What do Journalists do with Documents? Field Notes for Natural Language Processing Researchers

Jonathan Stray
Columbia Journalism School
jms2361@columbia.edu

ABSTRACT
Natural language processing and visualization systems have been proposed to help journalists analyze large sets of documents, but very little has been said on what journalists do with documents in practice. We review a collection of 15 stories completed with the Overview document mining platform, characterizing the source material and reporting tasks. The median document set contained 4,000 documents and the majority arrived as paper or scanned paper. In most cases journalists knew what they were looking for in advance, in contrast to the large research literature concerned with "exploring" a document set. We also review five cases where custom NLP techniques were used to produce a story, including applications of topic modeling, entity recognition, text classification, and sentiment analysis. Based on the cases in these two collections, we recommend six practice-driven themes for natural language processing researchers who want to assist journalists with large document sets: 1) Robust import. 2) Robust analysis. 3) Search, not exploration. 4) Quantitative summaries. 5) Interactive methods. 6) Clarity and Accuracy.

Keywords
Computational journalism; investigative journalism; natural language processing; document mining

1. INTRODUCTION
Journalists frequently work with large sets of documents, whether obtained through open data, leaks, or Freedom of Information Act (FOIA) requests, and increasingly use computers and algorithmic techniques in this work. Many people have sensed the potential of applying modern natural language processing and visualization techniques to the field, e.g Cohen et al.:

Stories will emerge from stacks of financial disclosure forms, court records, legislative hearings, officials' calendars or meeting notes, and regulators' email messages that no one today has time or money to mine. With a suite of reporting tools, a journalist will be able to scan, transcribe, analyze, and visualize the patterns in these documents [1]

But this inspiring vision is not a design for a journalistic document mining system. Stories don't just "emerge." They result from a journalist going through a specific set of actions on a particular document set.

This article focusses on the gap between the techniques that natural language processing (NLP), visualization, and machine learning researchers have proposed to help journalists analyze document sets, and how those techniques have fared in practice. It draws on the work completed with the Overview document mining system for investigative journalists [2], the author's own reporting work, and an analysis of notable document mining projects in journalism. As we discuss these cases, we note where they touch on six proposed research themes for computer scientists who wish to help journalists.

2. PREVIOUS WORK
A large strand of research ([3] [4] [5] [6] [7]) analyzes news articles using NLP techniques such as clustering or sentiment analysis, that is, it analyzes journalists' *output*. We are concerned instead with producing articles through analysis of source materials such as public records. This has been contemplated by many computer scientists, but most such work discusses journalism as a potential application area without ever consulting or testing with journalists ([8] [9] [10] [11]).

Conversely, journalists and communication scholars have recognized the importance of NLP techniques for reporting but have not specified how they would be applied in practice [1] [12] [13]. Only a small number of previous studies involve testing a real system with real journalists, such as Diakopoulos's work on helping reporters find relevant tweets [14] [15] and the author's work on the Overview visual document mining system [2].

Twitter sentiment analysis has been widely used in journalism as a proxy for public opinion around events such as political debates [16] [9]. This article discusses sentiment analysis as part of a larger reporting process, and not necessarily applied to social media data.

Entity detection and document classification have been used in journalism for automated tagging, recommendation, and other information management tasks [17]. Again, here we focus on the use of these techniques at the story production stage.

It is widely acknowledged that import and cleanup – also called ETL or data "wrangling" -- is most of the work of data projects [18] and these problems are well known to data journalists [19]. Our cases illustrate the (generally terrible) quality of journalists' source documents and suggest useful problems to solve.

While scholarship has been sparse, practical tools have made NLP techniques more widely available to journalists. DocumentCloud [20] has made it easy for journalists to ingest, search, publish documents in a variety of formats and supports some analysis tools such as entity detection (via Reuters' OpenCalais API [21]) and timelines. Most recently Aleph [22] integrates entity-based search of over 100 data sets relevant to journalism.

3. STORIES DONE WITH OVERVIEW
Overview is an open source platform designed specifically for investigative journalism on large sets of documents. It clusters documents based on text similarity and displays the resulting tree as an interactive visualization. It also does multi-lingual entity

detection, visualizes the network of co-occurring keywords, and draws word clouds. While these are powerful analysis tools, far more engineering effort has gone into workflow: Overview ingests and exports many document formats, supports Boolean and fuzzy searches, and provides a sophisticated tagging system.

Table 1 reports 15 stories completed using the Overview document mining platform, from the list maintained at [23]. The main limitation of this sample is that we only know of completed stories when our users contact us; over a thousand people have uploaded a document set to Overview, but either were not journalists, did not complete a story, or have not informed us. Crucially, for this set of stories we have insight into source documents and investigative process from correspondence with the reporters involved and/or their methodology posts. Four completed stories were omitted because we could not get details in this way.

Different stories used different units of analysis such as "file" or "page" but Overview abstracts these differences in the import process and deals simply with "documents." By this measure, the median document set size is 4000 documents. 5 out of 15 (33%) arrived on paper, and 10 (66%) arrived either as paper or scans of paper.

In 9 cases (60%) the journalist knew what they wanted to search for before the documents arrived, while in 6 cases (40%) the reporter engaged in more open-ended exploration, or wanted to visualize topical themes. In 4 cases (27%) the documents were emails, all of which arrived as either paper or scanned paper (as opposed to email archive formats like PST or mbox.)

While NLP researchers usually assume that a "document" is a string of characters, real document sets arrive in every conceivable format including paper, a large PDF containing thousands of pages of scans, or a deep directory structure with thousands of files. OCR of low quality scans produces garbled text which confounds tokenization and search algorithms *(robust analysis)*.

Reconstituting the original documents from the source material is a major unsolved problem. Email metadata and thread reconstruction from scans is a particularly common problem with no good solution. There is previous work addressing the general problem of document separation from page streams [24] [25] but such techniques have yet to be applied in journalism *(robust import)*. Overview allows users to split a file into pages as a simple workaround.

These results also suggest that journalists typically know what they are looking for when they begin analysis of a document set. In the author's experience pre-conceived search tasks are far more common than open-ended exploration tasks; this sample likely over-weights exploration tasks because Overview has been designed and promoted as an exploration tool. Many previous corpus visualization tools have also aimed to help the user "explore" a document set ([26] [27] [28] [29]) yet this is not what journalists usually do. Usually there is a reason the journalist went through the trouble of obtaining the documents in the first place, meaning that they have some idea what they're looking for. Unanticipated leaks are a notable exception, such as the WikiLeaks and Snowden materials, but these are comparatively rare in practice *(search not exploration)*.

Note that a "search" task does not mean standard text search is the best tool. For the Police Misconduct investigation, the reporter had to prove that several years worth of legislation did *not* discuss increased police accountability. He used Overview's clustering visualization to speed up his search by grouping closely related documents such as multiple drafts of the same bill [30]. Conversely, an "exploration" task does not require a visualization. The WikiLeaks releases were initially analyzed using nothing more than text search on a long list of interesting terms suggested by reporters [31].

Even when thematic analysis is the explicit goal, as in two of the stories in this sample, it is not obvious that NLP techniques such as clustering and topic modeling are the right tool. Overview includes hierarchical document clustering but "how does the algorithm decide which documents belong to which topics?" is a very frequently asked question with no straightforward answer. Reporters are accountable for the integrity and clarity of their results, which brings issues of interpretation and trust to the fore [32]. Eventually we added a simple editable word cloud to Overview, which is instantly interpretable (*clarity and accuracy)*.

4. STORIES USING CUSTOM NLP CODE

We now discuss five document-driven stories where journalists successfully applied NLP to complete a story. Notably, this is every case known to the author. These stories relied on topic modeling, sentiment analysis, text classification, and entity recognition.

4.1 Topic Modeling

For their 2014 story "The Echo Chamber," reporters Joan Biskupic, Janet Roberts and John Shiffman of Reuters wanted to show how a small group of elite lawyers have argued most of the cases before the U.S. Supreme Court [33] [34]. They assembled the 10,300 petitions to the court filed by over 17,000 lawyers from 2004 to 2012. Of these, only 528 cases were heard.

The reporters wanted to break down the number of accepted cases by type, for example whether filed by a business, individual, or government agency. They initially hired 20 freelancers who read every document over a period of three months and coded these categories by hand, but later decided to try topic modeling in the hopes of getting more detailed topic information. They applied LDA using the Gensim library and after some experimentation found that 40 topics seemed to capture the structure of the petitions most clearly. They examined the generated topics and manually labeled them with categories such as environmental regulation, congressional intent, utilities, etc. *(interactive techniques)*

But doubts remained about the reliability of this model. Roberts chose a sample of 1000 documents and read through them over two weeks. Interpreting a non-zero document-topic score to mean that the document concerned that topic, she discovered that the algorithm was only 36% accurate. But the highest numeric topic score for each document was accurate 93% of the time. Eventually, she determined thresholds for each topic which resulted in an overall accuracy of over 90%. This highlights the difficulty in interpreting the output from topic modeling technique *(clarity and accuracy)*.

Topic modeling still failed to capture variables that were key to the story: Was the case filed by a business or and individual, and was it a criminal or civil case? The final story combined manual coding with the algorithmic output. Topic assignments identified the cases that concerned business interests while the human coding identified which side the lawyer was working for, revealing that elite layers disproportionately represented businesses rather than individuals *(quantitative summaries)*.

Table 1. Stories completed with the Overview platform

Name	Organization	Year	Size	Unit	Type	Predefined search?	Task	Paper docs?
PG&E Regulators	KQED	2015	123,000	files	emails	yes	search / count threads	no
Military denies justice	Fusion	2014	112,000	files	discharge appeals	yes	count by regex match	no
Ryan federal funding	Associated Press	2012	9,000	pages	FOIA documents	yes	search	yes
Tulsa PD	Tulsa World	2012	8,680	pages	FOIA documents	yes	search	yes
St. Lukes hospital	Assn Health Care Journalists	2016	8,000	pages	court docs	no	explore	no
Texas Explosion	Dallas Morning News	2013	4,653	pages	emails	yes	search	no
Food Stamps	WRAL	2013	4,500	pages	emails	no	explore	yes
McCain Condors	Sunlight Foundation	2014	4,000	pages	FOIA documents	no	explore	no
Bridge Collapse	Seattle Times	2014	2,330	pages	NTSB report	no	explore	no
Athlete crimes	ESPN	2015	2,000	files	police reports	yes	count incidents by athlete	yes
Police Misconduct	Newsday	2013	1,900	records	proposed bills	yes	search	no
Credit Card repos	CreditCards.com	2014	1,600	files	card agreements	yes	search	no
Gun Debate	The Daily Beast	2012	1,300	comments	reader submitted	No	explore themes	No
Hawkins recall	Denton RC	2016	450	pages	emails	no	count by month	yes
Louis CK Emails	The Atlantic	2014	16	emails	promotional emails	no	explore themes	no

4.2 Sentiment Analysis

For the Washington Post story "Whistleblowers say USAID's IG removed critical details from public reports," Scott Higham and Stephen Rich compared drafts of 12 reports with their final versions. Using sentiment analysis, they found that more than 400 negative references were removed before publication [35] [34] *(quantitative summaries)*.

Rich found that existing algorithms misclassified the audit documents. For example the word "recommendation" is usually positive, but in this context it only occurs when there is a problem to be fixed. Such problems are not surprising, given that sentiment analysis accuracy is typically domain dependent [36]. Even with the work required to train the algorithm in this domain of audits *(interactive techniques)*, Rich believes this automated approach was "absolutely worth it … but most of what I do isn't going to require this." [34]

4.3 Text classification

For the story "License to Betray" Carrie Teegardin, Danny Robbins, Jeff Ernsthausen and Ariel Hart of the Atlanta Journal-Constitution scraped over 100,000 doctor disciplinary records from every state, looking for instances where doctors who had sexually abused patients were allowed to continue practice [37].

Ernsthausen drastically reduced this pile by applying machine learning to identify reports that were likely to concern sexual abuse. First the reporters manually labelled a few hundred documents to produce a training set. After trying several different classifiers including naïve Bayes on all TF-IDF features, he settled on logistic regression over a hand-selected set of relevant terms. This included both positive terms such as "sexual" and negative terms that suggested the incident concerned something else such as "narcotic." The final classifier had an area under ROC of >0.9 [personal communication].

Selecting only those documents with a rated probability of 0.5 or greater of concerning sexual abuse produced a set of 6,000 documents which the reporters then read and coded manually *(interactive techniques, clarity and accuracy)*. In this way they were able to identify substantially all cases within the larger set. The final story included details from notable cases as well as overall totals *(quantitative summaries)*.

The Los Angeles Times story "LAPD underreported serious assaults, skewing crime stats for 8 years" by Ben Poston, Joel Rubin and Anthony Pesce [38] was based comparing the narrative descriptions in more than 400,000 incident reports with the crime category assigned by police, e.g. "aggravated assault." The Cleary, this was too far many reports for a small team to read. However, the reporters had manually reviewed one year's worth of data for a previous story, providing a natural training set of over 20,000 incidents.

The narrative reports were written in a compressed shorthand, e.g. "VICTS AND SUSPS BECAME INV IN VERBA ARGUMENT SUSP THEN BEGAN HITTING VICTS IN THE FACE" *(robust analysis)*. After tokenizing and stemming, the reporters used a combination of two scikit-learn classifiers, SVM and Maximum Entropy, on bigram features. To assess the accuracy of the results they reviewed a random sample of 2,400 machine-labelled incidents *(clarity and accuracy)*. They discovered that the error rate was a hefty 24 ± 2%. Rather than attempting to improve the classifier, they simply adjusted their estimated yearly totals of misclassified crimes to account for this error [39] [40] (*quantitative summaries)*.

4.4 Entity Recognition

Although several production systems for journalists support named entity recognition (NER) including DocumentCloud, Overview, and Aleph, there is the question of where it is actually useful. If a reporter only wants to determine if any document refers to a particular entity, then text search suffices. In fact search may be better because existing NER systems often have very low recall. In the author's tests of OpenCalais vs. hand-annotated articles, the recall varies between 30-80%. Text search may also miss an entity due to name variations, typos, and OCR errors (*robust analysis)*, but at least the reporter has a clear idea of which variations will be detected and which will be missed *(clarity and accuracy)*.

The case for NER is stronger when reporters actually need a list of every entity in a document or corpus. Such was the case when Jennifer Golan and Shane Shifflet reported the story "Federal judge's rulings favored companies in which he owned stock" for the Center for Investigative Reporting [41].

Shifflet exhaustively transcribed California federal judges' "statement of economic interest" disclosures to generate lists of companies in which they owned stock. He then scraped the PACER database for every case those judges presided over (*robust import)* and used NER to generate a per-judge list of the entities involved [42][personal communication]. By comparing these lists the reporters were able to find cases in which judges had ruled favorably for companies in which they owned stock.

5. CONCLUSIONS

Our analysis of the stories completed with Overview sheds some light on the typical document set size (4000 documents), character (extremely dirty, often scanned), and reporting task (searching for something more-or-less well defined.) But the stories completed with custom techniques show that more sophisticated NLP techniques can also play a crucial role.

Throughout this paper we've highlighted aspects of the reporting work that touch on six larger themes. Synthesizing the problems encountered by journalists, we propose the following research directions for applying NLP to journalism.

Robust import. Preparing documents for analysis is a much bigger problem than is generally appreciated. Even structured data like email is often delivered on paper.

Robust analysis. Journalists routinely deal with unbelievably dirty documents. OCR error confounds classic algorithms. Shorthand and jargon break dictionaries and parsers.

Search, not exploration. Reporters are usually looking for something, but it may not be something that is easy to express in a keyword search. The ultimate example is "corruption."

Quantitative summaries. Journalists have long produced stories by counting the number of documents of a certain type. How can we make this easy, flexible, and accurate?

Interactive methods. Even with NLP, document-based reporting requires extensive human reading. How do we best integrate machine and human intelligence in an interactive loop?

Clarity and Accuracy. Journalists are accountable to the public for their results. They must be able to explain how they got their answer, and how they know the answer is right.

The cases presented here demonstrate that NLP-assisted reporting has broad potential, but only if researchers work on the problems that journalists truly need solved.

6. REFERENCES

[1] S. Cohen, J. T. Hamilton and F. Turner, "Computational Journalism," *Communications of the ACM,* vol. 54, no. 10, pp. 66-71, 2-11.

[2] M. Brehmer, S. Ingram, J. Stray and T. Munzner, "Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists," *IEEE Trans. Vis. Comput. Graphics,* vol. 20, no. 12, pp. 2271-2280, 2014.

[3] F. Moerchen, K. Brinker and C. Neubauer, "Any-time clustering of high frequency news streams," in *Proc. Data Mining Case Studies Workshop*, 2007.

[4] P. DiMaggio, M. Nag and D. Blei, "Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of US government arts funding," *Poetics,* vol. 41, no. 6, pp. 570-606, 2013.

[5] C. Suen, S. Huang, C. Eksombatchai, R. Sosic and J. Leskovec, "Nifty: a system for large scale information flow tracking and clustering," in *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[6] A. Balahur, R. Steinberger, M. Kabadjov, V. Zavarella, E. v. d. Goot, M. Halkia, B. Pouliquen and J. Belyaeva, "Sentiment Analysis in the News," in *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010.

[7] D. Newman, Y. Noh, E. Talley, S. Karimi and T. Baldwin, "Evaluating topic models for digital libraries," in *Proceedings of the 10th annual joint conference on Digital libraries*, 2010.

[8] T. Rusch, P. Hofmarcher, R. Hatzinger and K. Hornik, "Model trees with topic model preprocessing: An approach for data journalism illustrated with the wikileaks afghanistan war logs," *The Annals of Applied Statistics,* vol. 7, no. 2, pp. 613-639, 2013.

[9] N. A. Diakopoulos and D. A. Shamma, "Characterizing debate performance via aggregated twitter sentiment," in *SIGCHI*, 2010.

[10] C. Görg, Z. Liu, J. Kihm, J. Choo, H. Park and J. Stasko, "Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw," *IEEE Trans. Vis. Comput. Graphics,* vol. 19, no. 10,

pp. 1646-1663., 2013.

[11] P. Khare, P. Torres and B. R. Heravi, "What just happened? A Framework for Social Event Detection and Contextualisation," in *System Sciences (HICSS), 48th Hawaii International Conference on*, 2015.

[12] S. Cohen, C. Li, J. Yang and C. Yu, "Computational Journalism: A Call to Arms to Database Researchers," in *CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, 2011.

[13] C. W. Anderson, "Towards a sociology of computational and algorithmic journalism," *New media & Society,* vol. 15, no. 7, pp. 1005-1021, 2013.

[14] N. Diakopoulos, M. Naaman and F. Kivran-Swaine, "Diamonds in the rough: Social media visual analytics for journalistic inquiry," in *Visual Analytics Science and Technology (VAST), IEEE Symposium on*, 2010.

[15] N. Diakopoulos, M. D. Choudhury and M. Naaman, "Finding and assessing social media information sources in the context of journalism," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012.

[16] S. Petulla, "Feelings, nothing more than feelings: The measured rise of sentiment analysis in journalism," *Nieman Journalism Lab,* 13 January 2013.

[17] J. Ellis, "The New York Times built a robot to help make article tagging easier," *Nieman Journalism Lab,* 30 July 2015.

[18] I. Terrizzano, P. Schwarz, M. Roth and J. E. Colino, "Data Wrangling: The Challenging Journey from the Wild to the Lake," in *CIDR*, 2015.

[19] C. Groskopf, "The Quartz Guide to Bad Data," *Quartz,* 15 December 2015.

[20] Investigative Repoters and Editors, "DocumentCloud," [Online]. Available: documentcloud.org.

[21] Thompson Reuters, "Getting Started with Thomson Reuters Open Calais™ API," [Online]. Available: http://www.opencalais.com/opencalais-api/.

[22] F. Lindenburg, "A Tour Of Aleph, A Data Search Tool For Reporters," 19 July 2016. [Online]. Available: http://gijn.org/2016/07/19/a-tour-of-aleph-a-data-search-tool-for-reporters/.

[23] Overview Project, "Completed News Stories," [Online]. Available: https://github.com/overview/overview-server/wiki/News-stories.

[24] O. Agin, C. Ulas, M. Ahat and C. Bekar, "An approach to the segmentation of multi-page document flow using binary classification," in *Sixth International Conference on Graphic and Image Processing (ICGIP)*, 2014.

[25] A. Gordo, M. Rusiñol, D. Karatzas and A. D. Bagdanov, "Document Classification and Page Stream Segmentation for Digital Mailroom Applications," in *12th International Conference on Document Analysis and Recognition*, 2013.

[26] A. J. B. Chaney and D. M. Blei., "Visualizing topic models," in *Proc. Intl. AAAI Conf. Weblogs and Social Media (ICWSM)*, 2012.

[27] Y. Chen, L. Wang, M. Dong and J. Hua, "Exemplar-based visualization of large document corpus.," *IEEE Trans Vis. Comput. Graphics,* vol. 15, no. 6, pp. 1161-1168, 2009.

[28] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu and X. Tong, "Textflow: Towards better understanding of evolving topics in text," *IEEE Trans Vis. Comput. Graphics,* vol. 17, no. 12, pp. 2412-2421, 2011.

[29] W. Dou, L. Yu, X. Wang, Z. Ma and W. Ribarsky, "HierarchicalTopics: Visually exploring large text collections using topic hierarchies," *IEEE Trans Vis. Comput. Graphics,* vol. 19, no. 12, pp. 2002-2011, 2013.

[30] J. Stray, "The Document Mining Pulitzers," 1 May 2014. [Online]. Available: https://blog.overviewdocs.com/2014/05/01/the-document-mining-pulitzers/.

[31] J. Stray, "You got the documents, now what?," 13 March 2014. [Online]. Available: https://source.opennews.org/en-US/learning/you-got-documents-now-what.

[32] J. Chuang, D. Ramage, C. D. Manning and J. Heer, "Interpretation and Trust: Designing Model-Driven Visualizations for Text Analysis," in *SIGCHI*, 2012.

[33] J. Biskupic, J. Roberts and J. Shiffman, "The Echo Chamber," *Reuters,* 8 December 2014.

[34] "Machine learning in the wild -- #wins and #fails," 6 March 2015. [Online]. Available: https://ire.org/events-and-training/event/1494/1739/ .

[35] S. Higham and S. Rich, "Whistleblowers say USAID's IG removed critical details from public reports," *Washington Post,* 22 October 2014.

[36] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: A case study," *Proceedings of recent advances in natural language processing (RANLP),* vol. 1, no. 3.1, 2005.

[37] C. Teegardin, D. Robbins, J. Ernsthausen and A. Hart, "License to Betray," *Atlanta Journal-Constitution,* 5 July 2016.

[38] B. Poston, J. Rubin and A. Pesce, "LAPD underreported serious assaults, skewing crime stats for 8 years," *Los Angeles Times,* 15 October 2015 .

[39] B. Poston and A. Pesce, "How we reported this story," *Los Angeles Times,* 15 October 2015.

[40] A. Pesce, "Checking the LAPD's crime classifications," [Online]. Available: https://github.com/datadesk/lapd-crime-classification-analysis/blob/master/README.md.

[41] J. Gollan and S. Shifflett, "Federal judge's rulings favored companies in which he owned stock," *California Watch,* 20 November 2012.

[42] A. Bee, "Scraping for journalists reporting examples," [Online]. Available: https://github.com/amandabee/scraping-for-journalists/blob/master/examples.md.