information. We also found that context slices composed of documents with overlapping entities lead to better analysis quality.

Related Work

Making sense of large amounts of textual data is a demanding cognitive task often performed by experts. In this section, we review prior research that studies and models the sensemaking process of individuals and groups of expert analysts.

Collaboration among Experts

Collaboration among small groups of expert analysts is a common practice in intelligence agencies. Beyond information processing, communication patterns, geographic distribution, and group dynamics introduce additional complexity [11]. Researchers have sought to model collaborative sensemaking processes, including task decomposition granularity and team sizes from pairs of analysts [2,17] to small teams [4, 22]. Goyal et al. [16] proposed and evaluated an interface for distributed sensemaking in real time, which improves task performance without increasing cognitive workload via implicit information sharing. Zhao et al. [27] developed a system that supports handoff in asynchronous collaborations through knowledge transfer graph, through an interactive tool with rich annotation features.

These systems built to assist individual experts and small groups inspire us to bring much larger crowds into the sensemaking loop. However, unlike most work in this area, we recruit paid crowd workers who are non-experts and contribute for short time periods. We explore how to restructure the dataset and delegate appropriate tasks to these novice transient crowd workers so that they can collaborate asynchronously and contribute meaningfully to the process of intelligence analysis.

Crowdsourced Text Analysis

Crowdsourcing, either alone or combined with automated approaches [16,25], has been used to forage and synthesize information with unknown topics

and diverse structures [19,20]. One of the challenges of crowdsourcing is that it often lacks global context for local tasks. Researchers have developed iterative task designs to address this issue for text analysis tasks, such as clustering and categorizing [1]. Cascade [7] produces crowdsourced taxonomies of hierarchical data sets by letting workers generate, and later select, multiple categories per item. Frenzy [6] is a web-based collaborative conference session organizer that elicits conference paper metadata by letting crowd workers group papers into sessions using an asynchronous clustering tool. Alloy [5] leverages machine learning with clustering tasks in global contexts. In the Knowledge Accelerator [17], crowd workers are able to find relevant information about a given topic and aggregate the findings together meaningfully. Using context trees [23], crowd workers can provide ratings on global importance of documents via local views. We draw inspiration from these projects, particularly the notion of integrating micro-tasks into a more collaborative, unstructured interface embodied in Frenzy and other forms of crowdware [26]. Unlike these projects, however, we focus on leveraging crowds to support analysts in finding hidden connections between entities in textual data, requiring contextual analysis of the contents and going beyond common-sense knowledge.

Named Entity Recognition and Crowdsourcing

Identifying and classifying the key entities (people, places, organizations, etc.) mentioned in text documents is a valuable early step to enable more complex information processing and sensemaking. Automated approaches to named entity recognition (NER) have made significant headway (e.g. [3,13,14]), but human intervention is often required to achieve acceptable results in specialized contexts like intelligence analysis. Detecting semantically similar

entities in textual descriptions can be complicated even for humans, and beyond the capability of machine-based approaches [18]. In the Linked Open Data (LOD) community, researchers have sought to bridge the gap between algorithmic matching and manual techniques by parsimoniously using human workers to guide the automated process of linking entities in natural language texts to existing structured concepts [12]. Other efforts seek to employ paid crowdsourcing as part of a human-in-the-loop workflow. For example, Wang et al. proposed hybrid human-machine approach called *CrowdER* [24], that uses machine-based techniques to make a first pass, and only ask crowds to verify more difficult pairs.

In this paper, we build on these earlier efforts and consider how crowdsourcing could support more complex entity recognition and identify more semantically distant entity relationships. We use existing techniques for a first pass at entity extraction, and then use context slices to allow crowds to find more subtle connections (e.g. the same terrorist suspect using two different aliases).

System Description

We designed and implemented a web application called *Connect the Dots* (Figure 1) to help crowd workers analyze documents and make connections between entities. There are two main features in the web application to facilitate each crowd worker's analysis process within a given context slice: 1) the Document View, and 2) the Connection Workspace.

Document View

The left side of the interface lists all documents in the given context slice (Figure 1.B). Named entities are automatically extracted from the documents and highlighted in different colors by categories: person, location, organization, money, phone number, date and

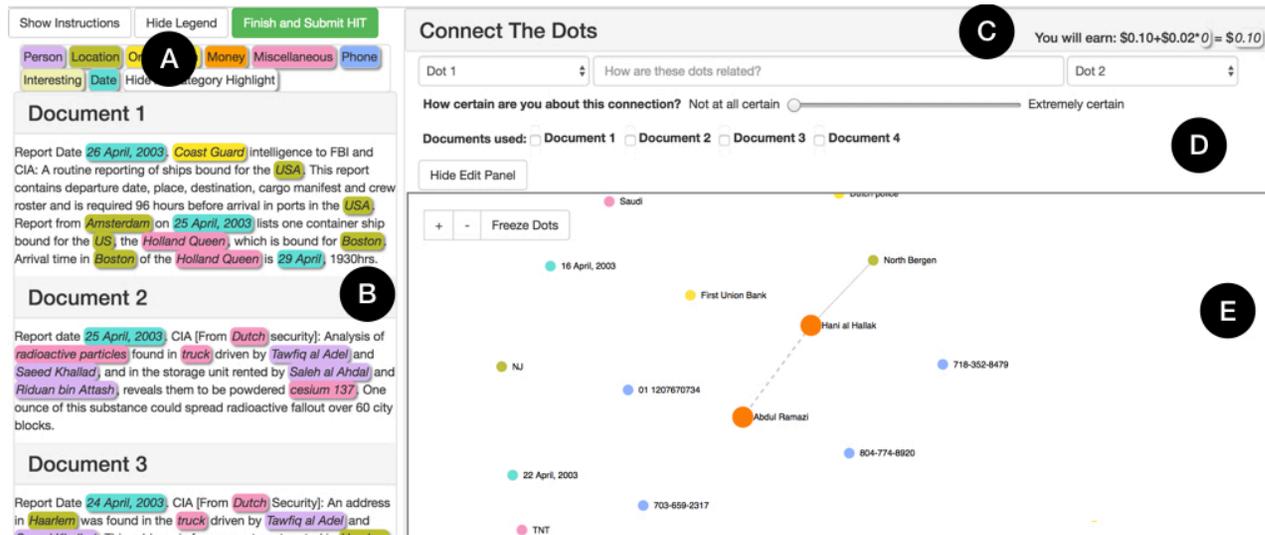


Figure 1: Connect the Dots web application interface.

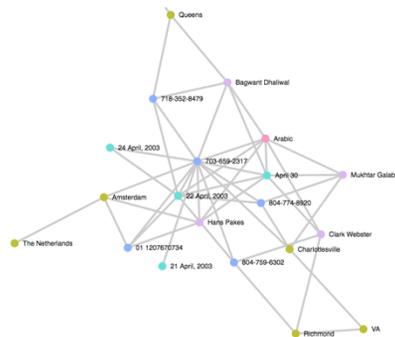


Figure 2: Example subgraph of connections made by five crowd workers for one context slice.

miscellaneous. A legend describes the category names and colors (Figure 1.A). Users can click a category name to show or hide all entities of that category in the document(s).

Connection Workspace

The Connection Workspace is composed of the visualization panel and the edit panel, both on the right side of the interface.

The visualization panel (Figure 1.E) displays the entities in documents as nodes ("dots"), colored based on their categories and labeled with the entity names. When the user selects two unconnected nodes, a dashed line appears to suggest a potential link. By default, the visualization uses a force-directed layout to minimize overlaps and intersections, but the user can click the "Freeze" / "Unfreeze" button to control the graph

movement and manipulate node positions via drag-and-drop.

The edit panel (Figure 1.D) is an input form where users can create and describe node connections. Four types of information are required for each connection: 1) the names of the two nodes to be connected, 2) a brief description of their relationship, 3) the user's certainty about the connection, and 4) checkboxes to indicate which documents provide evidence supporting the connection.

When the user selects two entities with no connection between them, a "Create Connection" button appears. If a connection already exists between the nodes, then "Update Connection" and "Delete Connection" buttons appear instead.

Users can select the nodes to connect in any of three ways: 1) choosing from alphabetized dropdown menus in the edit panel, 2) clicking on the entities in the

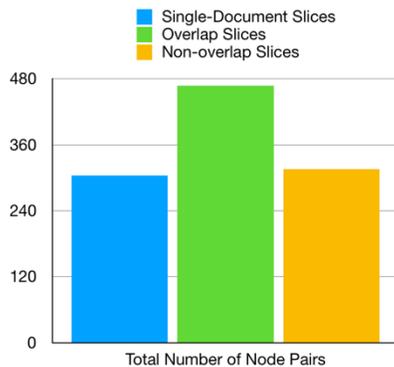


Figure 3: Comparison among slicing methods: Number of node pairs (identified by the two entities being connected).

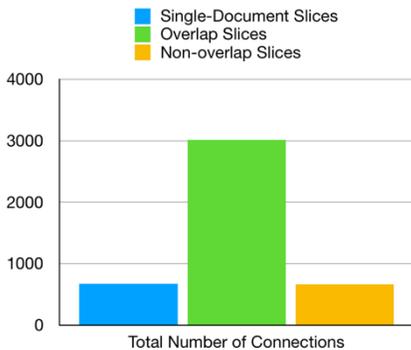


Figure 4: Comparison among slicing methods: Number of connections (identified by the two entities and their relationship descriptions by each crowd worker).

documents, or 3) clicking on the nodes in the visualization. Finally, the user’s number of connections made, and the corresponding payment earned, are updated on the upper right every time the user creates or deletes a connection (Figure 1.C). In the next section, we describe an experiment to evaluate the utility of the Connect the Dots system and the context slices approach.

Evaluation Study

Creating Context Slices

From our pilot studies, we found that a slice size of one or two documents usually takes 15 to 30 minutes for one crowd worker to finish, depending on the number of entities and other words in the documents. Therefore, we generated 55 possible context slices: 45 different combinations of double-document slices and 10 single-document slices. This covers three types of slicing methods:

1. single-document slices
2. double-document slices with overlapping entities
3. double-document slices without overlapping entities

Dataset

We use a subset of the *Sign of the Crescent* dataset [14] originally developed for training professional intelligence analysts. The complete dataset consists of 41 fictional text intelligence reports regarding three coordinated terrorist plots in three US cities. Each plot involves a group of at least four suspicious people. Each report, or *document*, contains a single prose paragraph ranging from 33 to 210 words. In this study, we focus on solving one of the three plots in this dataset. The relevant information for this plot is distributed across 10 of the documents.

Participants

We recruited crowd workers from Amazon Mechanical Turk (AMT), restricted to US-only workers with an acceptance rate greater than 90%. In total, we recruited 275 crowd workers and randomly assigned five workers to each context slice. Each worker was assigned to only one HIT (Human Intelligence Task), to mitigate learning effects or collusion.

Procedure

After accepting the HIT, each worker is randomly assigned to a context slice. If the worker accepts the IRB consent form, she will see a modal dialog box with HIT instructions. The instructions explain the background and documents (“a few pieces of evidence from a fake terrorist plot”), the task (“make connections based on the information”), how to use the interface, the minimum number of connections required, and the bonus policy, which rewarded extra connections. Once workers have connected enough pairs of entities, they can click the “Finish and Submit HIT” button and voluntarily provide feedback.

Data Collection and Analysis

For each worker who accepted the HIT, we collected their basic AMT credentials, the connections made by them, and their feedback, if any. For each connection, along with the entity pair and annotations (including relationship descriptions, evidence documents and level of certainty), we recorded the timestamp when a connection is made, the worker who created it, and the context slice from which it was created.

We conducted quantitative analysis to compare overall statistics of each slicing methods (Figure 3-6) and the precision/recall values against gold standard connections (Figure 7). We also qualitatively coded the types of crowd connections.

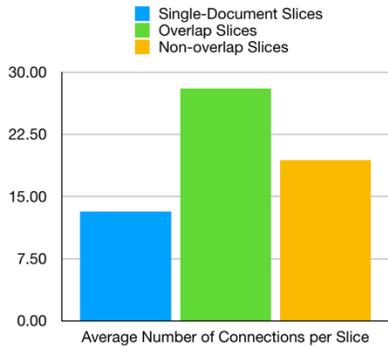


Figure 5: Comparison among slicing methods: Average number of connections created per slice.

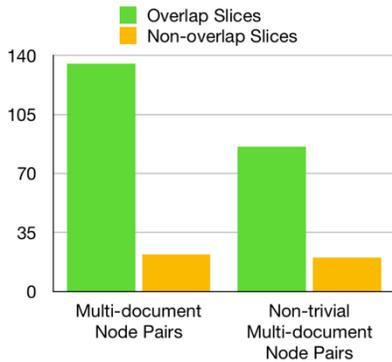


Figure 5: Multi-document node pairs in different slicing methods.

Results and Discussion

The 275 crowd workers participated in analyzing the 55 context slices generated by three different slicing methods. The average time spent on each HIT was 20 minutes (min=8.5, max=37, SD=6.3).

How Productive are Different Slicing Methods?

We measured productivity by the number of node pairs (identified by the two entities being connected) and connections between the same pair of entities (identified by different relationship descriptions) by different workers. In total, 622 pairs of entities and 5992 connections were connected by crowd workers. Specifically, for single-document slices (10 slices), workers created 671 connections (mean=13, SD=11.6) between 304 pairs of entities. For double-document slices with overlapping entities (26 slices), there were 3019 connections (mean=28, SD=16.75) between 467 pairs of entities. For double-document slices without overlapping entities, we examine a sample of 5 slices that cover all 10 documents. There were 666 connections (mean=19.4, SD=3.6) between 316 pairs of entities (Figures 3 and 4). As in double-document slices with overlapping entities, some of the documents were assigned to more than one group of crowd workers. We computed the average number of connections in each slice to normalize this difference. We can see in Figure 5 that overlapping documents led to more than double the number of connections, while non-overlapping slices led to less than double the number of connections, even as the number of documents was doubled. This indicates that increasing the amount of work without bringing in shared contexts will not increase and may even hinder productivity.

How Accurate Are Different Slicing Methods?

To measure how accurately the crowds can retrieve the connections needed for experts to uncover the plot, we computed the precision and recall values by comparing

entity pairs connected by crowd workers against a set of gold standard connections G (177) generated from the dataset’s solution sheet (Figure 8). Given a set of crowd-generated entity pairs C , the overlapped entity pairs $O = C \cap G$. The precision value is then computed as $P = \frac{|O|}{|C|}$ and the recall value is $R = \frac{|O|}{|G|}$.

For each slice, we used the number of workers that connected a certain pair of entities as a “majority vote” (1-5) threshold to decide whether to count this entity pair in the result or not. For example, if the threshold = 3, then we only considered entity pairs that were connected by 3 or more workers (out of the 5 working on this slice). We then aggregated the results from each slice to produce a set of crowd-generated entity pairs for each threshold. Let the set of connections of the i^{th} context slice with threshold t be C_i^t ; the set of combined connections given a threshold t is $C^t = \cup_i C_i^t$. Precision, recall, and f-measure (harmonic mean of precision and recall) for each slicing method using combined connections in each threshold are shown in Figure 7. Our algorithmic baseline generated by document co-occurrence gave a precision value of 0.17 and a recall value of 0.77.

The overall precision-recall values are similar between single-document slices and double-document slices without overlapping entities, reaching optimal f-measure at threshold = 4. Double-document slices with overlapping entities produce a maximum f-measure of 0.50 with threshold = 4 workers. This indicates that double-document slices with overlapping entities enabled the best overall accuracy.

With a threshold of less than 3 on majority vote, the f-measure of non-overlap slices is less than 0.4. This indicates that slices that contains overlapping contexts will lead to more stable quality from crowd-generated results. Double-document slices with overlapping

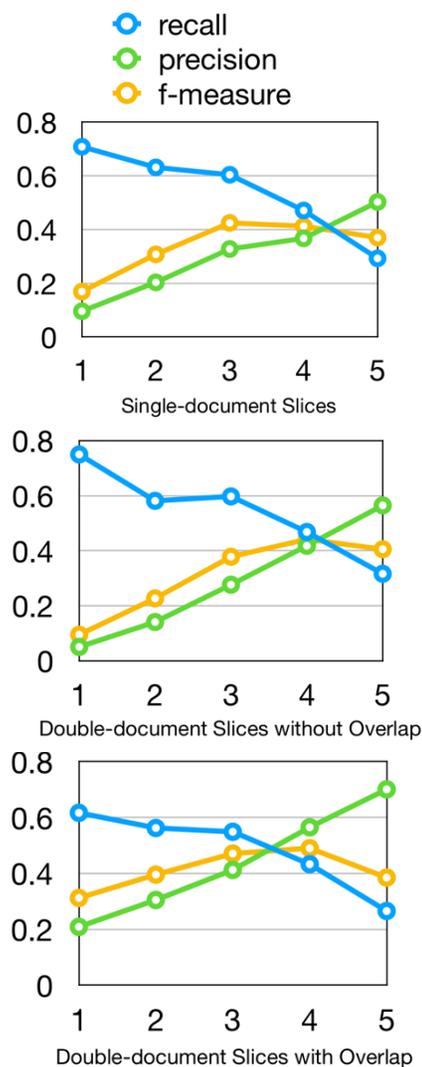


Figure 7: Precision, recall, and f-measure values for varying worker vote thresholds.

entities also require one less crowd worker (3 vs. 4) to achieve a better f-measure than other slicing methods. Even with a threshold of 5, single-document slices and double-document slices without overlapping entities only recall around half of the gold standard connections, while double-document slices with overlapping entities outperform the other two by 50% to achieve a recall value of 0.75. This is close to co-occurrence connections (0.77) but with 60% the number of node pairs (467 vs. 790) being connected. By comparing different slicing methods, we found that double-document slices with overlapping entities provide shared contexts between documents and outperform other slicing methods. Single document slices are efficient in terms of collecting contextual information pieces but lack the ability to generate insightful connections between documents. Using more than one document without overlapping entities will hinder the quality of work and is not recommended.

What Types of Connections Do Crowds Create?

To understand the types of connections workers made, we randomly sampled 727 of the 5992 crowd-generated connections to inspect in detail. Inspired by typologies of machine-recognized entities [14], we identified three types of connections:

T1. Contextual connections. This type of connection describes a semantic relationship given only in the documents. For example, a person, *Hans Pakes*, has the phone number *703-659-2317*. These connections are the most important to solving the hidden plot.

T2. Common-sense connections. This type of connection represents common sense or external knowledge related to the entities being connected. For example, *Queens* is a borough in *NYC*. These connections occur because realistic documents present entity information in inconsistent ways, and it is

challenging for named entity recognition algorithms to choose the perfect granularity for a given purpose.

T3. Collateral connections. This type of connections represents “metadata” about the entities that do not contribute to sensemaking, and could be generated as well or better by algorithms. For example, *April 30, 2003* and *April 25, 2003* are both dates.

Our analysis found that 586 crowd connections represented meaningful facts (327 T1 and 259 T2 connections) from the given context, even if some of them didn’t match the gold standard connections. Crowd workers made reasonable speculations with the given information and applied their domain knowledge relevant to the context slice. For example, a worker recognized a surname to be Arabic and connected it to a Middle Eastern country based on this domain knowledge. Some workers also used connection descriptions to suggest causation and pose hypotheses. For example, a worker connected “21-Apr-03” and “\$35,000”, describing their relationship as “After receiving this money more suspicious activity started on this day.” Although this description did not strictly align with our task instructions, it illustrates the crowd’s capability and willingness to provide more advanced and subjective insights.

Where did Crowds Miss Connections?

We examined the gold standard connections that the crowd were not able to create. In the 40 gold standard connections not created by any crowd worker, 23 are T1 connections that require more than two documents to connect. The remaining 17 connections are synonymous with connections that were created by crowd workers. For example, some gold standard connections use the surname entity *al Hallak*, but the crowds used the full name *Hani al Hallk* entity to connect to the same nodes.

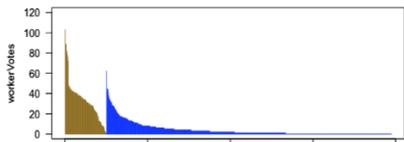


Figure 8: Ranked entity pairs by the number of workers connecting them. Gold lines are entity pairs from the gold standard. Blue lines are other possible entity pairs

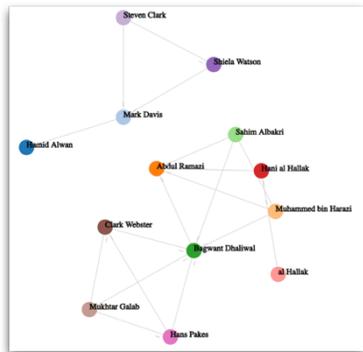


Figure 9: Relationship among person names by co-occurrence baseline.

How to Eliminate Useless Connections?

In the sample of 727 connections, we found 141 connections that were not meaningful (T3 connections). From a task design perspective, these connections could be discouraged by providing more specific instructions or style guides for workers, and by enhancing the interface to detect frequent mistakes. Several classes of such mistakes are already apparent from our evaluation. For example, a worker connected two person names and described their relationship as “[these are] both names” or “[these] appeared in the same report”. Another worker labeled a connection as “date they called this city”. The system could ask crowd workers to avoid using pronouns or repeating given entity names in their relationship descriptions, or automatically detect if the name of entity categories (e.g. “name”, “location”) appear in relationship descriptions and alert workers about possible mistakes. From a post-processing perspective, we suggest that by algorithmically generating T2 connections with external knowledge base, and T3 connections by document metadata, most of the T2 and T3 connections can be marked or removed by taking the difference from the crowd results.

How to Focus on Useful Connections?

The existence of T2 connections resulted in a lot of noise in crowd generated connections, even though the crowd outperforms the co-occurrence baseline. To address this, we applied a common strategy [8] for schematizing information in intelligence analysis: investigating and aggregating relationships between person names. We visualized both the crowd-generated and gold standard connections for person names, to evaluate the quality of crowd-generated connections. We found that crowd workers successfully connected and correctly described all pairs of person names whose relationship can be discovered using two documents.

We built visualization of person name networks from document co-occurrence (Figure 9), crowd-generated connections (Figure 10), and gold-standard connections (Figure 11). All of the circled person names in gold standard graph are connected to more than two other person names in the crowd-generated graph, whereas no similar patterns were found in the baseline co-occurrence graph. This indicates that the crowd-generated graph of person names accurately identify top suspects and get experts started on further investigation. It is also possible to learn the relationship between people by reading the most frequent crowd-generated labels for that connection. For example, the connection “Bagwant Dhaliwal---Sahim Albakri” is most frequently described with the words: ‘indian’, ‘alias’, ‘used’, ‘name’, ‘passport’. It can be inferred that these two names are used by the same person and it is even (correctly) suggests the fake name is used in an Indian passport. With a quick review of original descriptions written by crowd workers, expert analysts can easily retrieve relationship information about these two names.

How to Utilize Relationship Descriptions?

Based on observations above, we explored using a clustering algorithm to computationally aggregate useful connection descriptions. We ran K-Means algorithm based on tf-idf similarity between edge labels to cluster them. A quick ranking of description labels for each pair of entities reveals that there are many identical descriptions written by different crowd workers. In addition, non-identical descriptions are often very similar, with many repeated key words (e.g. “city in state” vs. “city is in this state”). Considering that common stop words are useful to convey information in our case (“is in”, “are from”, etc.), we only used four stop words: “the”, “a”, “this”, “that”. For each description, we first removed the words in the two

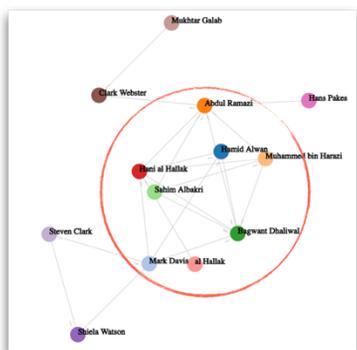


Figure 10: Crowd generated graph of person names.

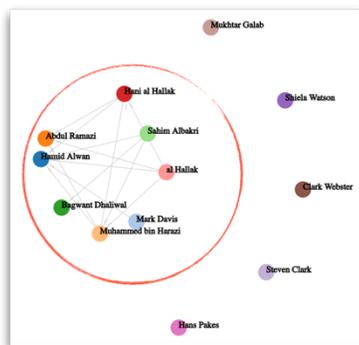


Figure 11: Gold standard graph of person names.

entities it connects then the four stop words. We tokenized and stemmed the remaining words in the description (if any) before computing tf-idf similarity. We tested cluster numbers of 3 and 10 for a K-Means algorithm to understand the number of relationship categories the crowd generated for each node pair. Both numbers yield highly similar top words in each cluster. After removing the duplicate top words, the overall centroid words in all clusters were less than 10. In almost all cases, the combined top words provided valuable semantic information to convey the relationship between the entity pair. For example, the connection between two person names *Hamid Alwan* and *Mark Davis* has the top centroid words: ['person', 'as', 'name', 'same', 'identified']. Example of crowd-generated connection descriptions are “name used by” and “same person”. Thus, despite the minor differences in description labels, the keywords used to portray the relationship between connected entities are usually similar and can be aggregated using representative centroid key words. This preserves the semantic meaning of the description and can be understood without reading either the crowd’s description labels or the original documents (those two names in the example refer to the same person identity).

Limitations and Future Work

We analyzed the quality of crowd connections by comparing them to gold standard connections provided by the creators of the *Crescent* dataset. We caution that the gold standard connections alone are not sufficient to evaluate crowd worker’s results. The solution given in the dataset is written with a global context and include high-level hypotheses that cannot be generated with only two local documents. These analyses revealed similarities between crowd and expert performance and other indications of value, but further research is needed to explore the impact of

crowd connections on an expert analyst’s sensemaking process [10]. Additionally, we only used a subset of one dataset for our experiment; follow-up studies are needed to understand how larger datasets or other types of documents affect crowd performance.

Conclusion

In this work, we explored non-expert crowds’ potential to support a complex sensemaking process of expert analysts. Our results indicate that crowdsourcing offers a promising opportunity in collaborative sensemaking by bringing the crowds into the sensemaking loop. We found that with the concept of “context slice”, the crowds can work in parallel and independently on easy and small tasks to find 93.6% of the entities pairs connected by experts. With a reasonable threshold (3 votes out of all 5 crowd workers, if using two-document with overlap as example slicing method), we can achieve 41% precision (comparing to 17% baseline) and 55% recall (comparing to 77% baseline). Crowd-generated connections can be strategically retrieved and schematized by experts to provide deep insights. Last but not least, applying clustering algorithm to crowd-generated description labels can generate meaningful keywords that appropriately describe the relationship between entities and help experts quickly understand the information as a precursor to reading the original documents. This indicates the potential for providing condensed contexts and suggesting an effective starting point for more efficient investigation by experts.

Acknowledgements

This research is funded by NSF 1527453 and 1651969. We wish to thank Maoyuan Sun, Yali Bian, Edward McEnrue, Jazmine Zurita and Chris Lai for their help.

References

1. Paul André, Aniket Kittur, and Steven P. Dow. 2014. Crowd Synthesis: Extracting Categories and Clusters from Complex Data. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*, 989–998. <https://doi.org/10.1145/2531602.2531653>
2. Richard Arias-Hernandez, Linda T. Kaastra, and Brian Fisher. 2011. Joint action theory and pair analytics: In-vivo studies of cognition and social interaction in collaborative visual analytics. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 3244–3249. Retrieved September 21, 2016 from <http://www.academia.edu/download/30933920/paper0747.pdf>
3. Mikhail Bilenko and Raymond J. Mooney. 2003. Adaptive Duplicate Detection Using Learnable String Similarity Measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, 39–48. <https://doi.org/10.1145/956750.956759>
4. Lauren Bradel, Alex Endert, Kristen Koch, Christopher Andrews, and Chris North. 2013. Large high resolution displays for co-located collaborative sensemaking: Display usage and territoriality. *International Journal of Human-Computer Studies* 71, 11: 1078–1088. <https://doi.org/10.1016/j.ijhcs.2013.07.004>
5. Joseph Chee Chang, Aniket Kittur, and Nathan Hahn. 2016. Alloy: Clustering with Crowds and Computation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*, 3180–3191. <https://doi.org/10.1145/2858036.2858411>
6. Lydia B. Chilton, Juho Kim, Paul André, Felicia Cordeiro, James A. Landay, Daniel S. Weld, Steven P. Dow, Robert C. Miller, and Haoqi Zhang. 2014. Frenzy: Collaborative Data Organization for Creating Conference Sessions. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*, 1255–1264. <https://doi.org/10.1145/2556288.2557375>
7. Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. 2013. Cascade: Crowdsourcing Taxonomy Creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, 1999–2008. <https://doi.org/10.1145/2470654.2466265>
8. George Chin Jr., Olga A. Kuchar, and Katherine E. Wolf. 2009. Exploring the Analytical Processes of Intelligence Analysts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, 11–20. <https://doi.org/10.1145/1518701.1518704>
9. Haeyong Chung, Sharon Lynn Chu, and Chris North. 2013. A Comparison of Two Display Models for Collaborative Sensemaking. In *Proceedings of the 2Nd ACM International Symposium on Pervasive Displays (PerDis '13)*, 37–42. <https://doi.org/10.1145/2491568.2491577>
10. Haeyong Chung, Sai Prashanth Dasari, Santhosh Nandhakumar, and Christopher Andrews. CRICTO: Supporting Sensemaking through Crowdsourced Information Schematization.
11. Gregorio Convertino, Dorrit Billman, Peter Pirolli, J. P. Massar, and Jeff Shrager. 2008. The CACHE Study: Group Effects in Computer-supported Collaborative Analysis. *Computer Supported Cooperative Work (CSCW)* 17, 4: 353–393. <https://doi.org/10.1007/s10606-008-9080-9>

12. Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-scale Entity Linking. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*, 469–478. <https://doi.org/10.1145/2187836.2187900>
13. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. 2007. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 19, 1: 1–16. <https://doi.org/10.1109/TKDE.2007.250581>
14. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*, 363–370. <https://doi.org/10.3115/1219840.1219885>
15. Ryan Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. 2011. Crowdclustering. In *NIPS 2011*. Retrieved September 21, 2016 from <http://papers.nips.cc/paper/4187-crowdclustering>
16. Nitesh Goyal and Susan R. Fussell. 2016. Effects of Sensemaking Translucence on Distributed Collaborative Analysis. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*, 288–302. <https://doi.org/10.1145/2818048.2820071>
17. Nathan Hahn, Joseph Chang, Ji Eun Kim, and Aniket Kittur. 2016. The Knowledge Accelerator: Big Picture Thinking in Small Pieces. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*, 2258–2270. <https://doi.org/10.1145/2858036.2858364>
18. Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of Entity Resolution Approaches on Real-world Match Problems. *Proc. VLDB Endow.* 3, 1–2: 484–493. <https://doi.org/10.14778/1920841.1920904>
19. Narges Mahyar and Melanie Tory. 2014. Supporting Communication and Coordination in Collaborative Sensemaking. *IEEE transactions on visualization and computer graphics* 20, 12: 1633–1642. <https://doi.org/10.1109/TVCG.2014.2346573>
20. Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, 2–4. Retrieved September 21, 2016 from https://www.e-education.psu.edu/geog885/sites/www.e-education.psu.edu/geog885/files/geog885q/file/Lesson_02/Sense_Making_206_Camera_Ready_Paper.pdf
21. Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. 2011. Adaptively Learning the Crowd Kernel. *arXiv:1105.1033 [cs]*. Retrieved September 21, 2016 from <http://arxiv.org/abs/1105.1033>
22. Anthony Tang, Melanie Tory, Barry Po, Petra Neumann, and Sheelagh Carpendale. 2006. Collaborative Coupling over Tabletop Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, 1181–1190. <https://doi.org/10.1145/1124772.1124950>
23. Vasilis Verroios and Michael S. Bernstein. 2014. Context Trees: Crowdsourcing Global Understanding from Local Views. In *Second AAAI Conference on Human Computation and Crowdsourcing*. Retrieved September 30, 2017

from
<https://www.aaai.org/ocs/index.php/HCOMP/HCOMP14/paper/view/8951>

24. Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. 2012. CrowdER: Crowdsourcing Entity Resolution. *Proc. VLDB Endow.* 5, 11: 1483–1494. <https://doi.org/10.14778/2350229.2350263>
25. Jinfeng Yi, Rong Jin, Anil Jain, and Shaili Jain. 2012. AAAI Workshop - Technical Report. Retrieved September 21, 2016 from <https://scholars.opb.msu.edu/en/publications/crowdclustering-with-sparse-pairwise-labels-a-matrix-completion-a-3>
26. Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. 2012. Human Computation Tasks with Global Constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, 217–226. <https://doi.org/10.1145/2207676.2207708>
27. J. Zhao, M. Glueck, P. Isenberg, F. Chevalier, and A. Khan. 2017. Supporting Handoff in Asynchronous Collaborative Sensemaking Using Knowledge-Transfer Graphs. *IEEE Transactions on Visualization and Computer Graphics* PP, 99: 1–1. <https://doi.org/10.1109/TVCG.2017.2745279>