Promoting Data Journalism with Purpose-Made Systems

A case study of the benefits of purpose-made data journalism systems among Norwegian Data Journalists

Vetle Prytz Warholm

INFO390
Master Thesis of Information Science
Department of Information Science and Media Studies
University of Bergen
Spring 2020

Supervisor: Truls André Pedersen

Acknowledgements

I wish to thank Senter for Undersøkende Journalistikk at Media City Bergen for their open call to all students who wished to collaborate with them for a master thesis project. The possibility to freely chose my own project was mostly given by them. Within SUJO I wish to thank Per Christian Magnus for his enthusiasm towards my project and for helping me find and contact relevant interview participants. I wish to extend gratitude towards my supervisor, Truls André Pedersen for giving me free reign over the project and supporting me all the way, despite the project being a little outside of his area of expertise. I also wish to say thanks to Ankica Babic at the institute for helping me out with the NSD application and for always checking up on everyone and making sure we were all doing fine.

Abstract

The research project presented in this thesis is a case study investigating the usefulness of purpose-made data journalism systems. The study consists of two investigations, the first informal and exploratory, and the other more extensive and rigorous. The study features interviews with Norwegian data journalists based in the city of Bergen, which constitutes the main source of data. As part of the research, a prototype purpose-made data journalism system has been developed, based on preliminary findings from the exploratory investigation. The research carried out indicates that there is potential for developing computer systems designed to solve certain specific data journalism systems, concluding with a proposed application.

Table of Contents

Acknowledgements i

Abstract ii

List of Figures and Tables v

Introduction 1

Definitions 2

Abbreviations 2

"Computational" and "Data" Journalism 2

General and Investigative data journalism 3

Background 4

The Promise of Computational Journalism 4

Constraints on Data Journalism 5

DocumentCloud 6

A Call to Arms to Database Researchers 7

"Overview" 8

BBC's Linked Data Platform 9

Methods 10

Case Study 10

Exploratory, intrinsic and instrumental, multiple case, holistic case study 10

Research Questions and Hypothesis 11

Units of Analysis 11

Data Collection - Semi-Structured Interviews with Experts 11

Data Analysis Plan 13

Design Science Research 14

Exploratory Case 16

Requirements Gathering – Informal Interview with Expert 16

Record of initial interview 16

Summary of Interview Findings 19

Prototype 20

Requirements 21

Development 21

Design Notes 23

Technical Notes 23

Features 25

Use as Technology Probe 28

Interviews with Prototype – Main Case 33

Semi-Structured Interviews with Experts 33

Analysis 34

Interview Findings 34

Summary of Findings 39

Results 40

Discussion 40

Distributed vs. Centralized Architectures and the Relativity of User Friendliness 40

FOI-centricity and Tools for Non-experts 42

Unorthodox Data Collection 43

Conclusion 44

References 45

Appendix .... a

List of Figures and Tables

Figure 1. Example of Overview's document viewer interface 8
Figure 2. Hevner et al.'s guidelines for Design Science Research 15
Figure 3. Figure showing a snippet of the admin view from the prototype tool. 20
Figure 4. Figure of design and development processes adapted and extended by Brehmer et al. 22
Figure 5. Figure showing an example of a respondent's view when filling out a form in the current prototype. 25
Figure 6. Figure showing the data table view in the current prototype. 26
Figure 7. Mockup of an anomaly highlighting feature when viewing data in a table. 28
Figure 8. Mockup showing how information extracted by Optical Character Recognition could be presented to the user. 29
Figure 9. Mockup presenting the idea of allowing searching and sorting by advanced metrics when viewing data in table format. 30
Figure 10. Mockup showing an imagined combined SPARQL query builder and graph view feature 31

Table 1. Table presenting themes identified during analysis of interview data. 34

Introduction

This thesis presents a research project investigating the application of Information Science solutions to data journalism problems. Data journalism is likely to become an ever more relevant journalistic method as more and more of the world is registered and represented as digital data. It is also an important tool for maintaining the media's traditional role as "watchdog" of the authorities. But data journalism is a form of reporting that requires expertise, IT-resources, and time, limiting its impact as news organizations' resources are far from infinite. The research presented in this thesis is an attempt to identify ways to promote data journalism by introducing purpose-made data journalism tools. The investigation revolves around identifying problem areas with potential for improvement, and proposing designs for what a purpose-made data journalism tool might be like. The focus of the research is narrowed down to the following two research questions:

RQ1: To what extent can data journalists benefit from purpose-made data journalism tools?

RQ2: When making a purpose-made data journalism tool, is it better to attempt to solve as many data journalism problems as possible with a single "centralized" system, or is it better to make specific programs for specific data journalism tasks?

The first research question seeks to determine whether this line of research is worth pursuing. The second research question is more direct, seeking to determine which is better of two distinct design philosophies for purpose-made data journalism tools. Is it better to attempt to "solve" data journalism with a sweeping "one size fits all" computer system housing everything a data journalist needs, or is it better to create specific, specialized tools for distinct data journalism problems? The second research question thus also begs the question, "what is an example of a specific data journalism problem?". These questions are investigated in the research project presented in this thesis. It is structured as a case study consisting of two investigations, the first informal and exploratory, and the other more extensive and rigorous. The study features interviews with Norwegian data journalists based in the city of Bergen, constituting the main source of data. As part of the research, a prototype purpose-made data journalism system was developed, based on preliminary findings from the exploratory investigation.

Definitions

Abbreviations

Throughout this text, two abbreviations are commonly used that are explained here.

SUJO, "Senter For Undersøkende Journalistikk" is an organization promoting critical and investigative journalism in Norway. The organization collaborates with several major news organizations in Norway and internationally, and is heavily involved in the journalism education programs offered at the University of Bergen and others (SUJO, 2020).

FOI, "Freedom Of Information", is a form of legislation present in Norway and most other Western democracies. Freedom of Information legislation covers many basic rights related to the expression and receiving of information and can be considered a subfield of the right to Free Speech (*Store Norske Leksikon* 2018). In this text, the abbreviation "FOI" is used to refer specifically to the parts of Freedom of Information legislation that covers the public's right to insight into government data. An example of this right in use is journalists or other private individuals requesting access to, or a copy of, data from public sources, usually at the regional or municipality level. This act is referred to multiple times throughout this text as "FOI requests", " FOI access requests", " FOI inquiries", or similar.

"Computational" and "Data" Journalism

Computational journalism is not simply journalism done with computers, journalists have been using computers since they stopped writing on typewriters, without becoming computational journalists. A computational, or simply "data" journalist is a journalist who "actively engages with techniques for the large-scale manipulation of data using computing software to enable new ways to access, organize, and present information", phrasing Flew et al. (2012, p. 157), where a distinction is made between learning how to use computers as tools, and learning computational techniques. To use computers as tools, it is sufficient to learn how to use the specific set of programs that accomplish the specific thing you want to do. Learning computational techniques involve processes such as "searching, correlating, filtering, and identifying patterns", Flew et al. say (2012, p. 158). Extending the definition of computational journalism to not just refer to the use of computer tools in journalism, but to include the use of computational techniques, Flew et al. (2012, p. 158) says, has the implication that it might bring journalists and information technology experts together to develop new tools with the aim of providing information that is "accurate, original, reliable, and socially useful".

According to the *Sage International Encyclopedia of Mass Media and Society* (2020), data journalism is:

> A way of enhancing reporting and news writing with the use and examination of statistics in order to provide a deeper insight into a news story and to highlight relevant data. One trend in the digital era of journalism has been to disseminate information to the public via interactive online content through data visualization tools such as tables, graphs, maps, infographics, microsites, and visual worlds. The in-depth examination of such data sets can lead to more concrete results and observations regarding timely topics of interest. In addition, data journalism may reveal hidden issues that seemingly were not a priority in the news coverage (*Sage International Encyclopedia of Mass Media and Society* 2020).

Whether or not "data journalism" and "computational journalism" are in fact two terms for the same method of reporting is not a relevant debate for this thesis. For the most part, this text will use the term "data journalism" as it currently appears to be the most commonly used term amongst journalists themselves.

General and Investigative data journalism

As the practice of analyzing data to uncover interesting facts is inherently an act of investigation, data journalism is closely related to investigative journalism. They are not inseparable however, as investigative journalism is more closely associated with the "deep dives" into some source material, which is not a necessity for general data journalism, and indeed, investigative journalism does not even need to have anything to do with the digital world at all. A simple system can be set up to automatically report on continuously updated data from an open source and be called data journalism, without being considered particularly investigative.

The distinction is further clarified in a 2015 paper by Uskali & Kuutti. The authors performed a survey interviewing data journalists in Finland, the US, and the UK, and categorized data journalism into two main streams; Investigative Data Journalism (IDJ) and General Data Journalism (GDJ). IDJ being the type where the story may be produced over a significant span of time, often in teams, involving advanced data and computer skills including development of purpose-made software, and data that may have been hard to come by such as leaks or datasets carefully assembled over time. GDJ, on the other hand, refers to "day-to-day" data journalism practices where the story "begins in the data", typically a large publicly available dataset that is analyzed using more "mundane" approaches that do not take as long to perform as a real deep dive.

GDJ and IDJ are useful terms when talking about the distinction of these two forms of data journalism and will be used with or without abbreviation at certain points in this text. Beyond this clarification, a further definition of "investigative journalism" should be unnecessary.

Background

The Promise of Computational Journalism

A 2012 paper by Flew et al. presents an overview of the state of computational journalism as of 2011, and what the authors believed it could become in the near future. They identify three major factors driving the potential value of computational journalism. The first is the increase in publicly available data, primarily from government sources, obtained through official channels or using "underground" sources such as WikiLeaks. The second is the decreasing costs and increasing ease of use of "data-mining software" and other relevant tools, which is made all the more potent in conjunction with the first factor. The third factor is the explosion of forms and channels of online participation and engagement, for example that of various social media sites, that allow journalists to interact with their readers in ways not previously possible. Using the increasing levels of interactivity and social, online engagement as background, Flew et al. describe the evolution of news stories from single published pieces to on-going, emerging, complex stories utilizing intricate interactive interfaces to focus on various events, timelines, or entities to offer "personalized reconstructions of events". This is an interesting prediction, as it seems to rely greatly on the assumption that a large mass of readers will be interested enough in particular news stories to be interested in immersing themselves in the source material of the story on their own initiative.

The first two factors synergize in an obvious way. More data, combined with easier, cheaper tools for working with that data, leads to more value from computational journalism. Flew et al. do not go into any detail concerning the design of better data journalism tools, sufficing to say that better tools make for more cost-efficient data journalism. Several use cases are presented where computational journalism can benefit from technical tools and techniques to offer novel improvements over "traditional" reporting, or to solve problems inherent to working with mass data. Particularly, dealing with massive datasets that may or may not be clean, uniform, and in an appropriate file-format carries with it a large amount of labor in regard to annotation, categorizing, cleaning, and so on. On dealing with this issue Flew et al. largely look to journalistic crowdsourcing as the answer. By splitting a large amount of manual work into smaller tasks and distributing them to a large set of volunteers, an insurmountable task for a single person or a small team is solved in a matter of days. This is a simple idea in theory, but a successful crowdsourcing project is dependent on a lot of factors in practice. Journalistic crowdsourcing has not in the near decade since Flew et al.'s paper become the "go-to" solution for data journalism involving large datasets. Could it be argued that crowdsourcing as the solution to analyzing large corpora of documents is "how they did it in the old days", or at least that was, for various reasons, a much more attractive prospect ten years ago? Flew et al. devotes the topic much discussion, and indeed the technique has been used successfully both before and after their paper was published (Rogers 2009, Meyers 2012, La Nacion 2014), but it remains a "niche" solution only really applicable in extreme cases of massive amounts of data that *cannot* be reliably transformed and analyzed using computational techniques like Optical Character Recognition and statistics. These cases are fewer and fewer as technologies like OCR continue to progress. It could be argued that today, the potential in journalistic crowdsourcing lies in data analysis tasks relying on human cognition that are as yet not satisfyingly "solved" by science.

Flew et al. make an interesting claim when they say that "ultimately the utility value of computational journalism comes when it frees the journalists from the low-level work of discovering and obtaining facts, thereby enabling greater focus on the verification, explanation, and communication of news"

(2012, p. 167). The papers presented under the next heading show that it is not a given that computational journalism necessarily reduces low-level workload.

Constraints on Data Journalism

Fink & Anderson (2015) have written about the organization and constraints of data journalists in the United States in their paper "*Data Journalism in the United States. Beyond the “usual suspects”*". A similar paper by Karlsen & Stavelin (2014) titled "*Computational Journalism in Norwegian Newsrooms*" presents a similar study of the conditions of data journalists in Norway. Borges-Rey (2016) in "*Unraveling Data Journalism: A study of data journalism practice in British newsrooms*" presents a similar look at the data journalism practices among journalists working for mainstream media in the United Kingdom. All three studies use semi-structured interviews with data journalists to explore the state of data journalism practice in their respective countries.

With relevance to this thesis, the most important finding from the American study (Fink & Anderson, 2015) is the lack of time, tools, and manpower, when listing the primary constraints on the "production of data-driven news stories". The lack of time urges data journalists to prefer datasets that are easy to acquire and that require minimal cleaning, at the expense of other potentially valuable sources that require more work. Lack of tools, particularly at smaller newsrooms without developer staff on hand to tailor software for individual projects, constrain the type of content data journalists can produce. Whether it be from a lack of personal programming skills or from a lack of funds for software licenses, journalists are restricted to produce the kind of content that happens to be possible with the tools at hand. The lack of manpower mostly refers to the economic hardships many American news organizations faced during the time of Fink & Anderson's study. It is a point worth noting that data journalism, as opposed to "day to day" reporting and reporting on major news may be a particularly costly form of reporting, likely being among the first to suffer when news organizations staff are required to cut their expenditures (of time as well as funds) to the bare minimum. One of Fink & Anderson's interviewees say they did more data journalism "a decade ago" but has since had to reduce this activity due to their organization's debt (Fink & Anderson 2015). Another related "lack" that is mentioned in Fink & Anderson's paper is the lack of legal resources to battle public officials that are reluctant to release data, or try to excessively charge news organizations for access. This also ties in with the lack of tools, as for example having access to decent OCR software and knowing how to use it makes it easier to deal with public officials providing data in (deliberately or accidentally) unfit or hard to read formats. With the lack of time and other resources leading to American data journalists producing content of whatever shape they happen to be able to, from whichever datasets are most easily available, it seems clear that purpose-made data handling tools for journalists, that are designed to save time, would be a welcome addition in American data journalist's toolboxes.

Karlsen & Stavelin's study of data journalism in Norwegian news organizations (2014) also points to the lack of time as the primary limiting factors on the production of data journalism stories. Time and "goodwill" from editors who have to trust in their data journalists to actually produce something worthwhile from projects that often take a long time to come to fruition. The study cites several interviewees when saying that the access to (public) data in Norway is usually pretty good, and that the required "technical infrastructure" is usually cheap and easy enough to set up, but that "visualization takes time, analysis takes time, and fetching data takes time". And while access to public data is good, the officials providing the data may lack the required technical knowledge to export the data in useful ways, leading to datasets that are unnecessarily hard to work with, like spreadsheets saved as pdfs. Other findings worth noting from Karlsen & Stavelin's study is that the interviewed data journalists do not think it is particularly worthwhile to publish stories containing tools and features that allow the readers to immerse themselves in and explore the source data. Most people are simply not that interested. This contradicts a prediction made by Flew et al. (2012) where they list the ability to allow readers to inspect source data in various ways as one of the promising prospects of computational journalism. Another point where Karlsen & Stavelin contradict Flew et al. is that where Flew et al. say computational journalism would free the journalist from performing low-level tasks (and thereby saving time for more analytical work) (2012, p.167), Karlsen & Stavelin's findings suggest that time is something you spend *more* of when performing computational journalism (2014, pp. 43-44).

Borges-Rey (2016) has performed interviews of data journalists and editors in newsrooms in the UK. The interviewees in Borges-Rey's paper state that some of the best stories produced by their data journalism units are those where the units collaborate with specialized correspondents (i.e. reporters with particular expertise in the field at hand), graphics designers, statisticians, and developers. In the case that the story is yet to be found in the data at hand, the data journalism unit may seek the aid of a specialized correspondent to help understand the significance of the numbers and to provide context/background. In other cases, it might be the specialized correspondents themselves that seek out data journalists for help in providing fact-checks or empirical evidence/numbers for use in other stories. Interviewees also stated that they believed that "data literacy" will become an essential skill for all journalists in the near future. Some interviewees expressed frustration at not being capable of programming their own software to handle their data, as third-party ready-made tools were often incompatible with their organization's systems and/or not perfectly applicable to the problem at hand, a problem also mentioned by Fink & Anderson (2015), where some of their interviewees express frustration at Content Management Systems that do not support content from third party systems.

What can be gathered from the above three studies on the state of data journalism in the US, the UK, and in Norway, is that data journalism is a form of reporting that promises accurate, factual stories grounded in data, at the cost of often being a time-consuming, arduous process. Data journalism may discover important news where other forms of reporting would just encounter a wall of data, but the task of tearing this wall down requires tools. A primary hurdle for the successful execution of data journalism projects is the amount of time it takes for the project to produce anything worthwhile. One factor that increases the amount of time it takes to produce a data journalism story is the use of computer tools and systems that need to be acquired, adapted to the task at hand, and then learned by the journalists before they can be of any use. I therefore argue that it is worth exploring whether it may instead be beneficial to collect commonly used data collection and analysis features into a single dedicated data journalism system designed to allow data journalists to do as much work as possible using only one tool.

DocumentCloud

DocumentCloud is a web-based platform allowing users with registered accounts to upload, annotate, review, and share vast corpora of documents publicly. All documents uploaded to DocumentCloud are processed by "Tomas Reuters OpenCalais", text-processing software providing entity-extraction and revealing various factual features like dates and times (DocumentCloud, 2019). DocumentCloud offers both private and publicly visible annotations and highlighting of documents. At the individual user's behest, private documents can be made public, joining them with the pre-existing public documents catalog. Documents hosted by DocumentCloud can be embedded on websites, allowing newsrooms to make visible the primary source documents they base their reporting on. Other tools can use DocumentCloud as a backend to host documents while providing their own frontends for purposes like crowd annotations or organizing and viewing a large document corpus. As of 2019, DocumentCloud is an independent organization, based in Philadelphia, PA. They have previously been affiliated with the IRE (Investigative Reporters and Editors). DocumentCloud is opensource and free to "journalism organizations", with the code available on the project's Github page[1] (DocumentCloud, 2019). DocumentCloud is an example of a purpose-made data journalism tool offering a specially selected set of features to allow journalists to use the system effectively. The primary use case of DocumentCloud is perhaps mostly crowdsourcing projects, though features like textual entity-extraction and the sharing of annotations are useful also in a wider range of applications where large sets of documents are involved. The platform was originally developed by a small team from the independent investigative newsroom *ProPublica* (DocumentCloud, 2019), making DocumentCloud an example of programmer journalists creating their own tools. The platform's existence and origins may be argued to support the idea that similarly purpose-made tools offering a selection of features gathered in a single system may be useful also in other data journalism projects.

A Call to Arms to Database Researchers

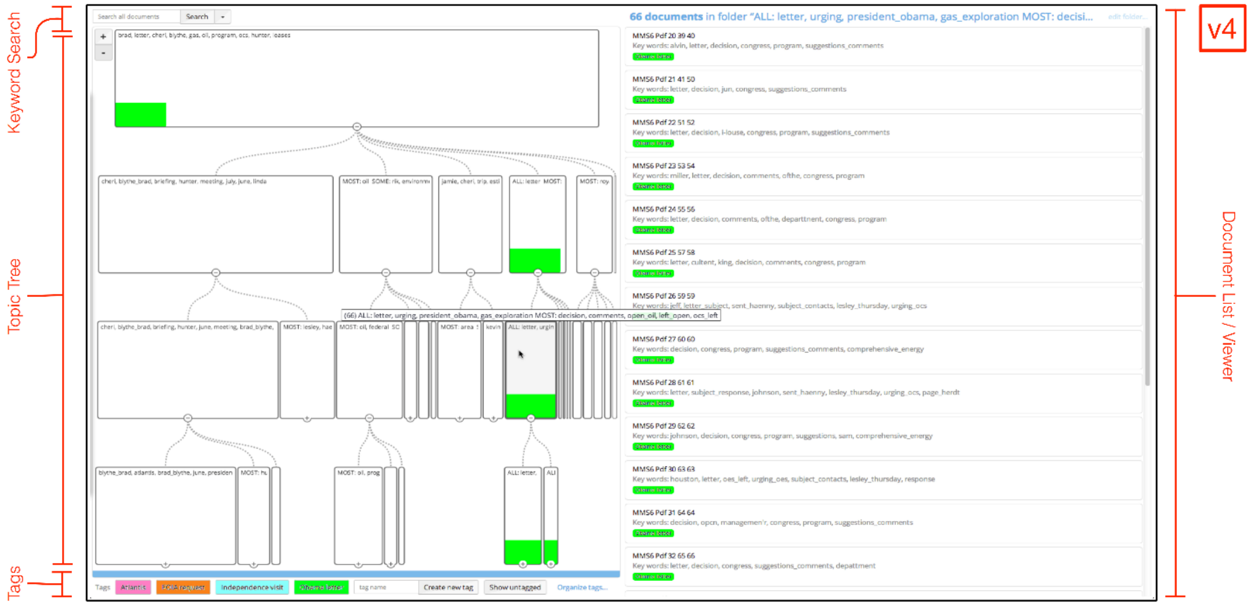
"Computational Journalism: A Call to Arms to Database Researchers" by Cohen et al. (2011) presents some interesting ideas for data journalism systems. DocumentCloud is mentioned as "a pioneering example of a service that can host original and user-annotated documents", while also providing tools for processing and publishing of said documents, again showcasing it as an example of a successful purpose-made data journalism system. Cohen et al. also present the interesting idea of a "reporter's black box"; a tool executing a set of queries that could be considered "standard" or "sensible" on a structured database, automatically producing useful statistics and patterns where such exists. Part of the idea is that the system will discover more useful query "templates" over time, with a ranking system used to keep track of the most useful templates for specific types of datasets, based for example on their use in "high-impact stories". Cohen et al. exemplify by looking at perhaps the only field of reporting where such a tool has been highly successfully deployed, namely sports. Statistics detailing how many specific actions have been performed by a specific player, scoring how many points, and in which games across which timespan are often provided as commentary during many sports events, as if pulled out of a magic hat. Achieving something similar in investigative journalism is a multi-faceted problem, including issues of funding as well as the complexity of the information one may be interested in, and the availability of data. Nevertheless, the vision of a "reporter's black box", as Cohen et al. describes it, that given a relevant dataset can instantly provide a set of statistics about entities of public interest makes a compelling case for the use of structured/semantic data technologies when constructing and organizing journalistic databases.

[1] https://github.com/documentcloud

"Overview"

Overview is a tool intended to help journalists *explore* comprehensive sets of documents, using various metrics to gauge document similarities and presenting them as clusters in a hierarchical tree. In addition to the automatically generated clusters, users of the program can also annotate documents with custom tags. A paper by the makers of "Overview" contains a presentation of Overview's design process and iterations, its current user interface, a presentation of case studies performed to evaluate the program in voluntary use by journalists, and an overview of the program's "design rationale" (Brehmer et al. 2014). The paper makes several points worth noting should one wish to develop a similar purpose-made data journalism tool, both in terms of features and metrics used in evaluation.

A good interface for viewing the overarching structure (if one is discovered) of a given dataset is a very useful feature that Overview offers. On the design rationale of Overview, among other things, Brehmer et al. outline the reasoning behind presenting the document corpus at hand as a tree, as shown in Figure 1. By representing clusters of documents as nodes in a tree, where the width of the node represents the number of documents it contains, as wells as allowing said nodes to be tagged and labelled with user-generated tags, Overview provides an example of a visualization method for large sets of annotated documents that would-be developers of similar systems could learn from.


Keyword Search
Topic Tree
Tags
Document List / Viewer
v4
Search all documents
Search
+
-
brad, letter, cheri, blythe, gas, oil, program, ocs, hunter, leases
cheri, blythe_brad, briefing, hunter, meeting, july, june, linda
MOST: oil SOME: rik, environm
jamie, cheri, trip, esti
ALL: letter MOST:
MOST: roy
(66) ALL: letter, urging, president_obama, gas_exploration MOST: decision, comments, open_oil, left_open, ocs_left
cheri, blythe_brad, briefing, hunter, june, meeting, brad_blythe,
MOST: lesley, hae
MOST: oil, federal SC
MOST: area
kevin
ALL: letter, urgin
blythe_brad, atlantis, brad_blythe, june, presiden
MOST: hu
MOST: oil, prog
ALL: letter,
ALI
Tags
Atlantis
FOIA request
Independence visit
Obama letter
tag name
Create new tag
Show untagged
Organize tags...
edit folder...

66 documents in folder "ALL: letter, urging, president_obama, gas_exploration MOST: decisi...

MMS6 Pdf 20 39 40
Key words: alvin, letter, decision, congress, program, suggestions_comments
Obama letter

MMS6 Pdf 21 41 50
Key words: letter, decision, jun, congress, suggestions_comments
Obama letter

MMS6 Pdf 22 51 52
Key words: letter, decision, I-louse, congress, program, suggestions_comments
Obama letter

MMS6 Pdf 23 53 54
Key words: miller, letter, decision, comments, ofthe, congress, program
Obama letter

MMS6 Pdf 24 55 56
Key words: letter, decision, comments, ofthe, departtnent, congress, program
Obama letter

MMS6 Pdf 25 57 58
Key words: letter, cultent, king, decision, comments, congress, program
Obama letter

MMS6 Pdf 26 59 59
Key words: jeff, letter_subject, sent_haenny, subject_contacts, lesley_thursday, urging_ocs

MMS6 Pdf 27 60 60
Key words: decision, congress, program, suggestions_comments, comprehensive_energy
Obama letter

MMS6 Pdf 28 61 61
Key words: letter, subject_response, johnson, sent_haenny, lesley_thursday, urging_ocs, page_herdt
Obama letter

MMS6 Pdf 29 62 62
Key words: johnson, decision, congress, program, suggestions, sam, comprehensive_energy
Obama letter

MMS6 Pdf 30 63 63
Key words: houston, letter, oes_left, urging_oes, subject_contacts, lesley_thursday, response
Obama letter

MMS6 Pdf 31 64 64
Key words: decision, opcn, managemen'r, congress, program, suggestions_comments
Obama letter

MMS6 Pdf 32 65 66
Key words: letter, decision, congress, suggestions_comments, depattment
Obama letter

Figure 1. An example of Overview's document viewer interface (Brehmer et al. 2014)

Also of note in Brehmer et al. (2014) is the mention of the importance of "simplifying for infrequent use and reducing data wrangling", by which the authors refer to earlier experiences with Overview version 2. Here it became evident that a major hurdle for prospective users of the system was the fact that Overview version 2 only supported document imports as csv-files. They state that "we quickly learned that journalists receive document collections in every conceivable format" (Brehmer et al. 2014). In addition, many users also apparently had difficulties manually downloading, installing, and correctly configuring the tool. As such later versions of Overview (v3-v4) are web-based with no requirement of local configuration or installation and supporting import of folders of pdf-documents as well as importing documents directly from DocumentCloud (as DocumentCloud already supports importing documents from a wide array of sources). This is an argument for others who would develop systems for journalists specifically, that the system must support data in "every conceivable format" and that it should be made web-based to eliminate the hassle of local installation.

Finally, Brehmer et al. proposes an interesting method of evaluating the success of their design. They say that "adoption" of the tool is their chosen measure of success, where adoption means "repeated instances of self-initiated use". Multiple case studies on specific uses of Overview by journalists, conducted over time across different deployments of the tool were necessary to achieve a clear understanding of users' needs. "Adoption" as an evaluation metric for an artifact is perhaps more applicable to research projects like Overview that go on for several years than it is to a master thesis. One way to instead "simulate" adoption as an evaluation metric could be to ask journalists who have explored the object of evaluation whether they would be interested in making use of a finished version of it. This would of course be a much more light-weight metric that must be viewed with skepticism.

BBC's Linked Data Platform

When searching for cases where semantic technologies are actively used by news organizations to organize data and assets and to assist in reporting, it is difficult to find anything other than BBC and their "Linked Data Platform". BBC's Linked Data Platform is a framework that stores a "generic metadata model" of all creative works across the various Content Management Systems found within the organization, combining data from different systems to allow connections to be made between all manner of "things" that are found in the resulting RDF graph (BBC 2013). The Linked Data Platform is built upon the "Dynamic Semantic Publishing" platform developed earlier for the publication of automated metadata-driven webpages for the football world cup of 2010. The "sports data" origin extends also to the Linked Data Platform, being developed to create individual athlete pages and more for the 2012 Olympics. The basis for the Linked Data Platform is formed by a set of ontologies covering the various types of information the system handles (BBC 2014). These ontologies are publicly available[2], and the Linked Data Platform itself can also be accessed indirectly by the public via the service "BBC Things[3]" (BBC 2014).

The work done by the BBC to link their data using semantic technologies allows them to make connections between creative works, public or private entities, content producers, places, etc. in ways not previously possible. Their work shows that there is no doubt about the feasibility and usefulness of the application of semantic technologies to the data owned and managed by news organizations. However, the Linked Data Platform is metadata-driven, it does not extend to the content of any creative work, like a news story, that is a much more complex task. What the Linked Data Platform does show is that the linking of data from different systems, and the making of connections between disparate types of "things" are applications where semantic technologies are currently highly applicable. An imagined data journalism tool could be made to integrate with a system like the Linked Data Platform, or even lay the foundation for such a system, by using an RDF script to automatically create triples about collected data. In this scenario, an interesting quality would be to capture as much as possible relevant metadata about the entities involved, and to which projects they provided data and when.

[2] https://www.bbc.co.uk/ontologies

[3] https://www.bbc.co.uk/things/

Methods

Case Study

A case study is a method of research applicable to a wide range of scientific areas, and as such they are defined and explained by many different sources. For my thesis project I rely on the definition of a case study and their requirements as presented by Lazar et al. in *Research Methods Human-Computer Interaction* (2017). A case study can take many forms, but is generally recognized as being a close investigation of a topic using only a small number of cases. The reason for using only a few sources might be lack of available cases, lack of time because the in-depth investigation (for example via lengthy observations or repeated interviews) is time-consuming, or because the nature of the investigation is such that it is best carried out in a qualitative manner. My research project is designed according to the requirements of a case study as presented by Lazar et al. (2017, chapter 7).

Exploratory, intrinsic and instrumental, multiple case, holistic case study

My thesis project falls into the general category of an exploratory case study. I seek to understand a problem and to inform a new design to solve that problem. I investigate the context of technology use that is (digital) data journalism, and how this technology use could be improved by introducing better software tools. A case study is either *intrinsic* (the case is very particular and results are likely to apply only to a narrow range of other cases), *instrumental* (the goal of the study is to inform designs and solutions applicable to a wider range of situations), or both (where the results are interesting in their own right, but also provide broader understanding applicable elsewhere). I consider my study to be both, as I am investigating cases from a small crowd (Norwegian data journalists) but hope to achieve understanding that is of interest also to data journalists in other countries, to other staff in news organizations, and to other researchers and developers investigating similar topics.

This study uses *multiple cases* (2) to achieve a more accurate understanding of the subject of research. The study relies on interviews with multiple Norwegian data journalists with various backgrounds that justify calling them "expert users". This is one case with multiple participants. The participants are interviewed separately, but they are not discussed individually or treated as distinct units of analysis. The other case is the initial, exploratory interview with a data journalist that preluded the main body of work involved in the thesis project. This interview was entirely informal and unstructured and is considered a unit of analysis separate from the interviews with the other journalists. A case study is *embedded* if it addresses multiple units of analysis in a single case, as opposed to *holistic* where each case investigates only a single unit of analysis. This case study uses two cases that are distinct from each other. Although one case consists of multiple participants, these are discussed together as a single unit. As such, this study is a multiple-case *holistic* case study. One case and one unit of analysis for the initial exploratory interview, and one case and one unit of analysis for the series of expert-interviews that followed towards the end of the project.

To summarize, according to key points presented by Lazar et al. (2017) in their definition of a case study as a research method in Human-Computer Interaction, this case study is e*xploratory*, *intrinsic* and *instrumental, multiple-case*, and *holistic*.

Research Questions and Hypothesis

The study attempts to answer the following two research questions:

RQ1: To what extent can data journalists benefit from purpose-made data journalism tools?

RQ2: When making a purpose-made data journalism tool, is it better to attempt to solve as many data journalism problems as possible with a single "centralized" system, or is it better to make specific programs for specific data journalism tasks?

Hypothesis

The hypothesis for this research project was to find that journalists are positive to the idea of purpose-made data journalism tools intended to make their jobs easier or less time-consuming by gathering several commonly used features into a single system where these features are otherwise available only via separate tools. An example is Optical Character Recognition, which in many implementations has to be accessed via console commands even when locally installed. A premise for the hypothesis is that many data journalism projects involve many of the same steps, thereby making it possible to save time or otherwise improve upon the status-quo by allowing these actions to be taken without requiring that the data journalist switch systems or move data.

Units of Analysis

As mentioned in the general description of the case study, it features two cases, each a distinct unit of analysis. They differ greatly both in their execution and purpose.

The first case is the initial exploratory expert interview. This interview was completely unstructured and informal, functioning like an "informal case study" as described by Lazar et al. (2017, pp. 180-182) where theoretical backgrounds and analytical frameworks are put aside in favor of a simple observation to gauge whether a point of research is worth pursuing. The interview was informal and relaxed, notes were taken by hand, and analysis of the data was performed without any defined method in mind. The purpose of this investigation was to gauge whether or not "dedicated data journalism tools" were a worthy line of pursuit for this thesis, and if so, what a prototype system should include.

The second case, and the second unit of analysis, is a small series of interviews with Norwegian data journalists. Three journalists were interviewed, each interview lasting between one and one and a half hours. More journalists were originally interested in contributing to the project, but were unable to due to various reasons stemming from the Covid-19 pandemic. The methods of data collection and data analysis used in this unit of analysis are detailed in the following paragraphs. Unlike the first unit of analysis featured in this study, the later series of interviews were intended to provide more conclusive data on the research questions.

Data Collection - Semi-Structured Interviews with Experts

The chosen method for data collection for the main case of the study was the use of semi-structured interviews. This was a straightforward choice considering the nature of the data being collected; an evaluation of an artifact both for the purpose of improving on the artifact's design but also to find out whether the artifact and its proposed designs actually offers solutions to problems encountered by a wider range of experts. Had the purpose of the interviews been purely to evaluate the artifact, or had they been usability-tests, tabulatable data collected with a fully structured interview may have been more practical, but as it stood, the wide, exploratory nature of the investigation was more easily handled with a qualitative approach. Not to mention the probability that only a small number of participants would be available, barring any quantitative method. Semi-structured interviews permit the interviewer and interviewee to deviate from the "script" enough to explore topics that may emerge unplanned, or to devote more or less time to certain topics depending on which questions or tasks the interviewee responds most enthusiastically to. The qualitative nature of data collection with semi-structured interviews is also highly practical when working on a research project alone.

Selecting Participants

The selection criteria for interview candidates were simply such that the interviewee must justifiably be able to be called an "expert". In this case that meant journalists that could in some way be called "data journalists" or "computational journalists" or journalists otherwise associated with a data journalism unit in a way that would give them insight into the data journalism practice within their news organization. Furthermore, the journalists must have some kind of experience with using digital data tools to produce news stories, which would give them some kind of idea of what they want from a computer program in this context. The number of desired interviewees was not explicitly determined, as it was assumed that under the circumstances (targeting data journalists working in Bergen and relying on contacts within SUJO to find them) it was unlikely that more than a handful would be available. Discussing the topic with my supervisor we determined that the more the better, but no more than "about five" would be necessary and three to four would be perfectly okay, given the qualitative nature of the investigation, the expected difficulty in finding participants, and later the difficult work situation caused by the Covid-19 pandemic.

The selection criteria were communicated to a key contact within SUJO and was met with a list of ten names of journalists working in Bergen that the contact suggested should be spoken to. The contact sent the listed journalists a primer via email describing the thesis project and letting them know they would shortly be contacted about participating in interviews, this correspondence happening on the 9th of March. All the listed journalists were directly contacted via email a little while later, on the 19th of March. Despite several of the potential participants quickly expressing interest in response to the primer sent out by the SUJO contact, the initial response to the interview invitations was underwhelming, probably related to the oncoming Covid-19 pandemic keeping many of the journalists busy. Multiple subsequent reminders about the research project were necessary to gather enough interviewees.

NSD Approval

When performing expert interviews where precise details on the nature of the subjects current or previous work, their workplace, and other potential personal details, it is necessary to obtain appropriate approvals. Any Norwegian research project that handles personal data in any way must be approved by the Norwegian Centre for Research Data (NSD) via an application describing the project's purpose, scope, data collection and handling plan, etc. The NSD application for this research project was sent the 13th of March and approval was received on the 16th. A copy of the NSD approval is included in this document's appendix.

Interview Guide

The interviews were planned such that interviewees should be able to talk freely about what they were most enthusiastic about. As such, the various stages of the interview were planned to last for appropriate amounts of time overall, but with exactly how the time slices would be spent being undefined. The interview guide was split into three sections, the first mainly involving question about the interviewed journalist and their prior experiences, the second focusing on the prototype and discussion of the applications of various technologies in the context of journalism, and the third being devoted to recapping and "closing down". The entire interview was planned to last for one to one and a half hours. Some questions in the interview guide were mostly there in case the interviewee had to be "prodded" to keep talking, and were in some cases never asked. A copy of the interview guide is available in the appendix.

Adapting the Interview Following the Covid-19 Restrictions

It was initially planned that the interview-stage of the project would begin the week before Easter and conclude the week after, with interviews being held physically in a room at Media City Bergen. The interviewee would be shown the prototype system locally installed on a laptop, with mockups and other resources also locally available. The interview was to be recorded with an audio recording device. This plan was invalidated when the first major restrictions were enforced to combat the Covid-19 pandemic. Adapting the interview procedure to be carried out remotely required deployment of the prototype system to an online platform, this process is described in the chapter describing the prototype. The mockups, consisting of images, were numbered and organized into a single pdf which was distributed to interviewees by email. At certain points during the interviews, interviewees were simply asked to look at the appropriately numbered mockup. Digital copies of the consent form were also distributed via email. Getting the consent form signed and returned this way presented problems as not all interviewees had access to a printer and/or scanner at home and were unable to use their workplace facilities because of the pandemic restrictions. The interviews were held using "Zoom[4]" as the digital meeting provider. As a positive side-effect of performing the interviews digitally, getting good recordings became much easier, using Zoom's built-in recording feature. It was considered unnecessary to perform the interviews using full video chat, as most of the interview would be conducted with the screen being used to display a shared screen of the interviewee exploring the prototype. With these adaptations, interviews were held successfully and in accordance with the interview guide, albeit over a delayed schedule. The interview phase of the project was extended to last until the end of April to accommodate for busy interviewees.

Data Analysis Plan

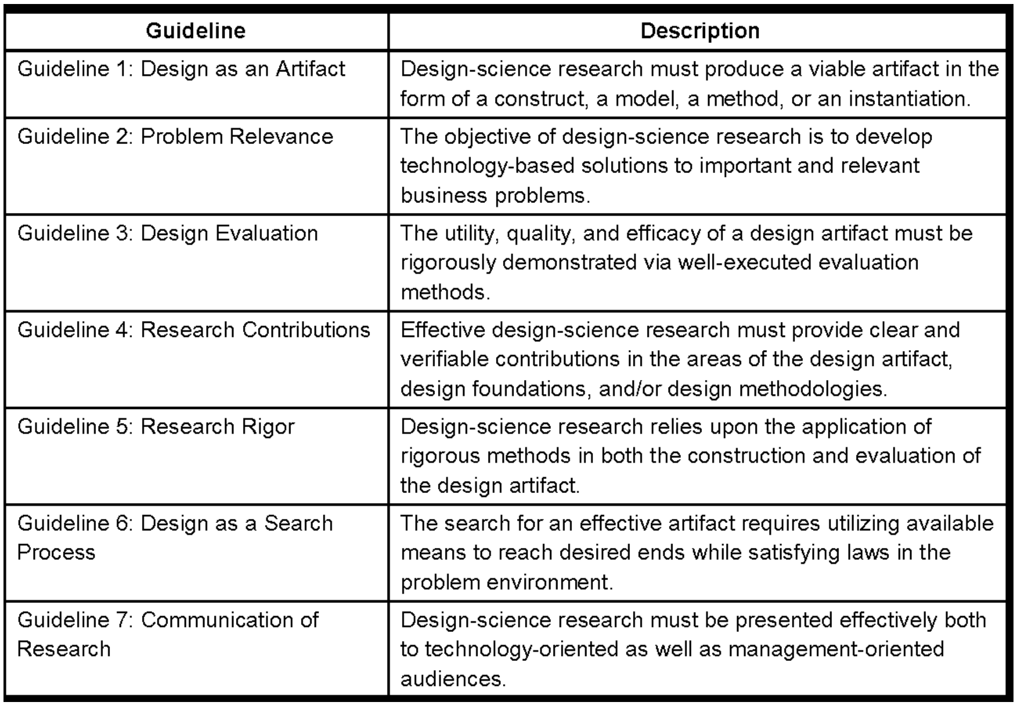
Different approaches to data analysis were considered, and employed, at different stages of the research project. The exploratory case, being informal and loosely planned in advance, produced interview data in the form of handwritten notes and a "record" of the interview based on those notes as well as fresh memory. No accurate transcript was made. The exploratory interview had a very clear purpose and the interviewee was enthusiastic, so the interview proceeded in such a fashion that the resulting interview record became a list of relevant findings by itself. It was determined that no further, formal analysis was needed, the information emerged from the data naturally.

[4] https://zoom.us/

For the main case, the analysis plan was only loosely defined prior to data collection because of uncertainty of how the interviews would be carried out; in person, or remotely. Planning data analysis was postponed until after it became clear that the interviews could not be executed as originally planned, and after the relevant adaptations to the interview procedure were made. The interviews would produce data in the form of audio records, including video of the shared screen but not of the participants, and computer notes taken along the interviews' duration. It was determined that producing accurate transcriptions of the interviews from the raw data would be unnecessary. Without being in close proximity to the interviewee, details on body language and other unspoken cues were lost, removing one useful aspect of potential transcriptions. Time-constraints were another factor. Furthermore, the interviews were low enough in number, and of manageable durations, to permit analysis of their contents without rigorously constructing word-by-word representations of the raw data. Instead, the data from the three expert interviews, in the form of computer notes and audio records, were collected into an aggregated record of their content. This record was then encoded using text highlighting with different colors to represent different themes that the units of text related to. This approach is a simplified version of an encoding scheme using both emergent and a priori codes, as described by Lazar et al. (2017, pp. 303-311). Going by the book, it would be appropriate to begin by annotating small units of text with keyword codes before counting, comparing, and grouping these codes to discover overarching themes. This step was skipped, as the main themes were discovered by themselves prior to the formal analysis, becoming apparent already during the interviews or during aggregation of the data. Some themes also emerged during the analysis itself when discovering that some previously identified themes needed to be split into narrower forms to represent the contents of the text more accurately.

Design Science Research

The research project behind this thesis is in some ways a work of Design Science Research. In broad terms, a problem has been identified, an artifact has been developed to attempt to solve that problem, and an evaluation of the artifact has been carried out to gauge to what extent it "made the world better". This is an oversimplification of the research process, and does not exactly capture the goal of the investigation, but the similarities between the design of the study and the method of Design-science research are great enough to warrant mentioning. Hevner et al. (2004) have defined seven criteria that function as guidelines for how Design-science research should be carried out, shown in Figure 2. Some of these guidelines, like number 4 and 5, are variations of rules that generally apply to all scientific research. In an attempt to describe how this research project fits or does not fit the description of Design-science research, some of these guidelines are compared to the design of the study in the paragraph below.



Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Figure 2. Hevner et al.'s guidelines for Design Science Research (Hevner et al. 2004)

Guideline 1 states that Design-science research must produce a viable artifact. The artifact produced as part of this thesis is a prototype tool for collecting journalistic data with web-forms in a system designed to save journalist's time. In accordance with guideline 2, the objective of this research is to develop a technology-based solution to a relevant problem; the problem being that computational journalism takes a lot of time to do, often because journalists struggle with adapting general-purpose tools.

After developing a functional prototype, guideline 3 requires "rigorously demonstrated evaluation". This point is *not* met in this study. The produced artifact is used primarily as a technology probe during interviews with experts, to promote discussion of the overarching theme of "purpose-made data journalism systems". This approach will doubtlessly lead to *some* evaluation of the artifact in its contemporary state, but evaluation is not a chief concern.

According to Dresch et al. in *Design Science Research: A Method for Science and Technology Advancement* (2015), Hevner et al.'s guideline 6 simply states that the Design-science researcher must "conduct research to understand the problem and to obtain potential problem-solving methods". This research project is in fact focused more towards understanding the problem, or indeed, exploring whether or not there even really is a problem, than it is about evaluating the artifact meant to help solve said problem.

Exploratory Case

Requirements Gathering – Informal Interview with Expert

The first case of this study was an informal interview with an expert user with recent experiences from a large data gathering and analysis task. The interviewee was a master student of Investigative Journalism about to finish his thesis, during which he made extensive use of the general-purpose online survey/form generator tool called "Skjemaker[5]" (which is hosted by the University of Bergen) to collect data from Norwegian municipalities (the forms/surveys were labelled as freedom-of-information access requests, Norwegian "innsynskrav"). Skjemaker allows users to design forms using a selection of different fields, parameters for what is allowed to be entered into said fields, and logic determining which fields to show or hide depending on answers given for previous fields.

The interview was conducted on the 12th of September 2019, and lasted for less than two hours, with my supervisor attending in addition to myself and the interviewee. The interview, being completely informal and unstructured in nature, was not digitally recorded, but handwritten and computer notes were taken both by me and by my supervisor. Shortly after the interview, a record (not a transcription) of its contents was written based on these notes as well as fresh memory. This record follows below, slightly rewritten for clarity. Any quotes are English translations by me, of the original Norwegian words.

Record of initial interview

The interview began with short introductions and quickly turned to the matter at hand. The interviewee began his account of Skjemaker's various shortcomings. He went about it in no particular order, providing some visual examples as he went along.

First on the list were difficulties related to filetypes and structure. Both the system for allowing respondents to upload this and that, and also the system for presenting respondents' uploads to the journalist are weak. The interviewee stated that it would be good to be able to see who has uploaded how many files, presented in a neat folder structure instead of the current system, which shows each individual file-upload in their respective fields in each form for each respondent (that is, not in a directory like form/field/respondent, but rather that each file had to be retrieved from the web-interface showing a single respondents answers to a single form).

Respondents also need to be able to save and resume their work at a later time as some forms may be longer and more time-consuming than the respondents first anticipate. They also should be able to make edits to their answers or reupload files without having to go through the entire form from start to finish again. Given the ability to edit answers, the interviewee says, the system needs to alert the journalist in the event that this happens.

The interviewee reported a general presence of bugs, and poor or simplistic design of many of the features of Skjemaker. Bugs are one thing, and while undesirable, are "always" to be expected to some degree, the interviewee stated. It is another that the status of the service and/or the hosting server is not visible anywhere to any of the users (journalist or respondent alike). The interviewee expresses frustration at constantly having to call the staff at the university IT desk to ask them to please check their server because the tool was not working or was unresponsive, without it being apparent if the

[5] https://skjemaker.app.uib.no/

server was at fault or if the problem was somewhere else. The interviewee suggests a status icon showing the status of the service and/or server would be a good addition to an improved tool.

Another feature that was found wanting was the view of the form's fields and questions when designing the form's logic. If a question or field is too long to present in the drop-down box, one must go by their respective number instead. This leads to an exercise of "linking numbers" as stated by the interviewee.

There is no feature in Skjemaker for "inviting" respondents to your form. A link to a form must first be generated within the web-interface, and then sent via email, outside the interface, to a list of respondents. Being able to do this from within the system would be an upgrade, the interviewee says. It is volunteered by the interviewer that this would probably allow for easier monitoring of each respondent's progress in filling out the form, which was another feature desired by the interviewee, as well as facilitating direct correspondence between respondent and journalist within the system.

As a word of caution, the interviewee stated that an improved system must not be "too static" (interviewee's words could also be translated to "too rigid") as there will always be respondents who only partially fill their forms out, or do so without properly following the stated parameters of certain fields (knowingly or not, as many respondents misinterpret questions or just plainly don't read the instructions). The system must be able to handle a degree of variation among answers as well as file-uploads.

The interviewee raised some concern about some respondents reporting that their email containing the link to the form was blocked by their email spam-filter. The interviewee states that he suspects some, or even most, of these reports to be "lies and excuses" from those municipality officials with little love for nosey journalists, given when pressed about why they had not yet responded to the journalist's requests (which they are legally required to do as long as the request is specified to be a freedom of information request, Norwegian "Innsynskrav"). Another related issue was broken invitation links, though this was suspected by both sides of the interview to be a matter of the respondents' email clients (or other involved programs) "cutting" the URL of the link because it encountered a border and had to be split between two lines (possibly because of excessively long invitation URLs), and as such is not necessarily an issue with Skjemaker per se. The interviewee reports respondents questioning the system's compatibility with various widely used web browsers, though the reported issues of this type were dismissed by the interviewee as "bogus" used by uncooperative respondents to avoid admitting they were delaying their response.

A simple feature that the interviewee says should really be available is the ability to change the banner displayed at the top of the page when answering a form. In Skjemaker, this banner is the University of Bergen logo, which is likely to raise a few eyebrows when the request for information is coming from a journalist presenting himself as working for a major newspaper. In addition to changing the logo displayed in the banner, the interviewee suggests, it should be possible to include a nice-looking "business-card" or similar containing the journalist's (or other affiliates') relevant contact information.

The interviewee described massive difficulties encountered during the first "deployment" of his form using the Skjemaker system. The form was issued to all Norwegian municipalities and counties at once, via email outside the system itself, and from one moment to the next, the interviewee's personal phone was flooded with calls from people wanting to ask him questions. The initial surge left him exhausted and clueless about how to handle the situation. An improved system should have features designed to mitigate this eventuality. All participants of the interview agree that the easiest solution would be to issue invitations to the form in appropriately sized batches. Another idea presented by the interviewee was to assign contact information for different journalistic team members on forms going to different respondents, so they don't all call the same person for questions and support.

After the initial surge of phone calls, the progress towards "completion", that is all respondents having finished filling out the form and the dataset being complete, was slow. A system for easily or even automatically presenting preliminary findings from partial data would be a huge benefit, the interviewee says. The interviewer here makes a note that it is possible to experiment with the idea of a "Reporter's Black Box" (as presented by Cohen et al. 2011), that exposes a dataset to a set of pre-selected queries designed to detect the most interesting aspects of the data, like outliers and features that suddenly change over a short span of time. Such a feature would not require downloading the entire dataset but would be available in the system's web-interface.

The question was raised between all attending whether a system like an improved form of Skjemaker should be ran in the cloud "somewhere", or in-house on the relevant newsroom's own servers. In the latter case, things like data security and sensitive information would be less of a concern for the developers and users of the actual tool, as these issues would likely be addressed and taken care of by the newsroom's own dedicated IT staff. On the other hand, running the system in the cloud, i.e. on some third-party server, as is the case with Skjemaker, is more likely to keep the system available to freelance journalists and journalists working for smaller newsrooms without the resources to maintain their own dedicated IT facilities.

The interview then turned towards discussion of automatic knowledge extraction from uploaded files. The interviewee explained that in the case of pdf-uploads containing for example invoices and pictures of receipts, he already used a simple scraper-script to extract information and write it to a new file. It was agreed between the attending that a system extracting key data from strictly formatted documents like invoices would be both beneficial and realistically possible. A system could read these files using some form of OCR and potentially aggregate the results in csv format, while also providing links to the source file-upload for manual inspection.

An issue encountered by the interviewee was that it was hard to sort and search the files uploaded by respondents (after he had manually compiled them in an appropriate folder structure) because they were all named whatever the respective respondents had deemed appropriate when creating them. Without any structure in the naming of the files, the interviewee had difficulties keeping track of what was what. An improved system, he suggests, could rename uploaded files according to some predefined structure, keeping the original filename stored as metadata in some fashion.

The interviewee explained that, using Skjemaker, while the journalist designing the form can designate whatever text they wish for each field and thus provide explanations for ambiguous questions where necessary, he thinks that a simple "help" button or question mark icon would have been a helpful feature. Some questions confused more respondents than others, it should have been possible to create expanded explanations or tooltips for such "difficult" questions, "a simple little thing that might save someone some headache" the interviewee said.

The interview concluded with the interviewee reiterating that people write differently, saying "you will never get homogeneous data" and that the Skjemaker tool was buggy, had poor UI, and that it was generally a POS even though it did get the job done.

Summary of Interview Findings

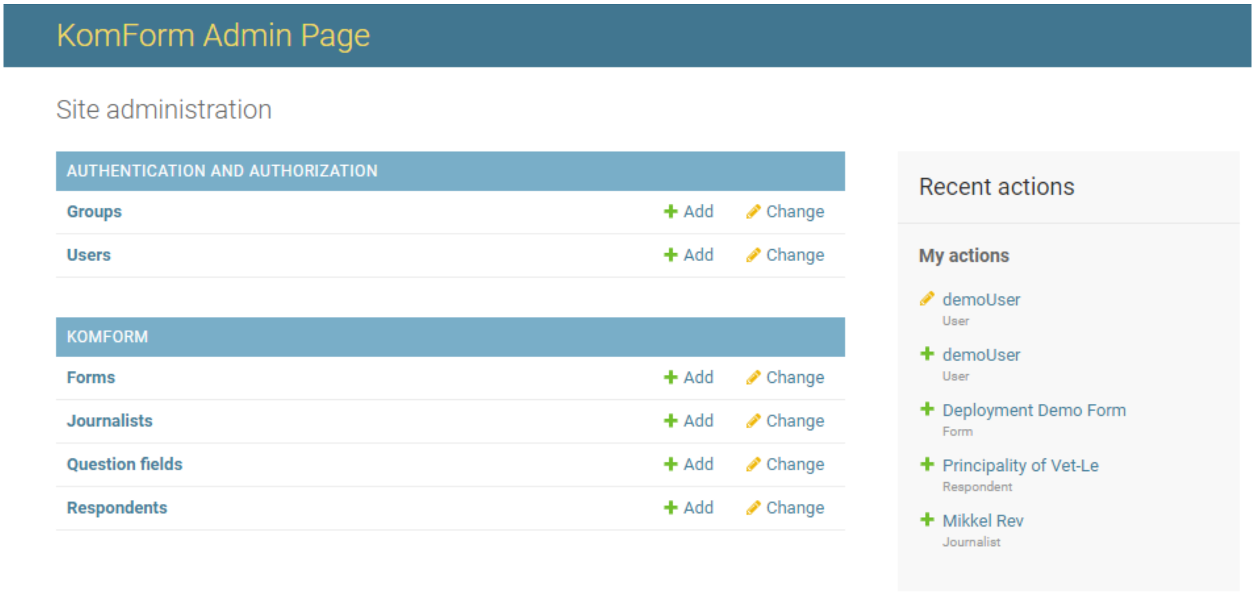
Following the interview and subsequent record-writing, the key findings were identified simply by going through what the interviewee had said. He was very concrete and to the point, so the immediate result of the interview was already practically a list of requirements, no particular coding or analysis necessary. The findings are listed here:

- Ability to see which respondents have uploaded how many files
- Ability to view and/or download files in a "neat" folder structure
- Respondents' ability to resume and edit previously or partially filled out forms
- System should alert journalist in the event of respondents editing answers
- Icon showing status of service and/or server
- Ability to invite respondents from within the tool's own interface
- Monitoring of respondents' progress in filling out the form
- Ability to communicate with respondents within the tool's own interface
- Flexibility in types of answers and file-uploads, system must not be too static
- Ability to change the banner displayed when filling out forms
- Ability to display a "business card" containing contact information on the page when filling out forms
- Ability to display different contact information on said "business cards" to different sets of respondents, to distribute questions or calls for support evenly
- Ability to invite respondents in batches, without having to manually keep track of who has been sent links and who have not
- Ability to view uncomplete dataset, i.e. view answers before all respondents are finished, within the tool's own interface
- Ability to automatically extract desired information from uploaded files where suitable, i.e. relatively uniformly formatted documents, like receipts
- Ability to rename user-uploaded files when downloading dataset, to something reflecting the uploaded file's "meaning" or content
- Ability to create "help"-buttons for individual fields in the form

Some of these desired features are relatively simple things that says more of the weaknesses of the Skjemaker system than about the particular needs of journalists using such a system to collect data. For example, there is no good reason why a modern online form filling system should not allow users to partially fill out their forms and resume or edit their answers later, in my anecdotal experience, most other form systems allow this. The ability to customize the look of the page presented to users filling out their forms is also a relatively simple thing that other systems allow, as well as designating extended help-texts (shown with a press of a help-button) for particular questions. The list of findings was later reduced to four main points that guided the development of the prototype. These are listed in the "Requirements" sub-section below.

Prototype

The produced artifact is a prototype journalistic web-form tool. It allows users to create forms with different types of fields, and making these forms available for filling out by respondents on the internet, using a link sent by email. Data collected via these forms are then available for inspection directly within the system's web interface, or can be downloaded. The prototype is intended to demonstrate ideas that I believe would make a more developed version an especially applicable form-tool for the journalistic context of use. This includes gathering advanced features that are otherwise frequently used by data journalists via other less user-friendly software, like OCR or graph visualizations, into a single centralized data collection and analysis system, as well as providing ways for journalists to easily gauge the state of their dataset at a glance. These advanced features are not implemented in the prototype directly, but are demonstrated via mockups. Figure 3 below shows a snippet of the "Komform[6]" admin page, the first view a user of the prototype system sees after logging in.



KomForm Admin Page

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

KOMFORM		
Forms	+ Add	✎ Change
Journalists	+ Add	✎ Change
Question fields	+ Add	✎ Change
Respondents	+ Add	✎ Change

Recent actions

My actions

✎ demoUser
User

+ demoUser
User

+ Deployment Demo Form
Form

+ Principality of Vet-Le
Respondent

+ Mikkel Rev
Journalist

Figure 3. This figure shows a snippet of the admin view from the prototype tool including a list of database objects the user can create and manipulate, as well as a summary of recent actions performed by the user.

[6] The name of the prototype changed several times over the course of development. The first name was "Foiform", where "foi" stood for "Freedom Of Information" as the primary use case of the tool was thought to be freedom of information requests for access or insight. This was later changed to "Komform" as the imagined use of the tool was narrowed further to data requests from municipalities, Norwegian: "Kommune", hence the "Kom" and "form" to make the word "conform" but with a k. Except, it was later realized, conform is not spelled "comform" so this name was also inadequate. The next name idea was "DCAF", being an acronym for "Data Collection and Analysis with Forms", alternatively with "Journalist's" or "Journalistic" appended in front to form "JDCAF". "DCAF" is however already the name of a Swiss think tank. The name of the application in the source code still remains "Komform" out of convenience.

Requirements

From the list of desired features identified during the initial expert interview, four main points were chosen to be the focus of development for the prototype. These were:

1. The ability for respondents to resume forms at different times, and edit previously given answers.
2. Handling respondents in a way that allows the journalist to keep track of who has been invited to the form so far (been sent a link by email), as well as progress respondents have made towards completion of their forms.
3. Handling of file uploads in a way that makes them easily accessible for the journalist, as well as offering some kind of light OCR information extraction from certain types of documents.
4. Presenting collected data to the journalist in a straightforward way within the system's own web-interface.

Other features identified in the initial expert interview were kept in mind during development and some were implemented when convenient. They were not explicit requirements, but still did their part shaping the prototype, or gave form to mockups of imagined designs. In no particular order, these secondary "requirements" were:

- The ability to view and/or download uploaded files in a configurable folder structure, and rename these files according to user's preferences
- The system should alert the journalist in the event of respondents editing answers
- The ability to communicate with respondents within the tool's own interface
- On form fill page, display a "business card" with contact information to appropriate journalist
- Ability to display different contact information on said "business cards" to different sets of respondents, to distribute questions or calls for support evenly
- Ability to invite respondents in batches, without having to manually keep track of who has been sent links and who have not

Development

The first major choice that had to be made before development could begin was whether to develop the prototype from scratch or to select an appropriate open-source web-form tool and use it as a base. The former allowing greater control of how everything works but necessitating a great deal of "trivial" groundwork, and the latter offering a presumably professional base architecture upon which to build desired features but at the expense of not understanding how the base architecture works. The open-source system that was considered as a candidate was "LimeSurvey[7]", offering a web-form system similar to the earlier discussed "Skjemaker" complete with an open code base that could be cloned and further built upon. After briefly inspecting LimeSurvey's source code it was decided that attempting to build upon it would be too big of an unknown variable, and the choice was made to instead build the prototype completely from scratch. Prior to developing the prototype I had no experience with building an application of this type or scale, and the desire to "learn by doing" was an influence on this choice.

[7] https://www.limesurvey.org/

Development of the prototype occurred during the months of October 2019 through March 2020. The beginning of development followed the initial exploratory expert interview, as soon as the conundrum described above had been resolved. Development of the prototype was planned to be carried out according to defining features of various Agile Development methodologies. The Kanban board for organizing and tracking tasks, and weekly or bi-weekly meetings with my supervisor between "sprints". Adherence to the planned development scheme was varied. Most of the month of October 2019 was spent going through tutorials and learning the basics of working with the chosen framework, culminating in a roughly functional tool offering base functionality at the end of the month. The following month of November 2019 saw more of the base functionality implemented and extended, before the project entered an early Christmas hiatus. Development resumed mid-January 2020 with much work being done quickly to improve the visual look of the various webpages, improve the security of the system, enable email functionality, and overall implement more of the desired core functionality. By the end of the month of January the prototype was in an almost completed state. February 2020 saw the last of the planned and emergent features implemented to a satisfying degree, and the prototype was judged ready for demonstration by the middle of the month. Development then entered a roughly one month break as work was put into other aspects of the project. Final development of the prototype occurred in March 2020, introducing more features that were previously overlooked, as well as important bugfixes. In the final week of the month, the prototype was adapted for deployment to a chosen platform-as-a-service provider in preparation for remote interviews. Successful deployment marked the end of development.

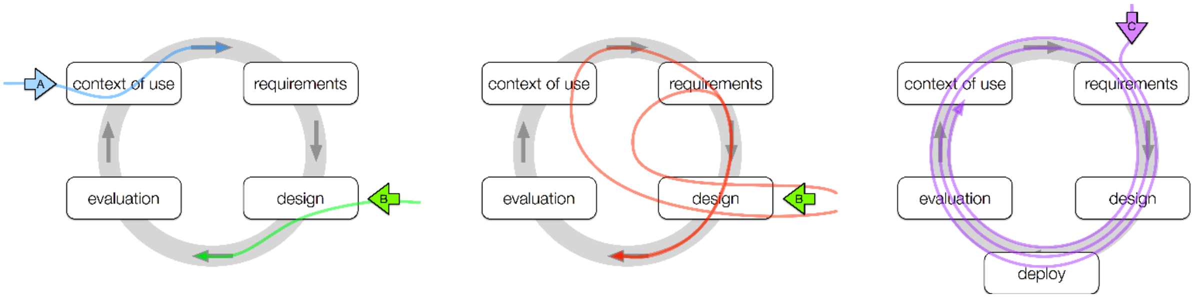


Figure 4. Figure of design and development processes adapted and extended by Brehmer et al. from Lloyd and Dykes. Figure from Brehmer et al. (2014). Figure shows traditional (green) and grounded (blue) approaches to design and development, as well as a special (red) approach in the middle where context of use and elicit requirements are established using example designs.

The prototype's development process can be likened to Brehmer et al.'s (2014) "special" design and development approach seen in Figure 4 above, in that it began with a design (albeit not my own) in "Skjemaker" which was examined in a context of use through an interview with an expert user, which simultaneously elicited requirements that led to a new design. This design would later be evaluated, producing new requirements that could have led to an improved design had the project been continued beyond the scope of this thesis. If we imagine a later, more mature iteration of the prototype, a suitable form of evaluation metric could have been "adoption" as described by Brehmer et al. (2014) in their article on Overview, meaning "repeated, voluntary use in real projects". In this case, a successful design would be one that is adopted into regular use by data journalists outside of the context of explicit evaluation.

Design Notes

This section contains notes on technical aspects of the prototype; the choice of frameworks, packages, and services and the motivations behind these choices, and notes on the prototype's features; a description and discussion of the choice of features that were implemented.

Technical Notes

The Django Framework

The prototype source code is written using the *Python* programming language, a choice not motivated by any technical aspect. For creating web-applications in Python there are several frameworks available that provide useful building blocks, allowing this non-trivial task to be done relatively easily. The prototype system uses the Python-to-web framework Django[8]. Django was chosen because it is fast, secure, scalable, and most importantly because it was "invented to meet fast-moving newsroom deadlines", according to the Django website. The prototype is a "Django application", as the system is built entirely within the architecture the framework provides. This architecture includes the entirety of the backend web server and security handling, and an easy-to-use configuration script that allows various other services to be connected via appropriate bindings, like a database or email service. The architecture used by Django applications also includes defined ways of creating the web pages the system will use, and allows developers to define their database objects as Python classes, called "models" by Django, which allows easy interaction with database objects within the python code using Django functions. The prototype system has models representing forms, questions and answers of various types, respondents, and journalists. Without creating any webpages at all, the system's models can be viewed, edited, or created via a built-in web-interface, the "Django admin view". The prototype uses a slightly modified version of the Django admin view as its main interface, that is, as the frontend that the journalist would use (see Figure 3), providing access to the models that have been registered in the code to be editable via the admin view. This provides a professional-looking interface allowing access to the systems core features very quickly and easily. The tradeoff to this convenience is the inconvenience of implementing other features not tied directly to the admin views intended use (interacting with database objects). Except for being built-in and coming with login and security features pre-configured, the admin view functions like any other "view", which is Django's name for the Python functions it uses to render desired webpages. As such the admin view is completely customizable, but only provided a level of "Django know-how" it was outside the scope of the prototype development to acquire. Therefore, using the built-in interface made the implementation of some desired features problematic, resulting in less than optimal workarounds.

Other Noteworthy Packages and Services

The prototype uses *PostgreSQL*[9] for persistent data storage. Django ships with a built-in SQLite database solution for testing and simple development. This was used early during development, but PostgreSQL was chosen later to be the production database. This choice was made out of convenience as Django recommends using PostgreSQL with Django applications and comes with good instructions on how to set the two up to interact nicely with each other.

Some of the prototype's features involve sending emails. Django provides easy-to-use functions that allow this to be done from Python code, provided the necessary bindings have been set up to an email

[8] https://www.djangoproject.com/
[9] https://www.postgresql.org/

service provider. Here the prototype system uses *SendGrid*[10]. SendGrid's free tier allows users to send up to 100 emails per day, which is more than enough for demonstration purposes, and is relatively easy to set up.

Due to the Covid-19 pandemic it became necessary to deploy the prototype to an online platform. Deployment was originally considered to be an optional step in development, as the interviews where the prototype would be demonstrated were planned to be held in person and demonstration of the prototype would happen using a separate computer with a local installation. When deploying a Django application, one of the most straightforward solutions is to use the Platform-as-a-Service provider *Heroku*[11]. The Django documentation provides an extensive tutorial on how to deploy Django apps using Heroku, with relatively easy to follow documentation provided from Heroku's side as well. The choice of Heroku as a PaaS provider was motivated by a combination of the access to these helpful tutorials, and their free "hobby" tier which provided satisfactory performance free of charge.

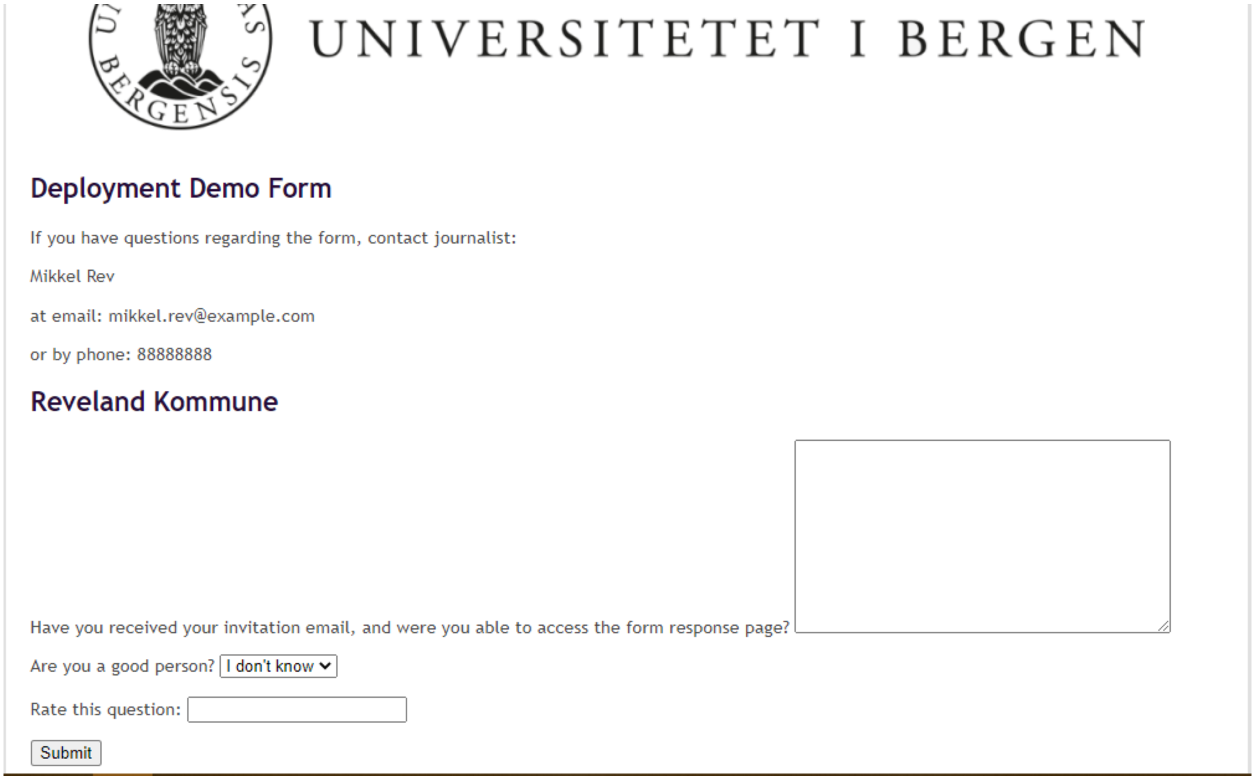
Security

Security was not a primary concern when planning the development of the prototype as the purpose of the system was only to be a relatively simple piece of demonstration software. It was felt that an explicit focus on data security would be a little over the top, and a wasted effort. However, during development it became apparent that some level of security had to be implemented to restrict access to the system's webpages, especially considering the emergent need to deploy the prototype to an online platform. Fortunately, the most pressing concern, securely logging in to the journalist-facing system, was already taken care of by the Django framework and its admin interface. The Django admin interface allows creation of new users (in our case that would be journalists' accounts), assigning privileges, and so on. Passwords in the Django "Users"-system are stored only as hashed values in the database and cannot be retrieved.

Every webpage the prototype system provides that is normally accessed via the admin interface requires a valid login, and cannot be accessed via direct link otherwise. This is not the case for the webpages serving the "respondent-facing" side of the system, that is, the pages not intended to be used by the journalist. In the prototype these pages only include the page for filling a given form (see Figure 5 below) as well as a simple "thank you" page where respondents are redirected after submitting their data. The "thanks"-page is currently unprotected as it is completely static and offers no way to interact with the system. The link to a respondent's view of a form (their invitation link) consists of multiple hashed values using a signer function that is "salted" with parts of the system's build path, making it effectively impossible to guess the link even if the system's secret key has been compromised. Anyone with a correct link may submit data to a form (each link leads to a given form/respondent pair) as no login or validation is required. This approach is convenient for developers and respondents alike, given that the link is safe in the recipient's email inbox and is not shared with others.

[10] https://sendgrid.com/

[11] https://www.heroku.com/



Deployment Demo Form

If you have questions regarding the form, contact journalist:

Mikkel Rev

at email: mikkel.rev@example.com

or by phone: 88888888

Reveland Kommune

Have you received your invitation email, and were you able to access the form response page?

Are you a good person? I don't know

Rate this question:

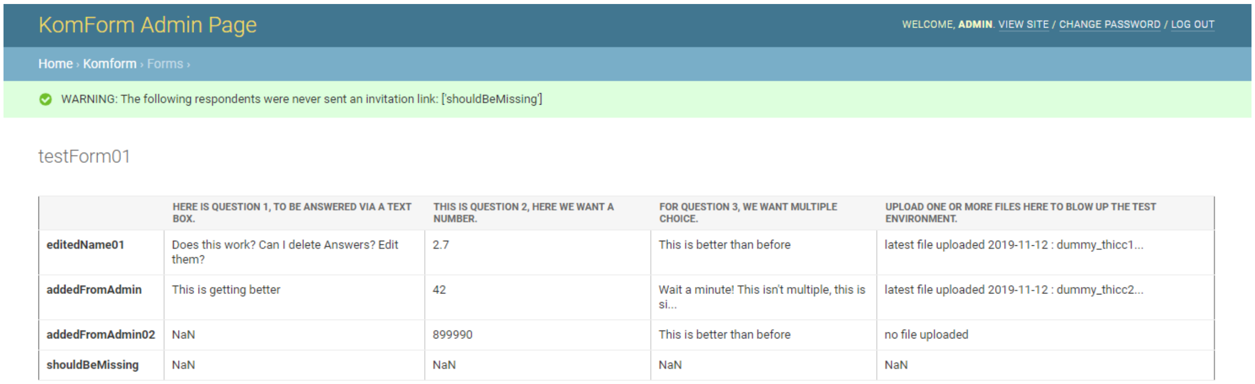
Figure 5. This figure shows an example of a respondent's view when filling out a form in the current prototype.

Features

Fulfilling the Requirements

Whether the chosen list of four major focus points comprised the four most interesting features identified during the initial expert interview is debatable. The ability for respondents to resume filling out forms or to return later to edit answers is not in any way a novel feature for a web form tool. The reason this feature was chosen as a requirement was exactly because it is an obvious requirement, one that the "benchmark" web form tool (Skjemaker) the prototype was supposed to improve upon did not meet. However, when developing a system like this prototype, the list of requirements is often synonymous to the "list of major features", and while point number one is definitely a requirement it is not a particularly exciting or novel feature.

As for the remaining three main requirements of development, points two (tracking respondents' progress and invitations) and four (straightforward overview of data within system interface) were implemented partially via a spreadsheet view of the data collected via a given form, available within the system's web interface. This view allows the journalist to track respondents' progress by directly inspecting which fields in the table are empty, as shown in Figure 6 below. Additionally, a warning label is displayed with the names of respondents tied to the given form who have not yet been sent an invitation email, if such exist. The table view also provides a straightforward way to inspect collected data without having to download the full set. These features were implemented in a rudimentary fashion, suitable for demonstration only, with proposed designs for a further development presented in mockups during the interviews.



WARNING: The following respondents were never sent an invitation link: ['shouldBeMissing']

testForm01

	HERE IS QUESTION 1, TO BE ANSWERED VIA A TEXT BOX.	THIS IS QUESTION 2, HERE WE WANT A NUMBER.	FOR QUESTION 3, WE WANT MULTIPLE CHOICE.	UPLOAD ONE OR MORE FILES HERE TO BLOW UP THE TEST ENVIRONMENT.
editedName01	Does this work? Can I delete Answers? Edit them?	2.7	This is better than before	latest file uploaded 2019-11-12 : dummy_thicc1...
addedFromAdmin	This is getting better	42	Wait a minute! This isn't multiple, this is si...	latest file uploaded 2019-11-12 : dummy_thicc2...
addedFromAdmin02	NaN	899990	This is better than before	no file uploaded
shouldBeMissing	NaN	NaN	NaN	NaN

Figure 6. As can be seen in the bottom row of the table, the bottom respondent has not answered any of the questions in the form, as all of their cells have the value "NaN" ("Not A Number", which is the default value for missing entries in a Pandas dataframe - the data object used to create the table). It can also be seen that the second-to-last respondent has not provided input to the first question. Looking near the top of the figure, it becomes apparent that the reason the bottom respondent has given no answers is because they were never issued an invitation to the survey.

Point three from the aforementioned list, about file uploads and OCR, was omitted towards the end of development, due to time constraints and the uncertainty of the imagined prototype design for the OCR feature. The idea was to offer extraction of "key elements" using OCR, from defined, relatively uniform documents, like receipts (where the "key element" could be the sum paid). As OCR can be unreliable with low-quality or otherwise messy documents, it was thought that restricting the offered information extraction to only a few "simple" cases would be the best course of action. But this presented new problems, how would the system know that the uploaded document is of a type suitable for extracting "key elements"? What exactly would the key element be? And is this even really a realistically useful feature? It was decided that instead of implementing this feature in the prototype, it would instead be proposed to journalists during the interviews in a mockup showing the imagined design. Instead of potentially wasting time implementing a feature of dubious realism and usefulness, time could be spent developing other desired features. These were features that emerged at different stages of prototype development and were therefore not part of the initial plan, or they were features identified after the initial expert interview that were not given main priority. These features are discussed below.

Notes on Secondary Features

The ability to view and download uploaded files in configurable folder structures (and renaming the files according to some configurable specification) was explored in a mockup, but this mockup was omitted from the interviews as it became apparent that this feature was of little importance. With the design philosophy of the prototype being to allow as much work as possible to be done from within the system without requiring downloads, it seems contradictory to focus on offering download features intended to make it more convenient to work with local scripts on downloaded files.

The prototype does display contact information for the journalist registered in the system as the forms "owner" at the form fill page presented to respondents, as can be seen in Figure 5. This does however only link to the first entry on the list of journalists associated with the given form, and does not distribute their contact information across respondents to distribute the load of support calls and questions. The feature was considered easy enough to demonstrate using only a simple implementation. Related to this feature was the desired ability to invite respondents in batches, again to distribute calls for support, but across time instead of across journalists. The prototype allows this, but not automatically. Respondents may be sent invitation links by email from within the web interface, but the journalist must manually select which. The system assists here by listing those respondents *not* invited to a given form where at least one other respondent has been.

When it comes to warning the journalist in the event that a respondent edits their answers, one simple way to do this was implemented in the prototype. The way this is done is that the system detects when a new answer is submitted where a previously recorded answer for the given respondent/question-pair exists, and in this event, an email is sent to the journalist registered as the respective form's "owner". This approach is unpractical in that it may lead to cumbersome amounts of incoming emails in cases where the number of respondents is high, although this problem would be lessened if several journalists share ownership of a form allowing emails to be distributed between them (however, this might again introduce problems keeping track of information across different mailboxes). A more desirable approach would be to alert the journalist(s) of respondent edits via notifications within the system interface. This approach could likely also pave the way for a chat feature allowing respondents and journalists to communicate within the system (notifications can be considered messages from the system, and so could be built using the same components as a chat feature). Communicating this way minimizes the risk of information being lost to the ether or miscommunicated between journalists collaborating on a project, as well as providing an accurate log of correspondence. Neither chat system nor in-interface notifications were explored in the prototype as these features would require extensively customizing the built-in Django admin interface. This task was judged too time-consuming to be worth the implementation of features that could easily be demonstrated and discussed in simpler ways.

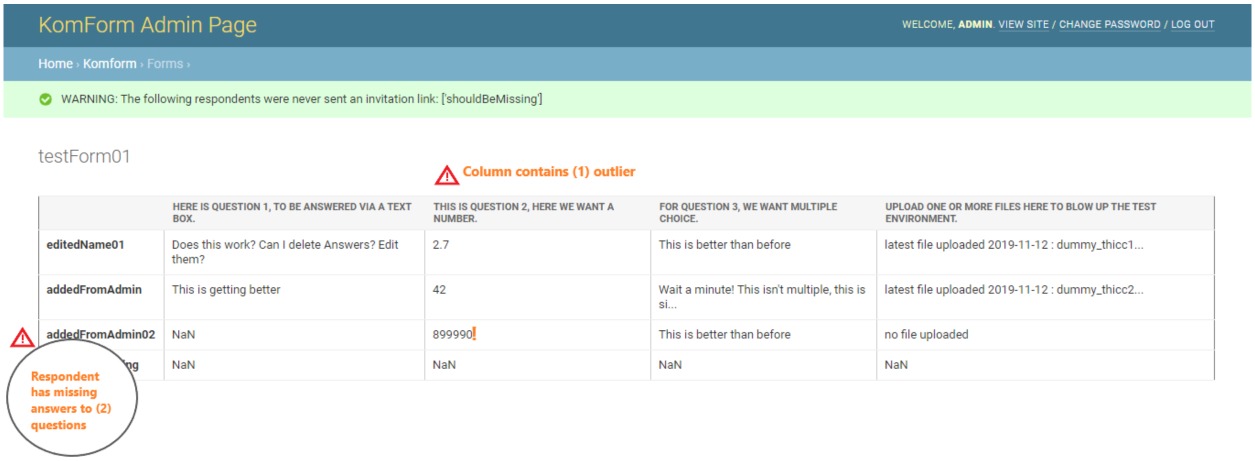
Semantic Graph

The prototype uses a Python package called *RDFLib* to create an RDF graph containing triples about forms, respondents, questions, and answers (journalists not included because they did not exist in the database when the RDF script was written) organized in a "naïve" ontology where only these four classes exist and their members are connected to each other via a network of object properties and their inverses. The graph is stored in a turtle-file inside the system's build environment. The semantic graph is included in the prototype only as an experimental demonstration feature and is not available via the interface. The script creating the triples and writing them to the file is ran when a user presses the "view data"-button for a given form. When this happens, all the relevant data for the given form is parsed to create triples which are then written to the turtle-file, adding to it (duplicate triples are not an issue). The first time this is done, the script does not add to a pre-existing turtle-file, instead creating one from a "base graph" file containing triples defining the classes and properties from the ontology. The feature was added late in the development process and was therefore left in its barebone state due to lack of time. Given a little more work, the RDF graph could be used to offer interactive graph visualization of collected data within the interface. A SPARQL query endpoint would also be desirable to make it possible to fully leverage the power of RDF. An example of how these features might have looked is presented in Figure 10 in the section describing the mockups below.

Use as Technology Probe

The protype was used as a technology probe during the interviews to stimulate discussion and exchange of ideas, as well as to gather feedback on the prototype itself. The purpose of the interviews was as such twofold, the purpose other than evaluation of the artifact was to explore the interviewed journalists' disposition towards the general idea of a dedicated data journalism tool. In this regard the prototype provided a suitable backdrop for demonstrating imagined features that were not implemented directly due to time constraints or because they were highly non-trivial. These features were demonstrated by describing their imagined use while showing the interviewee mockups in the form of screenshots of various views from the prototype with the imagined features edited in. The mockups used during the interviews (four in total) are shown in figures and described below.

Highlighting Anomalies in Table View



WARNING: The following respondents were never sent an invitation link: ['shouldBeMissing']

testForm01

⚠ Column contains (1) outlier

	HERE IS QUESTION 1, TO BE ANSWERED VIA A TEXT BOX.	THIS IS QUESTION 2, HERE WE WANT A NUMBER.	FOR QUESTION 3, WE WANT MULTIPLE CHOICE.	UPLOAD ONE OR MORE FILES HERE TO BLOW UP THE TEST ENVIRONMENT.
editedName01	Does this work? Can I delete Answers? Edit them?	2.7	This is better than before	latest file uploaded 2019-11-12 : dummy_thicc1...
addedFromAdmin	This is getting better	42	Wait a minute! This isn't multiple, this is si...	latest file uploaded 2019-11-12 : dummy_thicc2...
⚠ **addedFromAdmin02**	NaN	899990!	This is better than before	no file uploaded
ng	NaN	NaN	NaN	NaN

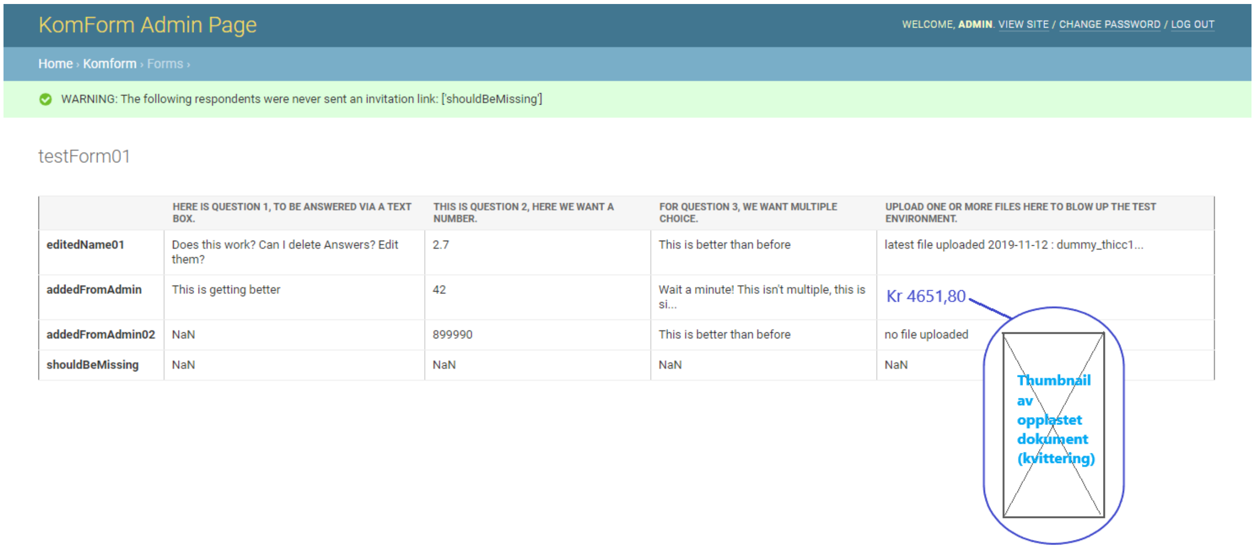
Respondent has missing answers to (2) questions

Figure 7. Mockup of an anomaly highlighting feature when viewing data in a table.

The idea demonstrated by the mockup shown above in Figure 7 is an anomaly highlighting feature intended to give journalists an idea of the state of the dataset at a quick glance, all without downloading anything or leaving the system's web interface. The imagined feature would serve both to immediately direct the journalist's attention to potentially interesting data, and to provide an overview of respondents' progress in submitting data. The warning on the left-hand side of the table could be changed to a small progress bar or circle showing how many answers the given respondent has provided out of the total questions. Given different specific applications of the system, for example a survey vs. an inquiry requesting officials to upload sets of documents, progress meters could represent different metrics, like number of, or size of, uploaded files (relative to some desired value).

While the idea of highlighting outliers and abnormal values is simple, anomaly detection is generally highly non-trivial. In some cases one can get away with using simple approaches like defining anomalies as being datapoints that differ from various statistical properties of a distribution (Oracle 2017). Use cases where simple statistics are enough includes those where the definitions of normal and abnormal behavior are static over time, and the boundary between the two is clear and precise. These conditions narrowly restrict the applicability of the simple statistical approaches, they would only deliver satisfying, trustworthy results in a few select cases. As noted by Chandola et al. (2009) in their article *Anomaly Detection: A Survey*, it is very difficult, perhaps impossible, to define a normal region that encompasses every possible normal behavior. In many fields, the notion of normal behavior also evolves over time, and the notions of normalcy between different fields also often differ. A proposed anomaly detection and highlighting feature intended to work with a variety of data from different sources would benefit from using more advanced detection methods. Such methods are described further in Chandola et al.'s article, and are often based on some form of machine learning. These techniques will not be described in detail here. It would suffice to say that implementing the proposed anomaly detection feature would represent a significant investment of development resources, very much warranting a closer investigation of its potential value to journalists first.

Automatically Extracting Key Values from Uploaded Documents Using OCR



KomForm Admin Page

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Komform › Forms ›

WARNING: The following respondents were never sent an invitation link: ['shouldBeMissing']

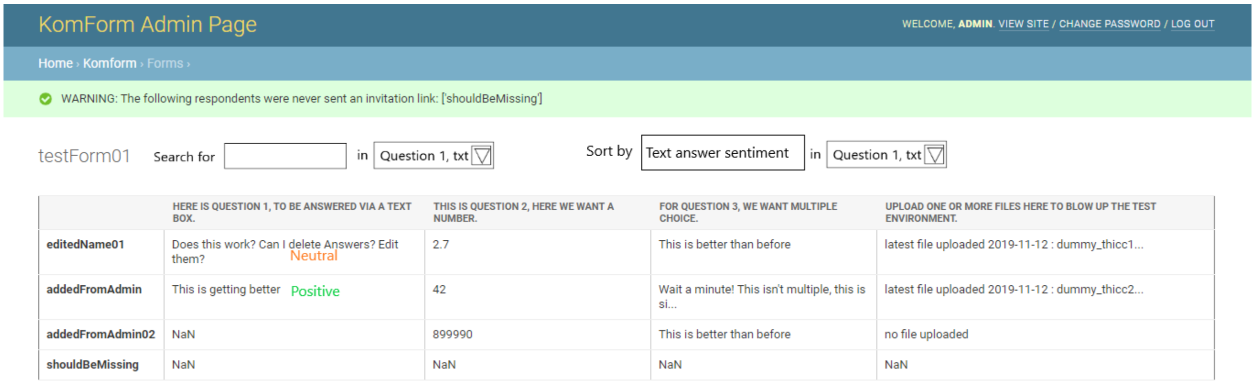
testForm01

	HERE IS QUESTION 1, TO BE ANSWERED VIA A TEXT BOX.	THIS IS QUESTION 2, HERE WE WANT A NUMBER.	FOR QUESTION 3, WE WANT MULTIPLE CHOICE.	UPLOAD ONE OR MORE FILES HERE TO BLOW UP THE TEST ENVIRONMENT.
editedName01	Does this work? Can I delete Answers? Edit them?	2.7	This is better than before	latest file uploaded 2019-11-12 : dummy_thicc1...
addedFromAdmin	This is getting better	42	Wait a minute! This isn't multiple, this is si...	Kr 4651,80
addedFromAdmin02	NaN	899990	This is better than before	no file uploaded
shouldBeMissing	NaN	NaN	NaN	NaN

Figure 8. Mockup showing how information extracted by Optical Character Recognition could be presented to the user.

Figure 8 shows the mockup that was used to demonstrate the imagined Optical Character Recognition feature that was originally planned to be fully or partially implemented in the prototype. As previously described under the "*Fulfilling the Requirements*" header, this feature was instead shown as a mockup as it became clear that its imagined use case might be unrealistic. The idea shown in this particular application of OCR on a dataset is that the system could automatically extract some "key value" from an uploaded file, and display that value directly in the spreadsheet with the actual uploaded file available as a thumbnail when hovering over the extracted value. In the mockup example, the document type is a receipt, and so the imagined system would know to look for a final sum value. Regardless of whether this particular use of OCR is realistically useful (there are usually more interesting things to discover in a document than one single value), the mockup above also served the purpose of steering the conversation during the interviews to the general topic of OCR and its applications in data journalism, particularly pertaining to its integration into a comprehensive data collection and analysis system. This idea is more general, simply theorizing that it would be convenient and time-saving for data journalists to be able to access OCR features directly from within a system like the prototype. An example use case could be making pdfs searchable and storing them as text documents in the same database, without requiring the journalist to download the files and expose them to their own locally installed OCR solution using console commands before re-uploading the results.

Advanced Sort and Search Features



KomForm Admin Page

WARNING: The following respondents were never sent an invitation link: ['shouldBeMissing']

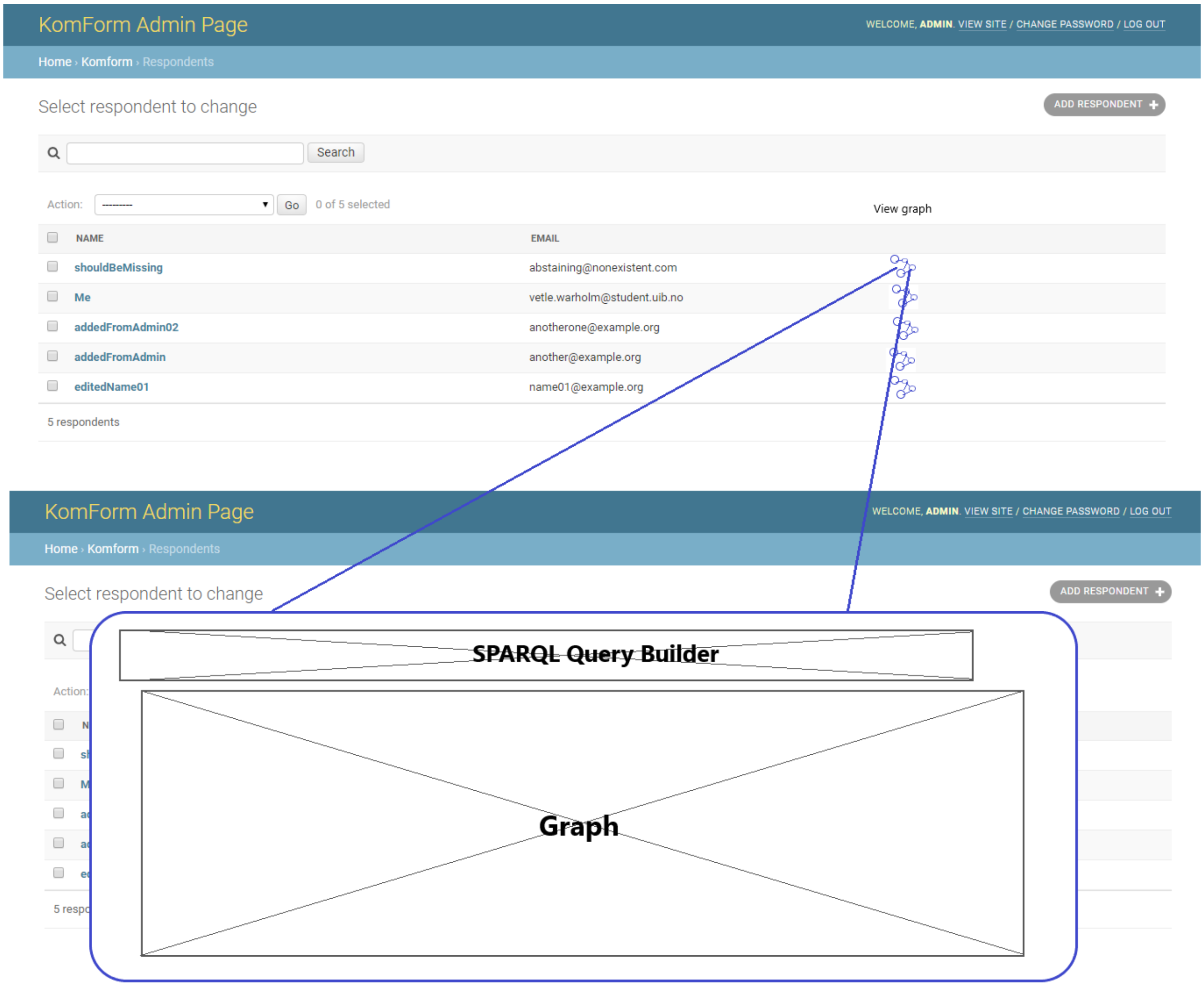
testForm01 Search for [ ] in Question 1, txt ▽ Sort by Text answer sentiment in Question 1, txt ▽

	HERE IS QUESTION 1, TO BE ANSWERED VIA A TEXT BOX.	THIS IS QUESTION 2, HERE WE WANT A NUMBER.	FOR QUESTION 3, WE WANT MULTIPLE CHOICE.	UPLOAD ONE OR MORE FILES HERE TO BLOW UP THE TEST ENVIRONMENT.
editedName01	Does this work? Can I delete Answers? Edit them? Neutral	2.7	This is better than before	latest file uploaded 2019-11-12 : dummy_thicc1...
addedFromAdmin	This is getting better Positive	42	Wait a minute! This isn't multiple, this is si...	latest file uploaded 2019-11-12 : dummy_thicc2...
addedFromAdmin02	NaN	899990	This is better than before	no file uploaded
shouldBeMissing	NaN	NaN	NaN	NaN

Figure 9. This mockup presents the idea of allowing searching and sorting by advanced metrics like text sentiment when viewing data in table format.

The mockup presented in Figure 9 shows an imagined way to search and sort data in a table view using various metrics. As the figure above shows, data can be filtered by searching for values or key words in answers to specific questions, or be sorted by more advanced metrics than are usually available in most spreadsheet programs. One such metric could be "sentiment" for text values, allowing rows in the table to be sorted by the negative/positive score of the answer to a given question. Other examples are various statistical metrics for numerical values, the likes of which a data journalist might otherwise turn to R to compute. The proposed sorting and searching feature was imagined to be of use in cases where the data being investigated are constructed from "raw" text or numeric responses to questions, instead of from uploaded files. This use case is more akin to a traditional survey. It is not uncommon for general-purpose survey tools to offer built-in support for generating simple statistics about collected data. The features presented above could be thought of as a step towards extending this kind of statistics-generating functionality to be more customizable and helping the journalist explore their data better.

Graph View and Graphic SPARQL Query Builder



KomForm Admin Page

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Komform › Respondents

Select respondent to change

Action: --------- Go 0 of 5 selected

□	NAME	EMAIL	View graph
□	shouldBeMissing	abstaining@nonexistent.com	
□	Me	vetle.warholm@student.uib.no	
□	addedFromAdmin02	anotherone@example.org	
□	addedFromAdmin	another@example.org	
□	editedName01	name01@example.org	

5 respondents

ADD RESPONDENT +
Search
KomForm Admin Page
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Komform › Respondents
Select respondent to change
ADD RESPONDENT +
SPARQL Query Builder
Graph

Figure 10. A mockup showing an imagined combined SPARQL query builder and graph view feature. When pressing the button to "View Graph" next to any object in the system, the view above would pop up in front and show the objects position in the semantic graph, as well as allowing the user to build SPARQL queries using a graphic query builder.

Figure 10 shows a mockup of a proposed combined semantic graph inspection and visual SPARQL query builder feature, extending across two "frames". The upper part of the image shows a view from the system containing a list of some database objects, where all have a small "view graph" icon next to them. With the proposed design, clicking the icon would produce a pop-up window in the foreground, transitioning us to the lower part of the figure above. This part of the mockup simply shows that the proposed pop-up window would be a part of the system interface in a way that would not produce an actual new window in the user's operating system, but would place itself in the foreground of the view the user was already in. The foreground window would contain a combined view of the semantic graph the selected object is part of, as well as a SPARQL query builder. The graph might be configured to display only the most immediate connections by default, or some other reduced form of the graph, to avoid cluttering the view. The SPARQL query builder would utilize a graphic interface to allow construction of a query in a way that is accessible to users not intimately familiar with SPARQL. The query builder could utilize the "filter/flow" design as proposed by Haag et al. (2014) where a query is constructed by adding nodes to a directed graph where each node represents a filter that data must pass through. Haag et al.'s design allow queries to be constructed step by step by users with no knowledge about SPARQL as the actual query text is never exposed to them. The design does however assume that the user understands the concept of a directed graph, which is perhaps reasonable, as most journalists should know how flow-charts work. Query results would be displayed either as a traditional list, either on top of the graph or in a separate window, or in the case of only a single returned object, it could shift the focus of the graph to that object.

Interactive graph visualization offers an immersive way to explore data and makes it easier to capture connections to other entities that might otherwise be hidden among other facts. This feature is interesting enough in itself to perhaps warrant its own window without sharing with the SPARQL query builder. Another point in favor of splitting the features into separate windows is that a visual query builder might occupy a significant section of the screen, depending on the chosen technique and the complexity of the query under construction. On the other hand, grouping features related to semantic technologies together in a single view might be practical if a discovery made by exploring the graph prompts a desire to know more by executing a SPARQL query. The features are grouped together in a single view in the mockup for convenience.

Interviews with Prototype – Main Case

The primary investigation carried out during this thesis project was a series of interviews with Norwegian data journalists in Bergen. The interviews featured the previously described prototype as a technology probe, as well as a number of questions about the journalists' previous experiences working as a data journalist. The goal of this investigation was to provide further evidence to answer the project's research questions one way or the other, where the initial exploratory case could only suggest.

Semi-Structured Interviews with Experts

The interviews and their planning, design, and adaptations to the Covid-19 pandemic regulations are covered in more detail in the "Methods" chapter, but are briefly recapped here. The interviews were semi-structured to allow the interviewed journalists to talk freely about topics they were enthusiastic about, after these topics were introduced via questions from the interview guide. Each interview began with questions about the interviewee's work as a data journalist before moving on to the prototype and discussions of its features and those features shown in the mockups. The questions about the interviewee and their work were motivated both by justifying the interviewee's status as an "expert" and to discover whether data journalists with differing backgrounds have different opinions on the proposed designs for a data journalism tool. The purpose of the prototype in the interviews was to give a high-fidelity interactive demonstration of what a proposed purpose-made data journalism tool might look like, specifically to the purpose of eliminating unnecessary steps when collecting data with online forms. Hopefully, this would not just trigger a reaction from the interviewee pertaining purely to the current prototype system, but instead lead to a response towards the general idea of a purpose-made data journalism system, further leading to a discussion of what such a system would ideally look like to meet their organization's needs.

A list of ten names of journalists matching the wanted description was provided by a contact within SUJO. These were introduced to the project in a separate primer-email distributed by the SUJO contact, before being contacted directly by me about participating in an interview. Several of the contacted journalists initially responded with enthusiasm and interest, but were later unable to participate. The interviews were planned to be conducted in the space of the three weeks, beginning the week before Easter of 2020 and ending the week after. Another two weeks were added to the timeframe to accommodate for the unexpected work-situation many potential interviewees found themselves in, bringing the interview-phase of the research project up to include the entire month of April. Despite the extended timeframe, only three interviews were able to be held before the interview-phase had to be concluded. Being a qualitative investigation, three interviewees were considered sufficient to provide insightful data, but were fewer than was desired.

In adaptation to the work-from-home orders during the Covid-19 pandemic, the interviews were held remotely using an online meeting provider. This medium eased the recording of the interviews, providing both audio and screen-capture recordings at the press of a button. Unfortunately, during one of the interviews, this button was forgotten, and the interview was instead recorded by immediately constructing a narrative of the interview after its conclusion, based on notes and fresh memory. Digital notes were taken during all interviews, complementing the recordings. The interviews were not attended by participants other than the individual journalists and myself. The interviews were held in Norwegian, and any quotes in the text by the interviewees are my own translations.

Analysis

The interview data were analyzed using a lightweight encoding scheme where units of text were attributed to nine identified themes. Some themes were identified prior to the analysis, and some emerged during the encoding itself. The interview recordings and notes were used to write an aggregated document containing the narratives from all three interviews. That text was then highlighted with different colors to show the individual units of text that closely related to specific themes (the aggregated text, including color highlights, is included in the appendix). The themes are presented in Table 1 below. Some themes are by their description quite vague, like "prototype criticism" and "prototype praise". These are not very informative by themselves, but helped structure and organize the analysis. Others better present an indication of the overall narrative of the interviews, suggesting the outline of the findings.

Theme 1	Uses advanced contemporary tools, has no problem working with third-party programs
Theme 2	Data sources mostly include public databases and Freedom of Information access requests
Theme 3	Official data (open database or FOI) are often more difficult to use than anticipated
Theme 4	Sometimes, simpler solutions are better, new systems may be unwanted
Theme 5	Documentation is an important part of good data journalism
Theme 6	Need to accommodate for less technologically competent/experienced journalists
Theme 7	Prototype criticism, against proposed designs
Theme 8	Suggesting improvements to proposed designs, or completely new features
Theme 9	Prototype praise, positive towards proposed designs

Table 1. Presented is the table that was used during data analysis, containing the identified themes. Their order is only out of practicality, and does not represent an order of importance. In the original analysis table, colors were used instead of numbers, and the themes were ordered differently.

Interview Findings

Below, relevant data from the expert interviews are described, grouped by the themes from Table 1 above. Findings related to themes 2 and 3 have been grouped together for relevance and practicality. Themes 7, 8, and 9 have also been grouped together as they all relate to the prototype and its proposed designs and are therefore highly interconnected.

Interviewee Backgrounds and Their Current Toolbox

Out of the three participating journalists, there were two men and one woman. Interviewee 1 and 2 (I1 and I2) work actively as data journalists, at different major news organizations. Interviewee 3 (I3) has an organizational role, working to ensure quality and rigor of analysis among other data journalists within their organization. I1 and I2 both have experience with general "day to day" data journalism as well as deep investigative projects. In their work they report using tools such as the statistics programming language R, the Python notebook Jupyter Notebook, and general spreadsheet programs and database solutions to handle and analyze data.

I3 works as a "News Developer" ("Nyhetsutvikler"), being a leader for a team of "journalists who know programming", in I3's own words. I3's team is multi-disciplinary, having a collection of members with varied backgrounds from subfields of journalism or the data and information sciences. I3's team works wide, executing data journalism projects of their own, and rendering assistance to other journalists within the organization who have need of data journalism expertise. Within their team, I3's role is both to be an overall leader and to assist in quality assurance, running through a project step by step with the relevant data journalists to double check calculations and analysis and make sure everything is watertight and by the book. This quality assurance-work is often done in meetings with more team members where they also discuss ways to best present the data and project results to their readers.

Over the course of the interviews, all three interviewees mention computer tools, or make comments, suggesting that data journalists, being highly technologically competent people, have little or no issues working with multiple different third-party tools with varying degrees of user-friendliness to get their jobs done. Both I1 and I2 talk about statistics, I1 mentioning "the summary function in R" as an example of a "simple" way to offer sorting and searching metrics for data in the prototype interface's table view, and I2 saying all prospective data journalists should learn some statistics to avoid making mistakes calculating numbers whose significance they don't understand. I2 and I3 mention using different third-party applications to work with graph data, mapping relationships and owner-structures between entities and producing visuals.

Challenges Introducing Comprehensive New Systems

I1 says that in the event they need to collect data via a survey, they feel Google Forms gets the job done perfectly fine. I1 also prefers using "Google Spreadsheet" instead of Microsoft's Excel, as they feel the latter is "too cumbersome". If introducing a new data journalism system to a news organization, the system should not necessarily be made to do everything in a single package, I1 suggests. At their news organization, any new system or plugin used to produce content needs to be pass through the IT-department to be approved for use with the organization's Content Management System, a process that is not trivial for a comprehensive system. Furthermore, I1 says about their own organization that it is "a many-headed troll" and that it "is so large that it seems impossible to establish a single system for any specific task". With many different teams and units within the organization approaching similar challenges from many different angles, applying standardized solutions would be bound to upset many people. On this topic, I1 is positive towards the idea behind the prototype, and suggests it would be useful for a collaborating team, if not as a standardized system for data journalism across the entire organization. They express interest in seeing and testing a future more developed iteration of the prototype system.

Data Sources and Challenges with Public Data

I1 and I2 mention SSB (the Norwegian Bureau of Statistics) and public data accessed via Freedom of Information requests as commonly used sources of data. I1 also mentions freely available government data, while complaining that regional authorities like municipalities are "each on their own system" making it difficult to access, search for, or cross-reference data from these sources. Another data source mentioned by I2 is Geonorge. They stress that knowing where data are to be found is an important part of the job, and lament that public officials are often inept at exporting data from their own systems. They are joined by I3 saying their team often receives data from FOI requests in "unreadable formats". I3 further states that Freedom of Information access requests are "an area of expertise in and of itself" and that there is no journalistic standard for this kind of work. They say that it is up to each individual journalist to structure their data and document their process.

Importance of Documentation and Logging

I1 defines a data journalist as a journalist with the ability to find, structure, and exploit data (of the digital kind) to create news content, "structuring" being mentioned on the same level as "finding" and "using" data. I1 speaks explicitly about documentation and logging when talking about the prototype and suggesting changes to its design, including "versioning" being introduced to the data storage, allowing journalists to inspect different versions of the dataset tracking its evolution over time as the project develops.

I2 has an added role within their organization to train new data journalists, and when discussing some of the most important "rules" they teach, says that a very important point is to always keep a copy of the raw data as a part of documentation and logging. They say their organization takes documentation of the data collection and analysis process very seriously, and each step taken should be logged.

I3 offers the most comments on logging and documentation throughout their interview, the topic being central to their work. I3 regularly meets with their team members to ensure quality of analysis and documentation. Ensuring quality of analysis is not only about making sure that the calculations are correct. I2 points out in their interview that a calculation may be perfectly correct, but may mean something else entirely than what the journalist thinks it does, hence it is important to be able to track which calculations, data transformations, and other steps have been taken. I3 speaks of structuring and organizing the data itself as a necessity for being able to do this. Making sure that these affairs are in order is a topic that permeates I3's interview. Particularly pertaining to "regular" or non-data journalists attempting to work with data of their own. When asked whether they could think of any desired "better computer tools" at their organization at the top of their head, I3 responds saying that anything that would help ensure better documentation and logging would be very beneficial.

Data Journalism for Non-Data Journalists

I3 says that the data journalists in their unit works on data journalism projects of their own, as well as assisting other journalists who lack the technological skills to handle data on their own. It is apparently not uncommon that non-data journalists wish to work with data somehow, for example requesting access to particular public documents related to their area of expertise, essentially performing "lightweight" data journalism. According to I3, most of these journalists are generally very reluctant to structure and log their data work, saying that "they struggle with it [structuring]" and "this is about [lack of] data competence". Another common problem for non-data journalists working with digital data is "not standing a chance" applying techniques like OCR on their own. Usage of tools with "less than user-friendly" interfaces generally requires assistance from a data journalist. Near the end of their interview, I3 was asked whether they thought challenges with less technologically skilled colleagues would disappear as a younger generation of journalists enter the scene, people who have grown up using computers all the time. To this, I3 responded "absolutely not; technological competency has little to do with age. Fresh young students of journalism who have not taken any computer courses are not better at excel [for example] than older journalists". Using a computer is not the same as working with data.

I1 and I2 both comment here and there during their interviews that certain things must not be too difficult to use, or that not all journalists are equally "professional" at using advanced computer tools. Primarily working with data journalism hands-on, as opposed to having an organizational role, I1 and I2 do not offer perspectives on this subject as comprehensive as I3.

Prototype Critique and Suggested Changes

I1 was the interviewee that showed the most enthusiasm in exploring the prototype on their own, having at their own initiative already accessed the system using their test credentials and created a test survey prior to the interview. I2 and I3 took less initiative in inspecting the prototype independently, but were able to navigate the interface and understand what the various parts of the system were for. I2 mentioned it was "cool" to see the Django Admin interface used this way, having prior experience with the framework from previous work.

While I2 was generally positive towards the prototype system in its demonstrated state, both I1 and I3 raised points against parts of the prototype's design. Most notable of these were the prototype's focus on gathering data via direct text or numerical answers to questions (as opposed to file uploads), as well as the proposed automatic OCR design. I3 pointed out that it is not usually the case that data journalists request officials to provide lengthy, manual answers to a great deal of questions (using FOI requests as an example case), saying "we'll never get a reply if we do that". It is far more common to keep questions and answers short to minimize the hazzle for the respondent, and instead ask for file uploads containing the relevant data. I1 even spoke of general-purpose survey tools suggesting that they work more than well enough for those unusual cases where the objective is to perform some form of actual survey requiring manual data input to questions. This criticism also extends to the proposed design offering advanced sorting and searching functionality using metrics like "sentiment" for text data. If there is no great amount of text data that is directly visible in the overview table, there is no need for advanced text-based features there. I3 additionally thinks terms like "text sentiment" require too much prerequisite knowledge about things like Natural Language Recognition for most users to understand and use properly.

Criticizing the proposed automatic OCR design as shown previously in Figure 8, I3 spoke of uploaded files, which are usually some form of scanned document, pdfs of documents, or spreadsheet files, saying that "there are more interesting pieces of information in a receipt than just a final sum". All three interviewees are positive towards the idea of offering OCR as a built-in feature in a tool like the prototype, but are, especially I3, critical towards the idea of specifying a "key value" and automatically extracting it, generally saying that it is not trivial to define such a value for any given document, and that most documents might contain other interesting information that would be missed if the document is reduced to only a key value. It is better to include a built-in OCR feature that offers less experimental functionality, generally referring to the extraction of text from pdfs or other scanned document formats.

Other concrete feedback on the prototype includes comments on the ability to create respondent "profiles" within the system, I1 says this feature is "useless" if it is only available via manual input. It should be possible to import an uploaded list of names and emails, or even to retrieve this information from some online registry or scraping feature that retrieves emails from for example municipalities automatically. I3 says something similar, asking "where do you get the emails from?", further suggesting that the respondent registering feature is only realistically useful if the process can be automated satisfyingly.

I1 also expressed distaste towards getting notifications transmitted by email, especially if they are always one email for each notification. If email warnings are necessary, they say, the system should aggregate multiple notifications into a single email to reduce email spam. Ideally, the system should be able to notify the journalists about important events within the system interface instead. On this topic, I1 also talks about "versioning", saying the system should keep track of changes to allow journalists to track and access versions of the dataset as it develops over time.

All interviewees agree that being able to view respondents' progress is very useful, but I1 posits it would be even more so if the journalist is able to easily send selective reminders to only those respondents below a certain threshold towards completion. Viewing data in an online spreadsheet view within the system's interface is also accepted by all interviewees as a good way to get an overview of the dataset or the status of respondents' progress at a given time.

On the topic of graphs and queries, I1 says that graph visualization of data is useful both for producing nice graphics and for exploring data, and agrees that a graph visualization feature might be a good idea to include in a data journalism tool, although using such a feature to produce graphics with the intent of visualizing something for a piece of news content would be problematic with the Content Management Systems at their organization. A graphic SPARQL query builder sounds "cool" to I1, but is thought to likely still be a little too much for users who are not already skilled enough with information systems to be able to write queries by themselves. I2 and I3 also like graph visualization as a data exploration technique, I3 saying that their organization already uses a tool that converts tables to graphs which is easy enough to use that "the journalists" (referring to non-data journalists) are able to use it unassisted. I3 thinks that a system producing a graph of respondents and their participation across different projects would be a valuable tool.

On the general usefulness of a tool like the prototype I1 says they think it would be most useful to a team of collaborating data journalists, rather than as an organization-wide standard interface for data journalism work, especially because of the ease of documentation when all team members access the data and do things with it through the same system. I1 does not think it is a good idea to attempt to "solve everything with a single program" referring to the expressed design philosophy of the prototype and its mockups. I1 instead thinks it would be better to focus the tool towards a single specific problem, but does not explicitly mention any concrete examples. I2 says that "you are onto something useful here, but it does not need to be restricted to data journalism only, surveys are useful to all journalists", continuing saying that they think the primary realistic application of the prototype system would be to handle Freedom of Information requests. This sentiment is shared also by I3, who says that they think there "really is a point" to develop "entry-level" or "low level" data journalism tools that would open this field of reporting to ordinary journalists with lower levels of IT-skills than the specialists. They exemplify talking about (non-data) journalists that are "sitting there with a whole bunch of documents that must be went though and logged manually", saying that a useful "entry-level" data journalism system would help ordinary journalists achieve things like setting up their own database, structuring information obtained from the data, and automatically documenting their work. In such a system, I3 thinks it would be a great idea to offer advanced features like OCR built-in, available at the press of a button, but perhaps in a "restricted" form only offering commonly used operations, like making pdfs searchable, so as not to scare away less experienced users by "looking too difficult".

Summary of Findings

From the paragraphs described above, the following 9 overall findings can be summarized:

F1: Data journalists working "hands-on" are expert users comfortable with advanced, "difficult to use" (user-friendliness is relative) computer tools, meaning that the proposed prototype design choice of collecting features into one program for ease-of-use is unnecessary or may even be counter-productive by attempting to solve a non-existent problem and thereby disrupting established procedures.

F2: Introducing a new system as a "one size fits all"-solution to all data journalism is unlikely to work, it is better to let individual data journalists or units decide for themselves what tools to use on a task-to-task basis.

F3: Most general data journalism (GDJ) revolves around public data and FOI requests. Data received from these sources are often a little tricky to work with, as they may be inconsistent/unclean and/or in unpractical formats.

F4: GDJ based on FOI access requests usually involves asking for data to be uploaded as files or file archives. It is *not* common to ask public officials to manually fill in answers to lengthy forms.

F5: No standard data journalism procedure exists for FOI requests, often leading to a lack of structure and documentation, especially in cases where the project is managed by a non-data journalist.

F6: It is not uncommon for data journalists to be called away from their own projects to help non-data journalists with "light" data journalism tasks, like making a FOI access request or structuring the data received. In other words, performing GDJ on behalf of other journalists.

F7: It is unlikely that future non-data journalists as a group are going to be more adept at handling their own data, despite newer generations of journalists having grown up with computers as a ubiquitous part of their lives. Using a computer is not the same as using specialized computer tools, and without the correct mindset and knowledge about data, the tools are worthless. Put plainly, the problem described in F6 will not solve itself.

F8: Documentation and logging are important factors in good data journalism. An organized structure and a log of all steps taken is paramount for ensuring a high quality of analysis and data provenance. This aspect is often lost on non-data journalists who are not used to the same rigor when working with digital data. Self-documenting systems helping non-data journalists structure their digital data-work are wanted.

F9: Graph visualization as a data exploration technique is already actively in use by data journalists to discover links between entities and to map relations. Promoting this technique by including graph visualization and creation features in a purpose-made data journalism tool appears to be a good idea.

Results

The findings listed above can further be reduced to the following 2 concrete results of the main case investigation:

R1: A system like the prototype with its proposed designs is unlikely to be accepted by news organizations as a blanket solution to all data journalism. The kind of system the prototype demonstrates is more likely to be of use if it is adapted to solve a specific data journalism problem, as opposed to the proposed design philosophy of gathering as many features into a single interface as possible.

R2: A specific data journalism problem that can possibly be solved by an adaptation of the prototype design is freeing data journalists from supporting non-data journalists with low-level data journalism work, specifically in the form of FOI requests. A further development of the prototype system could offer a single package for structuring, carrying out, and processing FOI requests, with focus on self-documenting and ease-of-use. Such a system should allow the user to set up their own data storage, gather and register respondent profiles, issue the request, and receive and handle the data with minimal support from expert data journalists. Commonly executed tasks involving advanced features, like using an OCR program to convert a pdf to searchable text, should be offered at the press of a button. Should such a system be accepted as a standard solution to FOI inquiries across a news organization, it would be interesting to explore the potential for allowing cross-referencing of data collected from multiple inquiries (as they are all using the same system), as well as offering built-in support for creating and viewing graph representations of data and metadata.

Note on the Emphasis on Interviewee Statements

When reaching R2, heavy emphasis has been placed on statements by Interviewee 3, which is defended by this interviewee having both a leading and an organizational role at their organization's data journalism unit. With their position, this interviewee offers insight into structural, systematic problems data journalists as a group have to deal with, as opposed to the individual data journalists themselves who may or may not have been subject to these. Statements by interviewee 1 and 2 have mainly been relied upon to reach R1, as it is the testimony of the working data journalists that carry the most weight when it comes to determining if a proposed design aimed at their use is good or not.

Discussion

Distributed vs. Centralized Architectures and the Relativity of User Friendliness

The choice between working environments where there is a multitude of different programs for different things, and environments where there are only a few central pillars taking care of everything, is a choice that it seems difficult to give a definite answer to. From the research presented here, a lesson that can perhaps be extrapolated to fields other than data journalism is that "specialized programs for specific tasks" seems to be the desirable approach in those environments where the requirements of work projects can change dramatically from case to case. When it comes to data journalism, particularly of the deep, investigative kind, the people involved are usually highly competent experts with distinct personal preferences for tools and approaches that are related to each other but are not the same, like preferring one spreadsheet program over another, or preferring to analyze data in R over using another programming environment like Jupyter Notebook. Forcing these people to adopt a single given solution is likely to be met with resistance. Attempting to create a single data journalism system to handle all tasks could be likened to creating a single universal programming language, it would never work because there are so many aspects of data journalism that can be approached in different ways, and with no way to practically determine the best ones. It is better to leave that decision-making to the data journalists themselves.

As is suggested by this study's findings, expert data journalists do not mind working with multiple different programs when collecting, organizing, and analyzing data. Using one system to issue a request for data, a second system for receiving the data, a third for managing storage, a fourth for cleaning and organizing and then several different other programs for performing various kinds of analysis is routine to expert data journalists. But from a user experience-perspective, it looks very much like a problem to be solved. Surely everything would be made better by collecting all that stuff into a single system? That was the hypothesis for this research project, but it was rejected because it failed to consider the relativity of user-friendliness and the modularity of data journalism projects. The latter point refers to the way different investigative data journalism projects can have very different requirements pertaining to data security, data analysis methods, storage requirements, exchange of information, and cross-referencing with other sources, without necessarily requiring *all* of those things. A single system designed to offer everything a data journalist needs would doubtlessly be encumbered by a great number of features that would rarely be used in conjunction. A next logical step then would be developing a modular design where the users may select which features they want for a given project. But then, that is the solution that data journalists already have, only the modules are not plugins for a single program, but instead are different programs and systems themselves.

The other important point that killed the hypothesis was the realization that user-friendliness is not an absolute thing. Expert data journalist have no problems using programming scripts and statistics programs to analyze data, or accessing OCR systems via console commands. To the expert data journalist, a system designed to be user-friendly by offering simplified versions of these things, would probably be perceived as being *hard* to use or downright use*less*. Simplifying the processes involved in a data journalism project by making it a single process fully contained in a centralized system would deny the experts the freedom to work the way they want, but would allow non-experts to approach a method of reporting they would otherwise consider too complex or difficult. A very simple comparison can be made to the "iPhone vs. Android" debate, where people who want more control over their phones call the Android phones user-friendly because it lets them do what they want, while the iPhones are not user-friendly because it restricts users from doing many things, while the people who want their devices to "just work" consider the iPhone and its related products to be user-friendly because they don't require making so many decisions, while Android devices are not user-friendly because they are too complex. Catering to experts and non-experts alike in a single interface is not trivial at all, and would fortunately not be necessary in a low-level data journalism tool designed to solve an appropriate GDJ problem like FOI requests, as such a tool could be likened to an "iPhone approach to data journalism".

The relativity of user-friendliness seems obvious in retrospect, but on the contrary; when developing the prototype system, what seemed obvious was that collecting features into a centralized system would be the best design philosophy, because it fulfills the UX dogma of "making things happen with as few required user operations as possible" and thereby saving the user's time. Certainly, it is annoying when a seemingly simple operation in a computer system requires disproportionately many steps, but this aspect of user experience is only one part of overall user-friendliness. To expert users, being able to do exactly what you want is worth a more complex system that requires more operations.

FOI-centricity and Tools for Non-experts

From findings F3-F5, it is apparent that FOI legislation is an important part of data journalism, both to experts and to other journalists. Access to information is an important part of a transparent democracy, and good FOI legislation makes it easier for the media to maintain their role as "watchdogs". Making it easier to use FOI inquiries as a source of information for journalists with less advanced IT-skills thus could have positive implications for society as a whole. With more journalists collecting and inspecting data, it would be harder for public officials than ever before to hide corruption and misconduct.

The difference between the use of computers and the use of data is the reason why the problem that expert data journalists have to spend time supporting other journalists with "data work" is not going to solve itself. Even though computers compute data, using computers a lot does not make a person skilled with data. When asking one of the interviewees whether or not they thought this problem would vanish by itself with the introduction of a new generation of journalists who have grown up with ubiquitous computing, the expected answer was that they would say something along the lines of "oh yes, the young people are very skilled with computers and have no trouble with any of the digital things the older journalists struggle with". Instead, the interviewee gave the answer "absolutely not". Structuring and analyzing data are skills that do not come for free with being comfortable with computers in general. The value of maintaining an appropriate degree of rigor in documentation is also something that needs to be learned. Non-data journalists will continue to struggle with working with data also in the future, providing all the more reason to introduce systems designed to make working with data, in the specific form of FOI access requests, easier to non-experts.

To go into further detail on a possible "FOI-centric" general data journalism system developed for non-data journalists, a design could be based on the prototype system developed in this project. The prototype already allows for the creation of surveys and issuing them to respondents from within a simple interface, but has a very limited ability to manage the actual data in its current form. An updated list of requirements for a FOI-centered version for non-data journalists would include the following points:

- Improvements to the existing system identified during interviews should be implemented, such as notifications and messaging being available inside the system interface.
- The system must self-document.

- It should be possible to import respondent profiles on relevant recipients of FOI requests, or these profiles could be built-in to the system. These profiles could for example include all Norwegian municipalities.
- The system should offer simplified OCR built-in, allowing appropriate document formats to be converted to searchable text at the press of a button.
- A future version should continue exploring the potential for automatically constructing semantic graphs of the entities involved across FOI inquiries system-wide. This is an interesting opportunity to simplify the process of cross-referencing data.

The system must also be easy to set up and use in general. Even in its current, barebones version, the prototype works as a web-based tool, where the individual user does not need to worry about installing anything. A user would simply access their organization's FOI-system and could use it from any computer as long as they log in with valid credentials. When developing a low-level data journalism system for non-data journalists, further requirements gathering and evaluation should include interviews with the intended users of the system; "ordinary" journalists with an interest in public data. Future interviews should also include more individuals with managerial positions, to identify the intersection of what the journalists want to do and what their bosses want them to do. Finally, whether a future FOI-centered system should focus primarily on the receiving of data in uploaded files, or if it should expand on the current prototype's ability to create and issue forms, depends on the question below.

Unorthodox Data Collection

From finding F4, it seems possible the development of the prototype was "misled" from the start as its design was based on feedback from a young journalist who had performed an unorthodox FOI inquiry by requesting that municipality officials reply to a lengthy survey. Their survey appeared intricately constructed with several questions prompting long text answers and form logic showing or hiding different questions based on input to previous questions. The journalist complained that the officials requested to answer the survey were reluctant to do so. The prototype was then developed with particular emphasis on data collection directly via forms and surveys, for it to be revealed during the later expert interviews that data journalists generally try to avoid shaping their requests that way, exactly because officials are reluctant to respond to inquiries that are likely to be time-consuming.

Is this a case of an aspiring journalist using unorthodox methods because he still has a lot to learn, or is he challenging established patterns that perhaps should be changed? Without knowing the legality of the issue, there are points for and against both stances. It is perhaps unreasonable to expect public officials to spend hours replying to lengthy surveys from nosey journalists every day, but where does the line go? If a member of the authorities is reluctant to surrender certain information, lack of time is an excuse that can easily be used to defend not replying to inquiries. Sometimes the information the journalist seeks may not be stored in any document; what else can they do then, but make a form and request that the public officials provide the data via manual input? Before developing a FOI-centered data journalism system, these questions should be addressed, particularly the legal question of to which degree an official is obligated to comply with an inquirers demands. If it really is best to not create lengthy, elaborate surveys, the system should not offer the ability to easily do so.

Conclusion

This thesis has described a case study carried out to explore the potential of applying Information Science solutions to problems faced by data journalists. Research was conducted to attempt to answer the following two research questions:

RQ1: To what extent can data journalists benefit from purpose-made data journalism tools?

RQ2: When making a purpose-made data journalism tool, is it better to attempt to solve as many data journalism problems as possible with a single "centralized" system, or is it better to make specific programs for specific data journalism tasks?

Based on the contents of this thesis, tentative answers to the research questions can be given. Only tentative due to the qualitative nature of the study. Extrapolating general results from a small number of cases must always be done with extreme caution. The research represented by this thesis does not constitute an investigation precise enough to offer conclusive results just yet.

For RQ1, the answer given by this research project is that data journalists can find *some* benefit from purpose-made data journalism systems. Particularly, those that are designed to offer smooth solutions to *specific problems*. One example of such a problem, managing FOI requests, has been identified in this thesis. Whether or not there are more such specific data journalism problems that are suitable to be solved by a purpose-made computer tool is a subject for further research.

The answer to RQ2 is given by the answer to RQ1, it is better to develop specific data journalism tools for specific data journalism problems. Before conducting the research project, the hypothesis was that the answer to RQ2 would be the opposite; that it would constitute the greatest time-saving potential to gather as many features into one tool as possible, and that that would therefore be the best approach. This hypothesis is rejected, and a new proposed design takes its place. With some modifications and more development, the prototype system developed for this study could be tailored to provide a low-level data journalism system for the specific purpose of managing FOI requests, providing an "out-of-the-box" solution of particular use to non-data journalists.

References

Antonopoulos, N, Karyotakis, MA 2020, 'Data Journalism' in *The SAGE International Encyclopedia of Mass Media and Society*, Debra L. Merskin (ed.), SAGE Publications, Thousand Oaks, CA, viewed 24 February 2020, https://sk.sagepub.com/Reference//the-sage-encyclopedia-of-mass-media-and-society/i5415.xml>

BBC 2013, *Linked Data: Connecting together the BBC's Online Content*, viewed 11 May 2020, <https://www.bbc.co.uk/blogs/internet/entries/af6b613e-6935-3165-93ca-9319e1887858>

BBC 2014, *Linked Data: new ontologies website*, BBC, viewed 11 May 2020, <https://www.bbc.co.uk/blogs/internet/entries/78d4a720-8796-30bd-830d-648de6fc9508>

BBC 2014, *Opening up the BBC's Linked Data with /things*, BBC, viewed 11 May 2020, <https://www.bbc.co.uk/blogs/internet/entries/afdf2190-4e60-3dfc-b15f-fc17f88c85a1>

Borges-Rey, E 2016, 'Unravelling Data Journalism', *Journalism Practice*, vol. 10, no. 7, pp. 833-843

Brehmer, M, Ingram, S, Stray, J & Munzner, T 2014, 'Overview: The Design, Adoption, and Analysis of a Visual Document Mining Tool For Investigative Journalists', *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2271-2280

Brehmer, M, Ingram, S, Stray, J & Munzner, T 2014, *Overview Figure 5*, digital image, viewed 8 April 2020, <https://www.cs.ubc.ca/labs/imager/tr/2014/Overview/figures/figure5.png>

Brehmer, M, Ingram, S, Stray, J & Munzner, T 2014, *Overview Figure 6*, digital image, viewed 8 April 2020, <https://www.cs.ubc.ca/labs/imager/tr/2014/Overview/figures/figure6.png>

Chandola, V, Banerjee, A, & Kumar, V 2009, 'Anomaly Detection: A Survey ', *ACM Computing Surveys,* vol. 41, no. 3

Cohen, S, Li, C, Yang, J & Yu, C 2011, 'Computational Journalism: A Call to Arms to Database Researchers', *5th Biennial Conference on Innovative Data Systems Research (CIDR '11)*, Asilomar, California, 9-12 January, viewed 19 March 2019, <http://cidrdb.org/cidr2011/Papers/CIDR11_Paper17.pdf>

DocumentCloud.org 2018, *DocumentCloud*, DocumentCloud, viewed 22 March 2020, <https://www.documentcloud.org/>

Dresch, A, Lacerda, DP & Antunes, JAV 2015, *Design Science Research – A Method for Science and Technology Advancement*, Springer, New York

Fink, K & Anderson, CW 2015, 'Data Journalism in the United States', *Journalism Studies*, vol. 16, no. 4, pp. 467-481

Flew, T, Spurgeon, C, Daniel, A & Swift, A 2012, 'The Promise of Computational Journalism', *Journalism Practice*, vol. 6, no. 2, pp. 157-171

Haag, F, Lohmann, S, Bold, S & Ertl, T 2014, 'Visual SPARQL Querying Based on Extended Filter/Flow Graphs', *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*, Como Italy, May 2014, viewed 11 May 2020, <https://dl.acm.org/doi/pdf/10.1145/2598153.2598185>

Hevner, AR, March, ST, Park, J, & Ram, S 2004, 'Design Science in Information Systems Research', *Management Information Systems Quarterly,* vol. 28, no. 1, pp. 75-105

Karlsen, J & Stavelin, E 2014, 'Computational Journalism in Norwegian Newsrooms', *Journalism Practice*, vol. 8, no. 1, pp. 34-48

Kierulf, A, Gisle, J, Elden JC, 2018, 'ytringsfrihet' in *Store norske leksikon*, viewed 28 May 2020 <https://snl.no/ytringsfrihet>

La Nacion 2014, *VozData: collaborating to free data from PDFs – The Senate Expenses part II*, La Nacion, viewed 7 May 2019, <http://blogs.lanacion.com.ar/projects/data/vozdata/>

Lazar, J, Feng, JH & Hochheiser, H 2017, *Research Methods in Human-Computer Interaction*, Morgan Kaufmann Publishers, Cambridge, MA

Meyer, T 2012, *What We Learned from Free the Files -- and How to Make It Better*, ProPublica, viewed 7 May 2019, <https://www.propublica.org/article/what-we-learned-from-free-the-files-and-how-to-make-it-better>

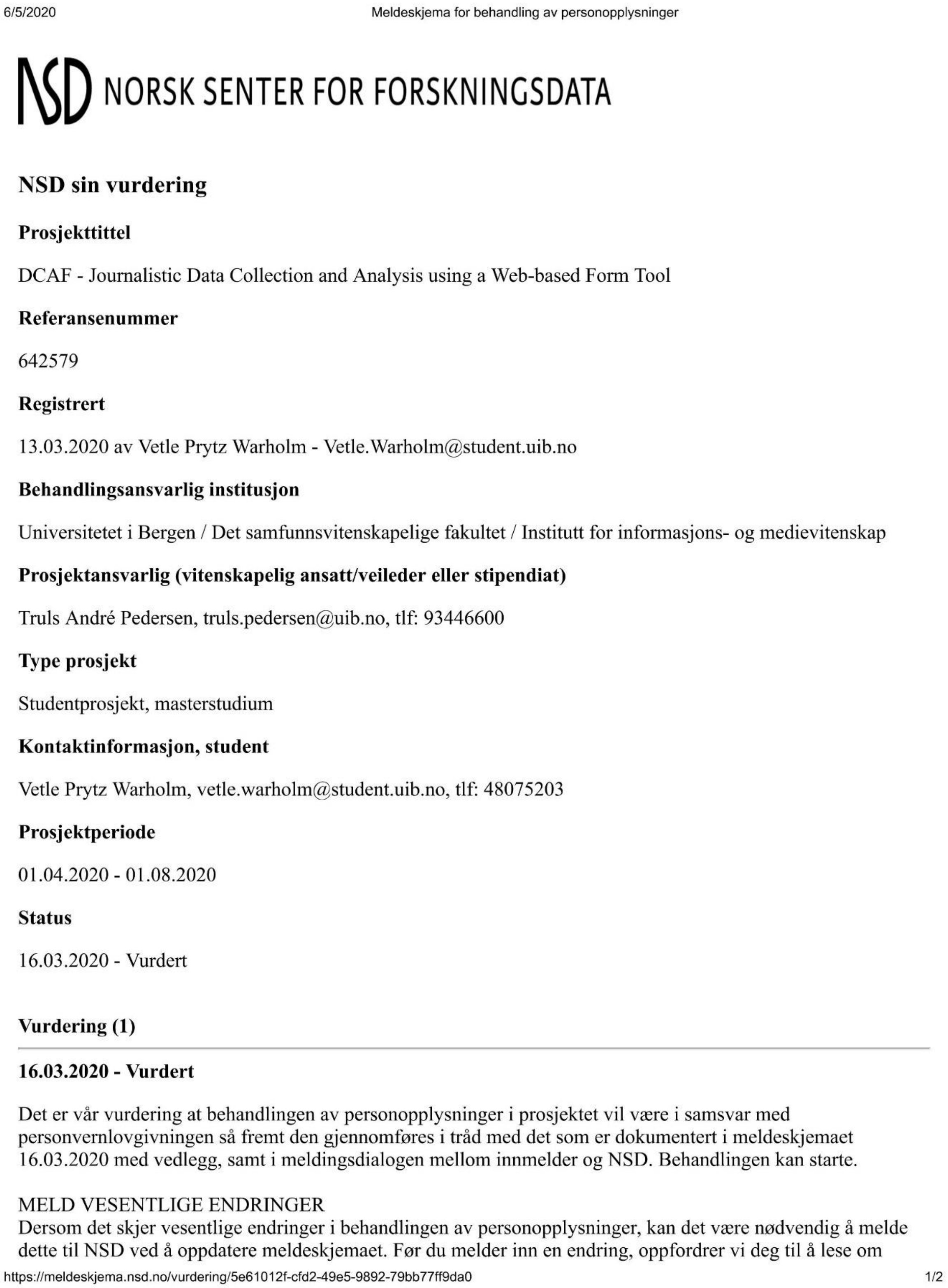
Oracle 2017, *Introduction to Anomaly Detection*, Oracle, viewed 11 May 2020, <https://blogs.oracle.com/datascience/introduction-to-anomaly-detection>

Rogers, S 2009, *How to crowdsource MP's expenses*, The Guardian, viewed 7 May 2019, <https://www.theguardian.com/news/datablog/2009/jun/18/mps-expenses-houseofcommons>

Senter for Undersøkende Journalistikk 2020, *Om SUJO*, Senter for Undersøkende Journalistikk, viewed 28 May 2020, <https://sujo.no/om-oss/>

Uskali, T & Kuutti, H 2015, 'Models and Streams of Data Journalism', *The Journal of Media Innovations*, vol. 2, no. 1, pp. 77-88

Appendix



6/5/2020 Meldeskjema for behandling av personopplysninger

NSD NORSK SENTER FOR FORSKNINGSDATA

NSD sin vurdering

Prosjekttittel

DCAF - Journalistic Data Collection and Analysis using a Web-based Form Tool

Referansenummer

642579

Registrert

13.03.2020 av Vetle Prytz Warholm - Vetle.Warholm@student.uib.no

Behandlingsansvarlig institusjon

Universitetet i Bergen / Det samfunnsvitenskapelige fakultet / Institutt for informasjons- og medievitenskap

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Truls André Pedersen, truls.pedersen@uib.no, tlf: 93446600

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Vetle Prytz Warholm, vetle.warholm@student.uib.no, tlf: 48075203

Prosjektperiode

01.04.2020 - 01.08.2020

Status

16.03.2020 - Vurdert

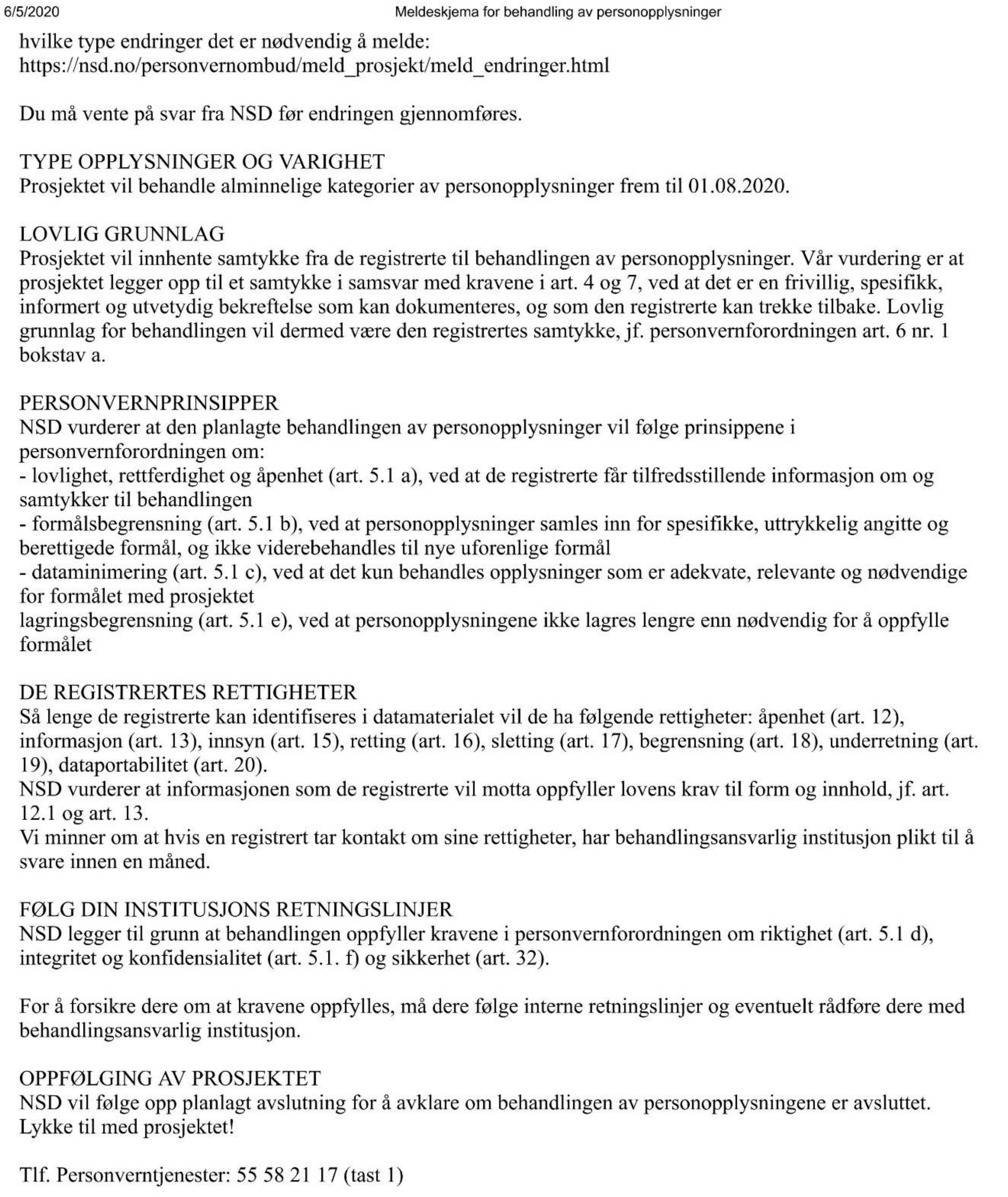
Vurdering (1)

16.03.2020 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet 16.03.2020 med vedlegg, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

MELD VESENTLIGE ENDRINGER
Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om

https://meldeskjema.nsd.no/vurdering/5e61012f-cfd2-49e5-9892-79bb77ff9da0 1/2



hvilke type endringer det er nødvendig å melde:
https://nsd.no/personvernombud/meld_prosjekt/meld_endringer.html

Du må vente på svar fra NSD før endringen gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET
Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 01.08.2020.

LOVLIG GRUNNLAG
Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER
NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:
- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke viderebehandles til nye uforenlige formål
- dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER
Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), underretning (art. 19), dataportabilitet (art. 20).
NSD vurderer at informasjonen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.
Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER
NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og eventuelt rådføre dere med behandlingsansvarlig institusjon.

OPPFØLGING AV PROSJEKTET
NSD vil følge opp planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.
Lykke til med prosjektet!

Tlf. Personverntjenester: 55 58 21 17 (tast 1)

b

COPY OF INTERVIEW GUIDE USED DURING EXPERT INTERVIEWS

Åpning – *ca. 5 min*

Signere samtykkeskjema.

Dette intervjuet utføres i forbindelse med min masteroppgave i Informasjonsvitenskap ved Universitetet i Bergen. Jeg heter Vetle Prytz Warholm, og veilederen for prosjektet er førsteamanuensis Truls André Pedersen.
Tidligere i prosjektet har jeg intervjuet en journalist som var i sluttfasen av et prosjekt der han samlet data fra landets kommuner via et generelt, nett-basert skjemaverktøy. Intervjuet med ham avdekket en rekke punkter der verktøyet han brukte kunne vært bedre. Basert på disse punktene har jeg utviklet en prototype av et lignende nett-basert skjemaverktøy som er ment å være bedre rustet til journalistarbeid.
Formålet med dette intervjuet er å få høre litt om din bakgrunn som (data)journalist og erfaringer du kanskje har gjort deg i tidligere møter med informasjonsteknologi og arbeid med datainnsamling. Jeg ønsker også å vise deg prototypen jeg har utviklet for å få tilbakemeldinger og innsikt som kan bidra til videre utvikling og forbedring av denne. Målet med utviklingen er å skape et system som journalister føler er enkelt å bruke, som er tidsbesparende, og oppleves trygt. Kort sagt bedre egnet enn konkurrerende generelle verktøy.

Om journalisten – *ca. 15 min*

Erfaring med arbeid med digitale data. Har du utført den type arbeid vi snakker om før?

Hva?

Hvordan? Omfang og verktøy som ble brukt? Scripts, databaser, e.l.

Hvordan opplevde du at systemene fungerte? Kunne noe vært bedre?

Hva slags typer data mener du er de mest interessante å få tak i og jobbe med?

Kilder, finnes andre viktige kilder enn offentlige myndigheter?

Type data og formater

Hvordan har dere i tidligere prosjekter håndtert datasikkerhet?

Mindre viktig dersom dataene i utgangspunktet er "offentlige"?

Avslag på forespørsler pga. dårlige systemer?

Prototype og mock-ups – *ca. 20 min*

Vi ser på prototypen. Jeg viser frem hovedfunksjonene, hvis intervjuobjektet ønsker det kan de utforske litt på egenhånd.

Hva tenker intervjuobjektet om måten respondenter håndteres på? i.e. som egne entiteter i databasen som legges til eksplisitt til enhver undersøkelse de blir bedt om å delta i, og hvor invitasjon loggføres?

Åpenbart har prototypen mangler i mengden spørsmålstyper som går an å lage, og måten disse opprettes på. Systemet bærer preg av å være "ankret" i den underliggende databasestrukturen. Har journalisten noen kommentar til dette?

Hva med muligheten for å bruke spørsmål om igjen? Eller å kunne lagre maler til egne typer skjemaer?

Hva tenker du om å kunne kikke på innsamlede data inne i verktøyets grensesnitt?

> Er "regneark" det best egnede formatet her? Hadde det vært bedre å presentere data på en annen måte, f.eks. som skjema per respondent?

Hva er intervjuobjektets vanlige prosedyre når det er på tide å analysere data skikkelig?

> Csv-fil og excel? Viser fram mockup: ***notify_data_view.png***
>
> Håndtering av filer? Viser fram mockup: ***download_form_view.png***

Hva med automatisk "information extraction" fra visse filtyper? Vis mockups: ***Sortsearch_data_view.png, ocr1_data_view.png***

> Har intervjuobjektet foretatt seg noe slikt før selv?
>
> Kan de komme på noen tilfeller der de tror dette vil være nyttig?

Semantiske data – *ca. 10 min*

Jeg spør om intervjuobjektet har noen formening om hva semantiske data går ut på. Har de vurdert om dette er nyttig teknologi i journalismearbeid?

Kort fortalt er RDF-tripler et datalagringsformat som representerer data som entiteter med relasjoner mellom hverandre (*trippel* fordi entitetene og relasjonene står i "subjekt-predikat-objekt"-form). Semantiske data er "linked", altså koblet sammen i et nettverk (en *graf*), som gjør det mulig å foreta spørringer som henter inn relasjoner på kryss og tvers. Eksempler som prototyp-systemet kan svare på er "hvilke respondenter har besvart undersøkelser som ble publisert før/etter en gitt dato?" eller "gi meg alle svar som er oppgitt av respondenter med epost-adresse som inneholder et visst mønster". Prototypen lagrer denne grafen i en turtle-fil, et av flere egnede formater for lagring av RDF-tripler. I et system med integrerte funksjoner som benytter seg av RDF-data ville ikke nettverket nødvendigvis lagres som en slik fil, men for prototypen er det et velegnet format for å vise slike funksjoner ved å åpne filen i andre programmer.

Vis mockup: ***Graph_resp_view.png*** og åpne grafen i Protege for å vise hvordan grafvisualisering kan se ut og hvordan SPARQL-spørringer ser ut.

Tror intervjuobjektet at slik teknologi kan være anvendbar til deres type arbeid? Ville de benyttet en spørringsbygger dersom de hadde tilgang til en?

Hva med en grafvisualisering? Etter å ha vist dem hvordan en implementering av dette fungerer i Protege, hva tror de om denne måten å utforske data på? Hvilke ledd er viktigst å vise fram i en "destillert" grafvisualisering?

Avsluttende spørsmål og debrief – *ca. 10 min*

Intervjuet avsluttes med åpne spørsmål om intervjuobjektet har noe mer å tilføye. Har de kommet på noen flere funksjoner som kan være nyttige?

Tror du din arbeidsplass ville foretatt seg mer undersøkende datajournalistikk dersom de hadde tilgang til slike verktøy som vi har snakket om i dette intervjuet?

> Hva med journalisten selv? Synes vedkommende prototypen virker vanskelig?
>
> Hva med kolleger med lavere grad av IT-kompetanse?

Til slutt går vi gjennom det som har blitt sagt og gjort i løpet av intervjuet for å forsikre oss om at ingen er misforstått.

COPY OF INTERVIEW DATA ANALYSIS DOCUMENT

Themes:

Turquoise	Uses advanced contemporary tools, has no problem working with third-party programs
Yellow	Data sources mostly include public databases and Freedom of Information access requests
Red	Official data (open database or FOI) are often more difficult to use than anticipated
Teal	Sometimes, simpler solutions are better, new systems may be unwanted
Pink	Documentation is an important part of good data journalism
Bright Green	Prototype criticism, against proposed designs
Blue	Suggesting improvements to proposed designs, or completely new features
Dark Yellow	Prototype praise, positive towards proposed designs
Violet	Need to accommodate for less technologically competent/experienced journalists

I1

I1 has worked for NRK Brennpunkt performing investigative data journalism. He defines a data journalist as a journalist with the ability to find, structure, and exploit data to create news content. In previous work I1 has drawn data from public databases and via Freedom Of Information requests. I1 says that Norwegian bureaucrats know little about data (in digital form, as in they don't know how to export their data in useful formats) (this is a point that is also made in other research papers, like Karlsen & Stavelin). Has simply used Google Forms to collect data, feels this structures the collected data sufficiently (?).

I1 inspects data using spreadsheets, prefers Google Spreadsheet over Excel as he thinks the latter is too "heavy" or "cumbersome". I1 also likes using R to analyze data and create statistics. R allows "good control" of the data. Many things can be done directly within the R working environment.

I1 has worked on some relatively recent digital features with the NRK, one is the live numbers of Corona patients infected/hospitalized/dead that are displayed on the NRK news site. A problem encountered during this work was that the updated numbers are transmitted between components via email. I1 does not like email. Another thing is the "Valgomat", an online tool for helping voters find the political party that most closely represents their stance on various issues. This tool collects data from municipalities (I1 was unclear as to exactly what kind of data) and is developed in-house at the NRK. I1 feels the programming is "tangled" or "unnecessarily complicated", and generally bad and old.

When asked which data sources (realistically available ones) I1 finds the most interesting, the reply is SSB (Norwegian bureau of statistics), the government, FOI requests, various state organs. I1 says municipalities "are each in their own system" making it hard or impossible to search for data across them. SSB has a public(?) API, but this is difficult to use.

I1's previous work has not seen the need to take particular care about data security.

When shown the prototype, I1 suggests it is useless to be able to register respondents within the system if this has to be done manually. This is just moving a manual labor task from one system to another, though they comment moving it to a system potentially helps with logging. It should be possible to upload a list of respondents, or even import from a "public register of mails" (if such exists, I1 did not provide concrete example). Likes the idea of being able to save questions or entire forms for later reuse. Likes being able to view respondent progress, but misses the ability to send selective reminders to those respondents who have not responded (idea: send reminder to those respondents below X% progress). Does not like receiving warnings/notifications on things via email. If it has to email, aggregate a series of notifications into a single email, but better to bring this function into the tool directly. Mentions "versioning", where the tool could save new versions of the dataset as it is updated (so the journalists could revert to, or inspect, older versions of the data if required). Reiterates that email-spam is the worst thing in the world.

Technical note, I1 thinks comma-separated-values are a potential hazard with Norwegian(European?) decimal commas (as opposed to ".").

For the spreadsheet view within the prototype system, I1 says that it would be easy and useful to be able to sort by common statistical metrics, mentions the "summary" function from R as an example.

Likes the idea of having OCR built-in to the tool, could save time on scraping pdfs for text. Has previously used OCR to obtain text from scanned documents. Uses R(?) for this, outputs to JSON objects. When asked if automatic information extraction (just automatic OCR really) with OCR might be useful, I1 responds mentioning "archives of scanned documents".

At their work in the NRK, I1 uses graph visualization to keep track of people (contact networks). Thinks graph visualization is useful both for producing cool graphics and for exploring data, says the technique may be beneficial to include in a data journalism system. Makes a point of the difficulties in creating content from plugins for NRK news articles. All plugin content must pass through NRK's "IT-desk" (development-desk?) before being approved for the Content Management System.

I1 says a graphic query builder would bee "cool", but "needs to not be too complicated for those who are not experienced with this kind of thing". Also makes a point of making things like a graphic query builder (and any other advanced feature, like the OCR) work with the Norwegian language. Compatibility and performance with "small" languages are not given.

When asked if his organization would benefit from a tool like the prototype acting as a standardized platform for specific kinds of tasks like general data journalism, I1 responds saying that "NRK is a many-headed troll" and "it [NRK] is so large that it seems impossible to establish a single system for any specific task", meaning that there are many different units and teams across the organization that approach the same challenges from different angles, and that standardizing these solutions would be unlikely to work for everyone. I1 also mentions that communications currently happen across multiple different platforms (MS Teams, Skype, email, etc.), adding to the difficulties in introducing a standardized solution for data journalism at this time. However, I1 does also say that a tool like the prototype would be useful for a collaborating team, like their own. Would like to see a more developed iteration of the system and could help test in a real use case with their team.

Asking I1 about the usefulness of a tool like the demonstrated prototype to less technologically competent colleagues, I1 responds saying that such tools must generally be easy to use. "Not all journalists are professionals" (probably referring to journalist with low levels of IT-skills). They say it might be better to focus the tool towards one or a few specific problems instead of trying to solve "everything with a single program". Exemplifies with MS Teams (it does many things, but none of them particularly well, as it were).

Lastly, I1 mentions the possibility of exporting and importing data to and from other form-platforms and formats as a possibly useful feature for future development.

I2

I2 works for Bergens Tidende as a data journalist. I2 says they and their data journalist colleagues work using tools like Jupyter Notebook with Pandas dataframes to inspect and analyze data. When asked about their data sources, I2 mentions SSB and Geonorge before mentioning that "knowing where data are to be found is important". Also mentions municipalities and other regional authorities/officials via FOI requests.

When asked about their approach to data journalism, I2 mentions they work with training new data journalists within their organization, and begins listing some of the things they teach. "Rule number one: understanding spreadsheets is crucial", with particular emphasis on filtration of data. "If you know and understand spreadsheets, you are nine miles further ahead". Another point is to always keep a copy of the raw data. Logging each step taken is important, documentation is taken seriously at their organization. I2 also talks about the importance of learning "some simple statistics", saying many data journalists make mistakes "doing math on things they don't understand" referring to calculating statistical properties without being quite aware of what exactly the numbers mean.

I2's organization are very serious about data security. In previous projects involving sensitive data, I2's team used a separate data server with restricted access. Their organization uses "Securedrop" for anonymous tips, which is a system using the Onion network. The only way to access the data collected via this service (securedrop) is to login to a specific computer using physical login credentials in the form of a memory stick (presumably containing a secret key).

Looking at the prototype, I2 expresses interest and thinks it is cool to see the Django framework used this way, having been introduced to it in a previous work project. I2 offers comments and asks questions about the prototype as we go along and different aspects of the prototype are introduced and explored. I2's disposition towards the prototype seems positive throughout the interview, although little concrete feedback is given beyond several general "this looks quite nice" and "that might be very useful". It appears I2 is reluctant to criticize a student's prototype, mostly asking questions that feel like they are not intended to place the prototype in a bad light.

On the topic of semantic data (particularly graphs) I2 says they use "NeoForJ" for graph-data. They use this to map "owner-structure and roles, and so on" and making connections between people involved in different things. Thinks graph visualizations are a useful way to explore data.

During closing comments, I2 says that "you are into something useful here, this does not have to be restricted to data journalism only, surveys are useful to all journalists", also saying that the main application of a tool like the demonstrated prototype would be FOI access requests. I2 mentions that as far as they are concerned, a very useful feature in a tool like this is tracking which respondents have responded or not.

I3

I3 works as a "News Developer"(?) (Nyhetsutvikler) at Bergens Tidende, as a leader for "journalists who know some programming" a small group. Their little group is multi-disciplinary (tverrfaglig, not a perfect translation), working on their own data journalism projects or rendering assistance to other journalists that need IT know-how. I3 says they "collect datasets" and "work wide". I3 themself does not work "hands-on" with data journalism, has a more organizational role verifying the work of other team members. Says they "run through calculations and analysis to ensure quality", in meetings with the team as well as discussing ways to best present data and results to readers.

On the topic of FOI access requests, I3 states that "this is a field of expertise [fagfelt] in and of itself". Their organization has a "postlist"-system for this purpose. Says that it is really up to the individual journalist how to structure this type of work. There is no standard.

Generally says that logging and documenting are applications where better systems would be beneficial.

On the prototype, I3 mentions that to create the respondents in the database, you have to know their emails. Where do you get those? Likes being able to get an overview of data via an in-system spreadsheet view. Useful to track respondents' progress. Not super excited about sorting and searching text answers by "sentiment" or other metrics. These terms require a certain degree of knowledge about language and NLR (Natural Language Recognition) technologies to understand and use correctly.

I3 makes an important point stating that it is very rare that data journalists construct datasets from extensive text-answer questions (i.e. large forms where the respondent has to type answers), the bigger the form/survey the less likely a respondent is to reply (FOI legislation does not require officials to respond to everything in the exact way the inquirer specifies, it needs to be within reason). When asking questions to be answered directly in the form (i.e. not by uploading a file) they try to keep things as short as possible, preferring numbers over text. It is more common that they instead request document uploads, mentions cross-referencing and aggregating documents ("sammenstilling", here translated to "collecting documents from multiple sources to aggregate and cross-reference information"). I3 says they often receive documents in "unreadable formats" (probably referring to the common phenomenon of government/regional officials not knowing how to export their data properly).

I3 also makes another important point against the proposed automatic OCR feature, saying that "there are more interesting pieces of information in a receipt than just a sum" as an example against the idea that a concrete "key value" can be defined for any kind of document. It is useful to use OCR to make data in scanned documents available (mentions making pdfs searchable), but is skeptical towards reducing an uploaded document to a single value. Is positive towards providing OCR as a built-in feature in the tool, for example to make files like pdfs searchable or other "simple things like that".

I3 says they like using graphs to explore data and to discover connections between things. Says they use a web-based tool called "Datawrapper" for this, upload a table, output a graph. Says this is easy enough to use that "the journalists" (referring to non-data journalists) can make their own graphs themselves. Says they do the same with maps? Likes the idea of having graph-based features inside the tool, thinks a graph over respondents and their participation across different projects would be valuable.

I3 says that there really is a point developing "entry-level" tools for data journalism that opens the field up to journalists with lower levels of IT-skill. Talks about organizing data and logging work, also providing an easy way for journalists to set up their own databases, exemplifies mentioning journalists "sitting there with a whole bunch of documents that must be went through and logged manually". Says that most (non-data) journalists "struggle with structuring and logging", and "this is about data-competence, structuring data is something they are really reluctant to do" ("det sitter langt inne"). Says that a "system that documents itself" would be great, or at least something that achieves better structures among those journalist not so inclined. Also structuring information obtained from data. In this case, I3 likes the idea of gathering advanced features like OCR and offering them within the system in some appropriate way, perhaps in a "restricted" form that does not scare less experienced users away by looking too difficult, the average journalist has "no chance" trying to use Tessearct (for example) via the console to read a pdf. From I3's comments, it is clear that experienced programmer-journalists have no issues working like that, and have no real need for a system that collects these features for themselves, but would be thankful for the reduced need to provide "tech-support" for less IT-experienced colleagues. (an issue possibly also mentioned by I2 and in other research papers; that data journalists often end up doing all kinds of work across sections of their organization because of their multi-disciplinary skillset).

Having already said that a tool like the prototype would be useful to allow less technologically competent journalists to carry out their own data journalism projects, I3 was asked whether or not they thought journalists like that would become less common in the future (referring to the increasing permeation of computers and data throughout society, and younger generations growing up using computers as a natural part of their lives). To this, I3 responded saying "absolutely not, technological competency has little to do with age; fresh young students of journalism who have not taken any computer courses are not better at excel than older journalists".