NEWS MATTER
Embedding Human Intuition in Machine Intelligence through Interactive Data Visualizations

by

Yasmine Rubinovitz
B.Sc, Hebrew University of Jerusalem (2014)
Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2017
© Massachusetts Institute of Technology 2017. All rights reserved.

Signature redacted
Author ...................... Program in Media Arts and Sciences
Signature redacted

May 12, 2017

Certified by ...........
Andrew Lippman
Associate Director & Senior Research Scientist, MIT Media Lab
Thesis Supervisor

Signature redacted
Accepted by ............
Professor Patricia Maes
Academic Head, Program in Media Arts and Sciences
NEWS MATTER

Embedding Human Intuition in Machine Intelligence through Interactive Data Visualizations

by

Yasmine Rubinovitz

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning on May 12, 2017, in partial fulfillment of the requirements for the degree of Master of Science in Media Arts and Sciences

Abstract

In this era of luxurious information, we are free to access as many news stories as we want. However, news is so abundant that people don’t have the time to consume all of it, nor the time to select which stories they want to know about. We trust editors and algorithms to decide for us, giving away our control and sometimes missing the big picture.

For computers, news stories are usually not annotated or categorized, they come in as an unstructured text that for machines is hard to generalize. While numerous tools exist that use Natural Language Processing to identify features of news articles, few use NLP to help readers navigate the universe of news stories.

This thesis proposes a novel interaction method, coupling principles of data visualization and user experience with an interactive machine learning approach to ease our understanding and exploration of mass information while collecting nuanced annotations for the same information. We present a human machine collaboration where the computer analyzes and renders the data, making it easier for the reader to explore. The user in turn gives annotated labels that help the computer better analyze the next data points.

As a proof of concept, we present Panorama, an interface for open, transparent and collaborative exploration of news. Panorama addresses information overload, by allowing users to filter, organize and control their news feed. Panorama is also an interactive machine learning system. As the user reads and explores the news that were analyzed by machine learning models, she is encouraged to submit feedback that is sent back to these underlying models, helping them improve.

This work explores the relationship between knowledge and design. It demonstrates how data visualization and interfaces help humans understand, build, control and improve a system based on machine intelligence.

Thesis Supervisor: Andrew Lippman
Title: Associate Director & Senior Research Scientist, MIT Media Lab
NEWS MATTER

Embedding Human Intuition in Machine Intelligence through Interactive Data Visualizations

by

Yasmine Rubinovitz

The following people served as readers for this thesis:

Signature redacted

Thesis Reader

Ethan Zuckerman
Director, Center for Civic Media, MIT
Associate Professor of the Practice, MIT Media Lab
NEWS MATTER

Embedding Human Intuition in Machine Intelligence through Interactive Data Visualizations

by

Yasmine Rubinovitz

The following people served as readers for this thesis:

Signature redacted

Thesis Reader . . .

Dr. Fernanda B. Viégas
Director, 'Big Picture' Visualization Group
Google Brain
Acknowledgments

I would like to thank my advisor, Andy Lippman, for the opportunity to join the Media Lab, and for the support, guidance, crazy ideas, conversations and inspiration that will remain with me for many years to come.

I also thank my readers Ethan Zuckerman and Fernanda Viégas for sparking the ideas that eventually became this thesis, and for fun brainstorming sessions and their great feedback, advice and encouragements along the way.

I thank my friends at the Viral Communications group: Travis, Thariq, Tomer, Ariel, Brian, Léo and Hisham, for your kindness, support and awesome discussions, and for making the lab a fun place to be at. I especially thank Travis Rich and Tomer Weller, for sharing your knowledge and code, and for all the technical and emotional support during the last two years. You taught me more than you can imagine. I also thank Deb Widener, for caring and helping all of us with anything we needed.

I thank Penny Webb, for a beautiful friendship, and for your practical guidance and help through the design process. Thanks for helping me and giving me your great eye and advice in so many steps along the way.

I thank all my friends and colleagues at this amazing place called MIT Media Lab, this adventure would not have been the same without you. I’m grateful for all the fun, laughs, amazing ideas and long conversations we shared.

I thank Keira and Linda for all their help and support from the very first moment I applied to the lab.

I thank Michael Fink, for making design and computer science possible, for being an amazing mentor and a source of inspiration and for always finding the time to listen and give me good advice.

I thank my beloved family and friends in Israel, for being patient and letting me know I’m loved even when we are 12 hours of flight apart. Mom, Dad and Barak, thanks for your endless love and support, for giving me the feeling there is nothing I can’t learn or do, and for your design and writing feedback, many times at 5am.

Finally, I thank Eran, for everything that you are.
Contents

1 Introduction .................................................. 15
  1.1 Motivation .................................................. 15
  1.2 Purpose .................................................... 16
  1.3 Contribution ............................................... 17
  1.4 Overview of the Thesis ................................. 17

2 Background and Related Work ................................ 19
  2.1 NLP and News Analysis .................................. 19
  2.2 Interactive Machine Learning and Collective Intelligence ............... 21
  2.3 Data Visualization ........................................ 23
    2.3.1 News Data Visualizations .......................... 24
  2.4 News Feeds and Content Aggregators ...................... 26
    2.4.1 The Power of Aggregators ........................ 28
  2.5 Synthesis .................................................. 34

3 Data Sources and Preliminary Work ......................... 35
  3.1 Data Sources ............................................. 35
    3.1.1 Media Cloud ........................................ 35
    3.1.2 Superglue ........................................... 36
    3.1.3 Feeding Back to the Data Sources .................. 36
  3.2 Preliminary Work ......................................... 37
    3.2.1 News Graph .......................................... 38
    3.2.2 Perspectives ........................................ 40
## 4 Implementation Details

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Overview of the Implementation</td>
<td>43</td>
</tr>
<tr>
<td>4.2 Modeling News Stories with Word2vec</td>
<td>43</td>
</tr>
<tr>
<td>4.3 Multi-label Classifier For News Topic Detection</td>
<td>46</td>
</tr>
<tr>
<td>4.3.1 NYT Annotated Corpus</td>
<td>46</td>
</tr>
<tr>
<td>4.3.2 Multi-Label Descriptors Classifier for News Stories</td>
<td>48</td>
</tr>
<tr>
<td>4.4 Sentiment and Subjectivity Analysis Classifiers</td>
<td>57</td>
</tr>
<tr>
<td>4.4.1 Sentiment Analysis</td>
<td>57</td>
</tr>
<tr>
<td>4.4.2 Subjectivity Analysis</td>
<td>58</td>
</tr>
<tr>
<td>4.4.3 Logistic Regression With Stochastic Gradient Descent</td>
<td>59</td>
</tr>
<tr>
<td>4.5 Left-Right Leaning Sources</td>
<td>60</td>
</tr>
<tr>
<td>4.6 Trend Detection</td>
<td>61</td>
</tr>
<tr>
<td>4.7 Panorama Database Structure</td>
<td>64</td>
</tr>
<tr>
<td>4.7.1 Panorama database tables</td>
<td>64</td>
</tr>
<tr>
<td>4.8 Data Aggregation Architecture</td>
<td>66</td>
</tr>
<tr>
<td>4.9 Interface Architecture</td>
<td>67</td>
</tr>
</tbody>
</table>

## 5 User Experience and User Interface

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Panorama Interface Components</td>
<td>69</td>
</tr>
</tbody>
</table>

## 6 Evaluation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Experiment Design</td>
<td>77</td>
</tr>
<tr>
<td>6.1.1 User Study</td>
<td>77</td>
</tr>
<tr>
<td>6.1.2 User Statistics</td>
<td>78</td>
</tr>
<tr>
<td>6.2 Evaluation Results</td>
<td>78</td>
</tr>
<tr>
<td>6.3 Study Conclusions</td>
<td>80</td>
</tr>
<tr>
<td>6.4 Further Evaluation</td>
<td>81</td>
</tr>
</tbody>
</table>

## 7 Conclusion and Future Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Future Work</td>
<td>83</td>
</tr>
<tr>
<td>7.2 Overall Achievements</td>
<td>85</td>
</tr>
</tbody>
</table>
List of Figures

2-1 Interactive vs. traditional machine learning ........................................... 21
2-2 Newsmap data visualization ................................................................. 24
2-3 Wall of Now ....................................................................................... 25
2-4 Unfiltered.News .............................................................................. 26
2-5 Google News ................................................................................. 28
2-6 Facebook Newsfeed ................................................................. 29
2-7 Blue Feed, Red Feed ...................................................................... 31
2-8 Perspecs News ............................................................................. 32
2-9 AllSides News .............................................................................. 33
3-1 Superglue architecture ............................................................... 37
3-2 News Graph .................................................................................. 38
3-3 News Graph - Connections highlighted ........................................ 39
3-4 News Graph - Entities selected ..................................................... 39
3-5 Perspectives .................................................................................. 40
4-1 Topic model flow ........................................................................ 50
4-2 Convolutional Neural Networks for Sentence Classification .......... 50
4-3 Sequential layer ........................................................................... 51
4-4 CNN Model ................................................................................ 51
4-5 Topics model coverage over Media Cloud topics ......................... 53
4-6 Oscar topic ................................................................................... 54
4-7 Common Core topic ................................................................. 54
<table>
<thead>
<tr>
<th>Page</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-8</td>
<td>Ebola Nigeria topic</td>
<td>55</td>
</tr>
<tr>
<td>4-9</td>
<td>Nigeria Contraceptives topic</td>
<td>55</td>
</tr>
<tr>
<td>4-10</td>
<td>India Rape topic</td>
<td>56</td>
</tr>
<tr>
<td>4-11</td>
<td>US MSM 2016</td>
<td>56</td>
</tr>
<tr>
<td>4-12</td>
<td>US MSM 2017</td>
<td>57</td>
</tr>
<tr>
<td>4-13</td>
<td>Subjectivity score distribution in Panorama</td>
<td>59</td>
</tr>
<tr>
<td>4-14</td>
<td>Trends on Google News</td>
<td>62</td>
</tr>
<tr>
<td>4-15</td>
<td>Trends on Twitter</td>
<td>63</td>
</tr>
<tr>
<td>5-1</td>
<td>Panorama interface components</td>
<td>70</td>
</tr>
<tr>
<td>5-2</td>
<td>Panorama - topic list</td>
<td>70</td>
</tr>
<tr>
<td>5-3</td>
<td>Panorama - selected topic</td>
<td>71</td>
</tr>
<tr>
<td>5-4</td>
<td>Panorama - search bar</td>
<td>71</td>
</tr>
<tr>
<td>5-5</td>
<td>Panorama - selected source</td>
<td>72</td>
</tr>
<tr>
<td>5-6</td>
<td>Panorama - sources component</td>
<td>72</td>
</tr>
<tr>
<td>5-7</td>
<td>Panorama - filters component</td>
<td>73</td>
</tr>
<tr>
<td>5-8</td>
<td>Panorama stories component</td>
<td>74</td>
</tr>
<tr>
<td>5-9</td>
<td>Panorama - story labeling</td>
<td>75</td>
</tr>
<tr>
<td>6-1</td>
<td>User study demographics</td>
<td>78</td>
</tr>
<tr>
<td>6-2</td>
<td>User study results</td>
<td>80</td>
</tr>
<tr>
<td>6-3</td>
<td>Evaluation - events tracked with Google Analytics</td>
<td>81</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Descriptors and Taxonomies in the NYT annotated corpus .................. 48
4.2 Stories table part 1 .................................................. 64
4.3 Stories table part 2 .................................................. 64
4.4 DescriptorsResults table ................................................. 65
4.5 Connections table ..................................................... 65
4.6 Labels table part 1 ..................................................... 66
4.7 Labels table part 2 ..................................................... 66
6.1 User study instructions .................................................. 79
Chapter 1

Introduction

1.1 Motivation

The news can feel like a flood, with stories flowing in as they happen, from multiple professional sources, eyewitnesses, blogs and broadcasts. Just to give an idea, the 24 most visited media sources in the U.S alone generate 8000 stories on the web every day. This amount of information can be overwhelming. People are free to access whatever news sources they desire, however, reading or watching all of this information becomes an impossible task.

A common solution to handle the information overload is the use of content aggregators, such as a Facebook feed, or the decision to trust one source, such as a specific newspaper or television show. Editors or algorithms choose the content, without the reader knowing what are the rules and reasons behind those decisions.

Consuming stories from a single source can give us only part of the big picture of the current events in the world. As information overflows - gathering, filtering and visualizing what is happening beyond what the eye can see, is essential.

For computers, news stories present a different problem. Machines can read copious amounts of text quickly, yet, making sense of this text is a harder task since news stories rarely contain machine-readable annotations or metadata. While machine learning and statistical models are improving at providing aggregate views and predictions, they are
often unable to provide the nuanced interpretations that humans can. Despite recent advances in methods and techniques in natural language processing, challenges still remain. As new topics, entities and trends constantly emerge it is complex or even impossible for a statistical model to predict something that was never seen before, leading to poor accuracy scores in sets rich of novel data. Many machine learning models are trained on datasets labeled by humans, which is an expensive and slow process. Labeling news stories is a tedious task and this manual effort can quickly become irrelevant, due to the timely nature of news.

1.2 Purpose

This thesis presents a novel method of human-machine collaboration designed to improve both humans’ and computers’ understanding of the news. It describes the algorithms and design principles of an interactive human-in-the-loop platform for news exploration.

This new method includes a set of machine learning classifiers with an online feedback mechanism from the user. The classifiers are trained to label the topics of each story, and to predict how positive, subjective or trending a story is. We visualize those labels and present them to the user for each story they read. The interaction with Panorama encourages the user to give better labels to stories, which in turn are fed back to the classifiers to improve their performance.

We apply principles of data visualization and user experience to address the problem of information overload. The graphical representation allows exploration of underlying patterns in the news, enhances perceivability of data, and boosts the user ability to process and understand this complex ecosystem.

As a proof of concept, we present Panorama, an interface for browsing through thousands of news stories from the past day. Panorama enables users to organize, filter and sort the information. While exploring news stories, users also label the stories as they read, giving valuable input to machine learning models that analyze the data.

As machine learning becomes increasingly important in helping us navigate complex media sets, the Panorama interface demonstrates how data visualization and interaction
plays a critical role in helping humans understand and improve a machine learning based system.

### 1.3 Contribution

This thesis presents a novel interaction method and associated user experience for exploring and reading recent news stories, which allows the user to control the content aggregation and filtering. Revealing the underlying mechanism that filters the stories the user reads addresses issues of "filter bubbles", allowing users to understand and control how content is delivered to them.

This thesis also introduces a software architecture and a set of design principles for systems that enable human-machine collaboration through interactive data visualizations. It combines state of the art deep learning tools with human annotated data to enable the ongoing training of new language models and classifiers, while collecting input data submitted by engaged users. This annotated data helps improve the language models and will eventually contribute to a better understanding of the large scale unstructured data that is news. This method could be applied to other types of unstructured data as well, such as medical records, collections of music, books, etc.

### 1.4 Overview of the Thesis

The next section situates this work in relation to historical and contemporary themes related to News analysis, aggregation and visualization. This section also introduces prior work in creating innovative user interfaces for "human in the loop" models. Section 3 describes the two systems that provide news data to the Panorama system, and the preliminary work that led to developing Panorama. Section 4 is an overview of the implementation of Panorama and the algorithms within the system. Section 5 provides a detailed walk through of the user experience and user interface of the tool. Section 6 describes how Panorama was tested and presents an evaluation of the tool along qualitative dimensions. Finally, section 7 concludes the thesis with a discussion of possible future work.
Chapter 2

Background and Related Work

This work is based on existing and emerging work in four fields:

- Natural Language Processing and News analysis
- Interactive Machine Learning and Collective Intelligence
- Information Visualization
- Content Aggregation

This section provides brief historical background for each field, tracing their origins to the present day.

2.1 NLP and News Analysis

Natural language processing (NLP) is a field of computer science concerned with programming computers to find meaning in large natural language corpora.

Research in NLP has started in 1950, when Alan Turing published his article "Computing Machinery and Intelligence" [1] that proposed what is now called the Turing test - judging a natural language conversation between a human and a machine as a criterion of intelligence. Up until the 1980s most NLP systems were based on complex sets of explicit rules designed by programmers. In the late 1980s to mid 90s, a statistical revolution took place, replacing explicit rules designed by programmers with statistical models that analyze the probabilities of words appearing in existing corpora to make predictions in
novel documents. These statistical and machine learning techniques dominate the field today, enabling applications like Google’s machine translation system [2, 3].

One effective way to create smarter natural language applications is to use supervised machine learning algorithms. These algorithms are designed to extrapolate rules from the knowledge base of texts they are given in order to apply those rules to unseen circumstances in the future [4]. Supervised machine learning algorithms take annotated text as input. This text has been augmented with metadata that helps the algorithm identify important elements and categories. Although powerful, this approach suffers from the bottleneck of creating a large number of annotated corpora, or bodies of text. Annotated corpora are expensive to make and they require a large investment of time from expert users [5].

In recent years, there has been substantial work on content analysis in news, and researchers are successfully utilizing machine learning models for tasks including topic detection and extraction from both text and broadcast news, news sentiment analysis and trending topics detection [6, 7, 8]. There are various public datasets for training and benchmarking news content analysis, such as the New York Times annotated corpus [9] (which I will discuss below) and the Reuters-21578 dataset [10]. Alas, the content of these corpora is limited to the media sources they include, and the languages, story types and metadata properties their creators chose to comprise.

This lack of annotated corpora is problematic for natural language programmers. It renders it nearly impossible to create sophisticated natural language applications. But it also has the effect that any natural language application cannot achieve a certain level of general intelligence due to gaps in knowledge related to those unrepresented fields. If we ever wish to have applications that are even more surface-level intelligent, it is incredibly important to have datasets across a very wide range of fields, news sources, or story types. As a result, creating natural language data sets with accurate metadata more cheaply and easily than existing methods is a problem worth addressing.
2.2 Interactive Machine Learning and Collective Intelligence

Interactive Machine Learning (IML) is an approach that couples human input and machine intelligence during the learning process, where the human is in the learning loop, observing the result of a machine learning model and providing input meant to improve the learning outcome.

Compared to traditional machine learning, where a classifier would be trained only once, in interactive machine learning the active learning algorithm first learns from a previously labeled collection of examples. It then classifies newly observed unlabeled examples. Some of the new examples are then manually labeled by a human. After the labeling is completed, the model is updated and the process is repeated for new incoming
examples.

Canonical applications of IML include scenarios involving humans interacting with robots to teach them how to perform certain tasks, humans helping virtual agents play computer games by giving them feedback on their performance or using a teaching curriculum to guide the machine learning.

There has been substantial work in exploring interactive machine learning techniques as applied to natural language processing [12], but this literature tends to focus on forming optimal data queries rather than on user behavior and participation. As the demand increases for computer programs capable of interacting with users through natural language, producing annotated datasets with which to train these programs is becoming highly essential.

Study in the fields of Collective Intelligence and Human Computer Interaction looks at new methods in which we can produce such datasets. Collective Intelligence is defined by Thomas W. Malone and Michael S. Bernstein as "groups of individuals acting collectively in ways that seem intelligent" [13]. In the last two decades, a new kind of Collective Intelligence has emerged: interconnected teams of people and computers, collectively doing intelligent things. This definition sets Collective Intelligence as an interdisciplinary research fields, with links to Biology, Economics, Political Science and Psychology, Computer Science, and Human Computer Interaction. Research in Human Computer Interaction (HCI) works to understand and design interactions between people and machines. It contributes to the collective intelligence by designing and studying the digital channels that give rise to it.

For example, Luis von Ahn’s “games with a purpose” (GWAP) paradigm [14] spawned a series of online games that incentivize voluntary contributions by providing tasks that people are already incentivized to do, and collecting data in the background. These tasks are generally easy for humans to accomplish but difficult or impossible for computers. Examples include tasks which rely on human experience, ability or common sense [15, 16]. For instance, the ESP game is a social game that encourages players to label images by pairing the players with partners who are also trying to label the same images. The game invites players to contribute labels, and its entertaining nature keeps players involved.
Likewise, Foldit is an online game in which players collaborate and compete to create accurate protein structure models. Foldit players found stable protein configurations that had eluded scientists for a decade [17]. Another example, also by von Ahn, is the website Duolingo [18, 19]. Duolingo provides a free platform for language learning, an inherently desirable activity for many people, while achieving text translation. Users translate article excerpts as language learning practice. Translations are corroborated across many users, and translated excerpts are combined into fully translated articles.

Essentially, GWAPs are appropriate when outsourcing of a computational task to humans can be achieved in a way that is enjoyable or gratifying for participants. A tool for annotating news stories while also exploring and reading the news may fall under this category.

2.3 Data Visualization

Data visualization is the graphical representation of numerical information. Early examples of data visualizations were seen as early as 1800 when William Playfair developed or improved upon nearly all the fundamental graphical designs, seeking to replace conventional tables of numbers with the systematic visual representations of his "linear arithmetic". Later on, John Tukey and Edward Tufte pushed the bounds of data visualization; Tukey with his new statistical approach of exploratory data analysis and Tufte with his book "The Visual Display of Quantitative Information" [20] which expanded data visualization for audiences beyond statisticians.

As technology progressed, so has data visualization. Technically sophisticated applications - including interactive visualizations - have broadened the field beyond statistics and made it a useful tool for journalism. Works from Martin Wattenberg and Fernanda Viégas, Ben Fry and others, as well as by journalists at institutions like The New York Times, helped shape the field and make visualization an accessible tool for everyone[21, 22, 23].

Information visualization attempts to harness quick perceptual systems for the purpose of processing information. Research shows that there are a number of cognitive benefits to visualization, such as increasing memory and processing resources available
and reducing search for information [24, 22]. Those benefits can be used to enhance the user experience and clearing the user’s mind to provide meaningful and valuable annotation to an interactive machine learning system.

Visualizing news stories is one challenge within the information visualization world. Below are some examples that inspired and informed the work presented in this thesis.

### 2.3.1 News Data Visualizations

**Newsmap**

Newsmap [25] is a treemap-based visualization of Google News. Using the treemap visualization, Newsmap indicates the importance of a story by the relative size of each story rectangle. Stories are also grouped by color to indicate various categories such as entertainment, health, technology or business.

![Newsmap data visualization](image)

Figure 2-2: Newsmap data visualization

The interface supports filtering based on geographic data for a select group of 23 countries, and also enables users to limit the visualization, based on keyword search through articles titles, descriptions, or sources.
Wall of Now

"Wall of Now" [26] is a project by Tomer Weller, developed on 2015 at the Viral Communications Group at MIT Media Lab. Wall of Now is a multi-dimensional media browser of recent news items. Every column in the wall represents a different type of entity: people, countries, states, companies and organizations. Each column contains the top-trending stories of that type in the last 24 hours. Pressing on an entity will reveal a stream of video that relates to that specific entity.

![Wall of Now](image)

Figure 2-3: Wall of Now

Unfiltered.News

"Unfiltered.News" [27] is a project by the Google Jigsaw team, in collaboration with Google News, Google Big Picture team, Instrument, and Periscopic. It is a data visualization that uses Google News data to show which daily news topics are being published in every region. Headlines for these topics can be viewed from around the world, with translations provided in 40 languages. It is built to allow internet users to discover which stories are being covered in certain locations, how different countries cover stories differently, and how issue coverage changes over time. "Unfiltered.News" allows its users to find news
beyond their border, see which regions are reporting on a topic, read topics and headlines in any language and explore trends over time [9, 28].

Figure 2-4: Unfiltered.News

2.4 News Feeds and Content Aggregators

Magazines, newspapers, television shows and online news feeds are a compilation of articles, but they also go beyond mere compilation. Journalists not only create news stories, but also organize them in a way that makes sense of each story in relation to other stories in the news product. The news product as a whole magnifies each story by placing it in context both in relation to other items within the news product as well as against all items not deemed worthy for inclusion in the product. Therefore, journalism possesses presentational authority, or the ability to generate meaning through the purposive ordering of news items [29].

Many top online news sites are connected with traditional media outlets. They use much of the graphical vocabulary of traditional media. The most important stories are signaled through the use of graphics and different-sized headline fonts, especially with breaking stories. Placement toward the top of the page connotes importance and fresh-
ness, while users scroll down for older, less central stories.

In 2000s there has been a rise in search engines and content aggregators that index headline links across news sites. Some of the more popular sites are Google News [30], Yahoo! News [31] and Rocket News [32]. The sites vary in the number of sources they cull from, the length of time they archive stories, and the amount of personalization they allow.

Recently, with the rise of social media websites, many adults turn to their social networks to get news. Facebook is an important source of website referrals for many news outlets. In 2012, 49% of U.S. adults reported seeing news on social media. According to a 2016 survey by Pew Research Center, 62% of U.S. adults get news on social media and 18% do so often [33, 34, 35].

Google News and the Facebook News Feed are two interesting and important examples of modern news aggregation tools.

**Google News**

Google News [30] is a web-based news portal launched in 2002. It is a computer-generated news site that aggregates headlines from news sources worldwide, groups similar stories together and displays them according to each reader’s personalized interests. Google News offers links to several articles on every story. The articles selection is fully automated, and features stories based on how often and on what sites a story appears online as well as factors like story freshness, location, relevance and diversity [36].

**Facebook News Feed**

The Facebook News Feed is a constantly updating list of updates and stories which appears on every user’s homepage. The News Feed surfaces recent friend activity such as profile changes, shared links, comments, and posted notes. An important feature of News Feed is that it allows for passive information sharing, where users can broadcast an action to their entire network of friends through News Feed (instead of active sharing methods such as a private message, where a user picks a specific recipient or recipients). These stories are aggregated and filtered through an algorithm that ranks stories based on social and
content-based features, then displayed to the friends along with stories from other users in their networks [37]. According to Facebook, the stories shown in the News Feed are influenced by the connections and activity of the individual user on the site. Facebook's stated intent is to help the user see stories that interest her from friends she interacts with the most. The number of Comments and Likes a post receives and what kind of story it is (ex: photo, video, status update) likely influence whether or not it appears in a given News Feed [38].

2.4.1 The Power of Aggregators

The power of the media to highlight or ignore particular stories is widely acknowledged and criticized in media studies work. This is true for both the traditional and modern news
Figure 2-6: Facebook Newsfeed

sources described above. In 1922 Walter Lippman, newspaper columnist, first posed the idea that the mass media shapes public perception with images. Lippman’s notion, based on the public’s limited first-hand knowledge of the real world, created the foundation for what has come to be known as agenda-setting. Bernard Cohen’s statement in 1963 predicted that "the press may not be successful much of the time in telling people what to think, but it is stunningly successful in telling its readers what to think about". Agenda-setting theory, formalized by McCombs and Shaw [39] describes the "ability [of the news media] to influence the salience of topics on the public agenda" [40, 41]. This is especially true today with aggregators, which aren’t taking a political stance with what they write, but are explicitly telling you what’s worth your attention. In choosing and displaying news, editors, newsroom, staff, broadcasters and aggregators all play an important role in shaping political reality.

As web companies strive to tailor their services, including news and search results,
to our personal tastes and use filtering and personalization mechanisms that are hidden from the user, they create a "Filter Bubble" that prevents different sides of a debate from engaging with one another. Anthony Smith, in his book about the newspaper revolution of the 1980s wrote:

"It is now becoming much easier to supply them [the news audience] with a stream of the facts they elect to receive, rather than those that somebody elects to give them, and this switch in the balance of sovereignty over the content of available information has profound implications for the future of citizenship and for the evolution of the sphere of privacy, of identity." [42, p.311]

Eli Pariser [43] argues that this personalization has unintended consequences that will ultimately prove to be bad for us and bad for democracy, causing us to receive mainly news that is pleasant, familiar and confirms our beliefs. Since these filters are invisible, we won’t know what is being hidden from us. Our past interests will determine what we are exposed to in the future, leaving less room for the unexpected encounters that spark creativity, innovation and the democratic exchange of ideas. Cass Sunstein warns against "information cocoons" and "echo chambers," wherein people avoid the news and opinions that they don’t want to hear [44]. Christian Sandvig et al. describe the Facebook News Feed curation algorithms as black boxes that prevent the users from understanding the details of their functionality [45, 46]. David Lamb mentioned that "it is important to be aware that the opinions and stories you see on Facebook are strongly influenced by what Facebook’s news feed algorithm thinks you should see" [47] [48] [49].

There is a number of alternative aggregators that are trying to create a more balanced views of the news, or to create conversation about what an ideal balance might be. Below are selected examples of alternative aggregators.

**Blue Feed, Red Feed**

"Blue Feed, Red Feed" [50] is a data visualization produced by the Wall Street Journal and presented on the paper’s website. To demonstrate how reality may differ for different
Facebook users, this interactive interface shows two feeds, one "blue" and the other "red." If a source appears in the red feed, a majority of the articles shared from the source were classified as "very conservatively aligned" in a large 2015 Facebook study \(^1\). For the blue feed, a majority of each source's articles was aligned "very liberal". These give a rare side-by-side look at real conversations from different perspectives. The user can select a topic from a list of topics that includes "Affordable Care Act", "Guns", "ISIS" and more. Clicking on a topic will present the two matching feeds. This is a great example and a data visualization for creating a public discussion around filter bubbles and different perspectives on the same story. The list of topics to select from is manually curated and does not update as new political issues rise. Thereby, it does not replace a user's personal Facebook feed or news aggregator.

\[ \text{Blue Feed, Red Feed} \]

See Liberal Facebook and Conservative Facebook. Side by side.

\[ \text{Figure 2-7: Blue Feed, Red Feed} \]

\(^1\)In this study by Lada Adamic et al.[51] the researchers examined how 10.1 million U.S. Facebook users interact with socially shared news. As part of the study, they obtain a measure of content alignment for news stories by averaging the ideological affiliation of each user who shared the article. These scores, averaged over websites, capture key differences in well-known ideologically-aligned media sources: FoxNews.com is aligned with conservatives \((A_s = +.80)\), while the HuffingtonPost.com is aligned with liberals \((A_s = -0.65)\).
Perspecs News

"Perspecs News" [52] is a web page and an app that offers readers three polarized opinions of the same story for each story presented on the interface. The opinion categorization can vary. For political stories this could be in the form of 'left', 'background' or 'right'. For review items the categories could be 'negative', 'neutral' or 'positive'. Perspecs News is manually curated, and publishes once a day. Due to the intensive manual effort required, they publish only a handful of stories each day, and do not cover all news stories, thus can not be used as a user’s single news source.

Sidelines and BALANCE

Sidelines [53] and BALANCE [54, 55] are both projects from the School of Information at University of Michigan led by Professor Paul Resnick. These works investigate diversity-aware news recommendation services to improve people’s news reading experiences. Sidelines is an algorithm for increasing diversity in news and opinion aggregators. It does so in aggregators that rely on votes by temporarily suppressing a voter’s preferences after a preferred item has been selected. BALANCE is a system that classifies news and opinion articles as conservative or liberal using data from social news aggregator Digg, which
allows readers to vote for articles they like. The system combines the patterns of readers who voted that they "liked" certain articles and automatically classifies news and opinion articles as liberal or conservative.

**AllSides News**

"AllSides News"[56] is also a news aggregation website. Similar to the concepts presented in Sidelines and BALANCE, "AllSides News" presents news stories from all sides of the political map, in order to prevent polarization. "AllSides" has a patented bias rating and bias detection model which is powered by a combination of wisdom-of-the-crowd technology and statistical research and methodologies. They encourage the user to submit her opinion regarding a story bias and political leaning and use this data to present biased story and stories across the political map. Users of AllSides News can read, organize, submit, and share news and issues from multiple perspectives. They can also review and contribute to bias ratings and get weekly news highlights.
2.5 Synthesis

The work presented in this thesis was created in dialog with and in response to the history and recent work in NLP, news content analysis, data visualization, interactive machine learning and news presentation and aggregation. The proposed interface, Panorama, articulates a vision for a system enabling human-machine collaboration through interactive data visualizations, allowing for a better understanding of information by both people and computers.
Chapter 3

Data Sources and Preliminary Work

This section describes the preliminary work done prior to developing Panorama and the two data sources that provide the stream of news stories to Panorama.

3.1 Data Sources

Panorama uses two systems to retrieve thousands of news stories each day. Media Cloud, a tool for studying digital news media, developed at the MIT Media Lab and at the Berkman Center at Harvard University. And Superglue, developed at the MIT Media Lab to allow analysis and software development around broadcast media content.

3.1.1 Media Cloud

Media Cloud is an open source platform for studying media ecosystems. It is a joint project of the MIT Center for Civic Media and the Berkman Center for Internet and Society at Harvard University [57, 58]. Media Cloud retrieves stories from a set of 46,000 active crawled digital media sources - newspapers, other established news organizations and blogs, that generate 440,000 stories a day. It also includes a set of tools for analyzing this data, such as word count extractions, dashboards to search and compare specific topics. Media Cloud users are able to query the massive dataset and request stories that were published within a specific time frame, that include or exclude specific terms, that were
written in certain languages and by certain sources and more.

Media Cloud is organized around collections, sets of media sources representing media ecosystems around the world. To identify appropriate sources, Media Cloud uses a combination of automated search and discovery, identified lists of influential sources, and expert input from journalists and media practitioners.

For Panorama, we use the "U.S. Mainstream Media collection", a set of the Top U.S. mainstream media sources based on Google Ad Planner’s measure of unique monthly users, created in October 2010. It includes 24 Media sources \(^1\). The U.S. Mainstream Media collections collects 107,000-180,000 sentences a day and 3000-8000 stories in a 24 hours time span.

### 3.1.2 Superglue

Superglue \([59]\) is a project built in the Viral Communication group at MIT Media Lab. Superglue is a "digestion system" and metadata generator for mass media that provides information about broadcast media to several Viral Communications projects. Superglue is a set of analysis modules that annotate 14 DirecTV live news broadcast channels. The video is archived and synchronized with the analysis. The system provides named-entity extraction, audio expression markers, face detectors, scene and edit point markers, excitement trackers, and thumbnail summarization. It is used to organize material for presentation, analysis, and summarization. Superglue records and analyzes about 200 videos a day, that generate 600-800 news story segments in a 24-hour time frame.

### 3.1.3 Feeding Back to the Data Sources

As the Panorama classifiers improve, they could be added back into the Superglue and Media Cloud systems, becoming part of those research tools and providing additional insights and knowledge to the researchers using them. The Topic detection model built and trained for the purpose of this thesis work is already deployed in both Media Cloud

---

and Superglue. As this work progress, it will extend the capabilities of the Media Cloud and Superglue systems and the projects that depend on them.

3.2 Preliminary Work

During my time as a graduate student at the Viral Communications group at the Media Lab, I developed interfaces exploring the visual representation and exploration of news, as well as various machine learning algorithms to better analyze and automatically process and make sense of news stories. These explorations informed the latter design of Panorama. The following section describe this earlier work.
3.2.1 News Graph

"News Graph" was my first iteration in exploring how to graphically represent and interact with large-scale news data, built during April 2016.

![News Graph](image)

Figure 3-2: News Graph

By analyzing the words that are said in the videos, extracting entities that appear, and finding the connections between them, "News Graph" generates a mapping between entities identified in video segments. Each connection represents two entities that were mentioned in the same video segment. One video segment can be included in multiple connections.

News Graph interactive graph is rendered using Processing.js [60]. Each circle in the graph represents an entity, the size of the circle corresponds to the number of mentions of that entity in the Superglue database in the past 24 hours. The color of the circles represents different types of entities (people, countries, companies, states, etc.) When hovering on a circle, all the related entities and connections are highlighted as shown in
Clicking on a circle will pull it towards the video player and will start playing a video mentioning this entity. It will also send all the unrelated entities out of the screen and will bring in related entities that were not seen on the landing page. This is now the primary entity. Clicking on a second circle, making it the secondary entity, will also send it over to the video player, and will start playing a video segment mentioning the two selected entities.

Once two entities have been selected, clicking on an additional circle on the left side will replace it with the current secondary entity. Clicking on the secondary entity will set it as the primary entity, and show all of its connections. Clicking on the first circle selected will take the user back to the initial view.

This interaction, coupled with a particle physics and springs animation, gamifies the experience of exploring the current news. By playing with the different entities, moving
from one connection to another, the user can get updates about the entities he is interested in, while also discovering new connections that were not necessarily obvious.

3.2.2 Perspectives

"Perspectives" was my second take on exploring various uses for the Superglue system, understanding propaganda and perspectives in news presentation, built during the summer of 2016. This time I used clustering algorithms on the video story segments, in order to group news stories into meaningful groups and automatically generate a comparative news analysis view. The goal was to give a viewer different points of view on a specific story.

To generate the Perspectives view, the algorithm collects all stories that were broadcast in a 24-hours time frame, averaging 700 segments from 14 broadcast sources. We run Agglomerative Hierarchical Clustering on the text of all those stories, to find groups of stories that are similar. This process outputs clusters of video segments, allowing us to sort the stories in each group by source, length, broadcast date, or keyword. Each cluster is then represented by the words most mentioned in all the stories within that cluster.

The top row in the interface is a series of tabs with words in each tab. The tabs repre-
sent the clusters that had the most stories in them, going from left to right. The words in each tab are the words that appeared the most in all the stories in that cluster. Keyword sizing is determined by the total appearance of that word in all videos from the past 24 hours. Within each cluster, the stories are grouped according to their source, enabling the user to see the different perspective regarding the same story, presented by different media sources.

Users interact with Perspectives by selecting a story of interest. When a story is selected all the videos related to that cluster, grouped by media source, appear on the screen under the stories menu and start to play with audio muted. A click on a specific video will turn on the sound of that video. Above each video appears the channel name and the air date of that video. Beneath each video there is a row of thumbnails, representing all the videos that a specific channel has broadcast regarding the selected story. The videos thumbnails are sorted by time - from oldest to newest, but they all date from the past 24 hours. As soon as a video ends, the next video in the list will start playing. The user can also click on any of the thumbnails to start playing a specific piece.

By tracking published stories and clustering those that use similar language, Perspectives helps identify which stories are reported and which news outlets have introduced new frames into the public debate.

This experimentation led me to try and find better ways to model, sort, analyze and visualize news stories, which led to the Panorama system described here.
Chapter 4

Implementation Details

4.1 Overview of the Implementation

This section explains in detail the implementation of news analysis models that fuel Panorama. Panorama is powered by a set of language models that were trained using a variety of manually labeled datasets. For each model, the input is the full text of a news story and the output is a score between -1 and 1. For example, in the sentiment analysis model explained below, the model will output 1 if the story is completely positive, and -1 if the story is completely negative. If the story is neutral the model will output 0, and it might output 0.6 if the story is somewhat positive. In the case of the topics model - the output is a list of topics and a score for each topic representing its relevance to the story, for example [Finance - 0.7, Politics - 0.3, Animals - 0.001].

The first section in this chapter describes the word2vec model that is used in each of the language models to represent stories as matrices.

4.2 Modeling News Stories with Word2vec

Word2vec was developed by Tomas Mikolov at Google research in 2013 [61, 62].

Over the years, research in Natural Language Processing have shown various ways of representing words, sentences and documents as vectors, to be used by statistical models
and machine learning algorithms. Representing text as vector is less straightforward and intuitive than representing images as vectors since images are represented as matrices of a color number for each pixel, but matrix representation allows powerful mathematical techniques to be used on documents. Texts are often represented in NLP systems via Dictionary, "Bag of Words" and "TF-IDF" methods. In a dictionary, each word is mapped to a specific score; for example, when using dictionary for sentiment analysis, we can use a dictionary of good and bad words. Each word within the sentence has a score, typically +1 for positive sentiment and -1 for negative. We simply add up the scores of all the words in the sentence to get a total sentiment score. Clearly, this process has many limitations, the most important being that it neglects context and surrounding words. For example, in our simple model the phrase "not good" may be classified as 0 sentiment, given "not" has a score of -1 and "good" a score of +1. A human would likely classify "not good" as negative, despite the presence of "good".

Another common method is to treat text as "Bag of Words". We treat each text as a 1 by N vector, where N is the size of our vocabulary. Each column is a word, and the value is the number of times a word appears. For example, the phrase "bag of bag of words" might be encoded as [2, 2, 1]. TF-IDF, which stands for "Term Frequency Inverse Document Frequency", is very similar to "Bag of Words". Instead of encoding just word counts we multiply each word count (term frequency) with its inverse document frequency - a number that represents whether the term is common or rare across all documents in our corpus. Vectors extracted using the "Bag of Words" or the TF-IDF methods could then be fed into a machine learning algorithm for classification, such as logistic regression or Support Vector Machine, to predict sentiment on unseen data, or can be used to calculate scores such as cosine similarity. In cosine similarity, for example, we would like the answer the question how similar are two documents, and to do so, we take the cosine of the angle between the two vectors representing those documents. The angle between vectors A and B is a measure of how similar the sets are, and taking the cosine of that angle is a simple way to scale the value to be between 0 and 1 for angles between 0 and 90 degrees. If two documents are identical in word frequency - both mention Obama 23 times, Libya 5 times and basketball twice - they score a 1. If they’ve got no words in common, they score
a zero. While this is an improvement over the previous method, it still ignores context, and the size of the data increases with the size of the vocabulary.

Vector Space Models (VSM) represent words in a continuous vector space where semantically similar words are mapped to nearby points. VSMs have a long, rich history in NLP, but all methods depend in some way or another on the Distributional Hypothesis, which states that words that appear in the same contexts share semantic meaning. Word2vec is a particularly computationally-efficient predictive model for learning word embeddings (word vectors) from raw text. The word2vec method captures the context of words, while at the same time reducing the size of the data. It comes in two flavors, Continuous Bag of Words (CBOW) and Skip-gram. In the CBOW method, the goal is to predict a word given the surrounding words. Skip-gram is the converse: we want to predict a window of words given a single word. Both methods use artificial neural networks (described below) as their classification algorithm. Initially, each word in the vocabulary is a random N-dimensional vector. During training, the algorithm learns the optimal vector for each word using the CBOW or Skip-gram method.

These word vectors captures the context of surrounding words. This can be seen by using basic algebra to find word relations (i.e. "king" - "man" + "woman" = "queen")[63]. These word vectors can be fed into a classification algorithm. The advantage is that we now have some word context, and our feature space is much lower (typically ~300 as opposed to ~35,000, which is the size of the vocabulary used in Superglue to previously create word vectors). Since texts have varying length, one might take the average of all word vectors as the input to a classification algorithm to classify whole text documents. Another way is to take only the vectors of the first N words, which is the method used in Panorama. We always look at the first 200 words in a story and pad shorter documents in 0 vectors, so each document is an equally sized matrix of 200 by 300 (200 words, 300 dimensions for each word).

Google has published vectors pre-trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases, trained using the CWOB and Skip-gram methods [61]. This model is used to represent all documents in the Panorama system. With very little tuning of the param-
eters, this model achieves excellent results on multiple benchmarks, suggesting that the pre-trained vectors are 'universal' feature extractors that can be utilized for various classification tasks. We found them good enough for the data represented in Panorama, taking into account that the vectors were trained on news stories from a large variety of sources, very similar to the data processed within Panorama.

4.3 Multi-label Classifier For News Topic Detection

The goal in training a multi-label topic classifier was to be able to determine what topics a given news story is about. This would allow us to know what topics are most talked about in the news, what topics each source is talking about, how topics co-occur and how stories topically relate to one another.

4.3.1 NYT Annotated Corpus

To train a topic classifier for news stories, we used the New York Times annotated corpus. We chose this dataset due to its large scale (1.8 million articles), high quality (mostly manually labeled) and variety (all of the NYT stories published over a period of 20 years, including sports, finance, opinion and more). The NYT annotated corpus is a corpus drawn from the historical archive of the New York Times and it includes metadata provided by the New York Times Newsroom, the New York Times Indexing Service and the online production staff at nytimes.com. This corpus contains nearly every article published in the New York Times between January 01, 1987 and June 19th, 2007. Articles are tagged for persons, places, organizations, titles and topics using a controlled vocabulary that is applied consistently across articles [9].

For the topic detection model, we used the descriptors fields and the taxonomic classifiers field (also referred to as "taxonomies"). There are three different fields containing descriptors, and one field with taxonomies. The following are the descriptions of each one of those fields, as written in the NYT annotated corpus overview document.
**Descriptors**

The three types of descriptors in the corpus are ‘descriptors’, ‘general online descriptors’ and ‘online descriptors’. The ‘descriptors’ field is the largest collection, totaling more than 30,000 unique items. These are descriptive terms drawn from a normalized controlled vocabulary corresponding to subjects mentioned in the article. ‘Descriptors’ tags are hand-assigned by a team of library scientists. The ‘online descriptors’ are descriptors from the same normalized vocabulary, with the difference being that these tags are algorithmically assigned and manually verified by the nytimes.com production staff. The ‘general online descriptors’ are generated in the same method as the ‘online descriptors’ but are at a higher level of generality than the other tags associated with the article. Examples include:

- Economic Conditions and Trends
- Surfing
- Venice Biennale
- Ranches

**Taxonomic Classifiers**

This field specifies a list of taxonomic classifiers that place this article into a hierarchy of articles. The individual terms of each taxonomic classifier are separated with a ‘/’ character. These tags are algorithmically assigned and manually verified by nytimes.com production staff. Examples include:

- Top/Features/Travel/Guides/Destinations/North America/United States/Arizona
- Top/News/U.S./Rockies

**Corpus Statistics**

Table 4.1 include information about the descriptors and taxonomies within the NYT annotated corpus. The corpus includes 26,048 unique descriptors (combined online, general and regular, and compared when all lowercase). Out of the 26,048 unique descriptors, 20,593 appear in less than 10 articles (leaving 5,454 descriptors that appear in 10 articles
4.3.2 Multi-Label Descriptors Classifier for News Stories

Using the New York Times Annotated corpus, we trained a number of Convolutional Neural Networks as multi label classifiers for classifying news stories to different topics (i.e. descriptors and taxonomies). The code for training the classifiers is based on Magpie - an open source deep learning tool for multi-label text classification. It learns the training corpus to assign labels to arbitrary text and can be used to predict those labels on unknown data. Magpie has been developed at CERN to assign subject categories to High Energy Physics abstracts and extract keywords from them [64, 65].

Our model is a convolutional Neural Network, implemented using Keras, which is a high-level neural networks API, written in Python and capable of running on top of either TensorFlow or Theano. In order to train the classifiers, all 1,800,000 documents from the NYT corpus were parsed and uploaded to a Mongo database. They were shuffled and split to 85% training set and 15% testing set.

5 different models were trained using different sets of labels:

- "600 Descriptors" : 600 most used descriptors
Convolutional Neural Network and model Parameters

Neural networks is a machine learning technique which is inspired by the brain structure. It utilizes a network of learning units called neurons. These neurons learn how to transform input signals (e.g. picture of a cat or matrix representing a document) into corresponding output signals (e.g. the label "cat", or the label "politics and government"), forming the basis of automated recognition.

Recent advances in computer vision research as led to a new type of neural networks, called Convolution neural networks, initially used for solving the task of classifying images. The input for a convolutional neural network is a matrix, which for images is straightforward, since an image already takes the form of a matrix of pixels. Since we could transform documents into matrices using the word2vec method described above, we can now use convolutional neural networks to solve challenges in the NLP field as well [66, 67, 68].

In our case, we give the computer an array of numbers that represents a text document and it will output numbers that describe the probability of the document matching a particular topic (.80 for politics, .15 for finance, .05 for crime, etc).
There are four main steps in CNN: convolution, subsampling, activation and full connectedness. Our model is built of 5 sequential layers, a layer for phrases of one, two, three, four or five words (ngram 1-5). Each layer receives as input a 200 * 300 matrix representing a news story, runs the input through a 1D convolutional layer sized 256 * ngram_length and then performs max-pooling to output a 1 * 256 result. Layers use a Lecun Uniform init function and a Tanh activation function.

Figure 4-2: Convolutional Neural Networks for Sentence Classification (credit Kim, Y. (2014).[68])
The 5 sequential layers are then merged, flattened and the output is run through a dense layer with a sigmoid activation function. The loss function used is Binary Cross Entropy and a default Adam optimizer.

The model was trained over 5 epochs.

**Model Evaluation**

For training the model - a random set of 85% of the articles in the corpus was selected, reserving the rest as a test set (15%). On this test set the model loss is 0.01 (for each of the 5 models described above), and precision and recall were both above 0.97.

Since the corpus include only articles from the years 1987-2007, it was important to validate it on more recent articles as well. For this purpose, we used the NYT API, which allows developers to query the nytimes.com and retrieve articles and their semantic labels. We retrieved a random set of 3,000 articles from the New York Times website, published between 2010 to 2016. For each of the 3,000 articles, we extracted the story text and the story descriptors. This set included 180 descriptors that were not part of the original NYT corpus. This happens since new topics emerge as well as the indexing vocabulary slightly shifts over time (e.g. the "Greenhouse Effect" becomes "Global Warming"). The only processing done on to the retrieved descriptors was turning all of them to lowercase.
As a result, labels like "Israeli settlements" and "Israeli settlements (occupied territories)" were not considered the same label in this evaluation.

Out of the 5 models, the model with best results was the one trained on 600 descriptors. The evaluation results for the "600 Descriptors" set were:

**Average precision: 0.590366.** This score corresponds to the area under the precision-recall curve. For comparison, a study that compared various SVM and LDA algorithms for multi-label classification on news documents [69] presents scores between 0.449 - 0.612, evaluated on the original NYT corpus dataset.

**Label ranking loss: 0.057090.** This is the ranking loss which averages over the samples the number of label pairs that are incorrectly ordered, i.e. true labels have a lower score than false labels, weighted by the inverse number of false and true labels. The best achievable ranking loss is zero. The same study mentioned above present scores label ranking loss between 3.51 - 0.93 on the NYT dataset. These were calculated using sklearn.metrics functions [70].

A second evaluation included the conversion of the resulted labels frequency vector for each article to a 0-1 vector (every label with probability > 0.01 = 1 and the rest are 0). Results for this evaluation were recall of 0.82 and precision of 0.27 (using the sklearn.metrics.classification_report function).

**Media Cloud Topics Evaluation**

Rahul Bhargava from the Center for Civic Media, ran tests on Media Cloud topics as well, to get validation for the model for sources other the New York Times. The rest of this section explains his work.

The model was tested on 5 different Media Cloud topics - pre-selected sets of stories from the Media Cloud corpus on a given topic- and also on two weeks in March, one in March 2016 and the other in March 2017, with all the stories from the U.S Mainstream Media sources which a set of 24 media sources that includes the NYT but also 23 other sources.

The evaluation process was as follows: For each topic, all of the stories in that topic were fed into the model. For each story output that came from the model, only the labels
that returned with a score $> 0.75$ (higher than a 75% confidence) were kept. A sample of 10,000 sentences from all of the stories from that topic, and the following statistics are based on these samples. The sampling of sentences was used for efficiency reasons, but it should be a representative set of the stories in each topic. The models evaluated were the "Taxonomies" and the "600 Descriptors" models. The topics used and the total story count in each topic are:

- Oscars Topic: 9,980 stories total
- Common Core Topic (education policy in the US): 12,920 stories total
- Ebola Nigeria Topic: 18,848 stories total
- Nigeria Contraceptives Topic: 1,790 stories total
- India Rape Topic: 28,332 stories total

Figure 4-5 shows the percentages of stories in each topic that had at least one label with score $> 0.7$. It is notable that the Nigerian Contraceptives topic and the India Rape topic were difficult for the language models to assign tags to. It suggest that the models don’t work as well on international local sources, having trained on US media.

**Top Labels in Test Topics:**

The following figures 4-6 to 4-10 present the percentage of total stories with that label
at a confidence larger than 70%

Figure 4-6: Oscar topic

Figure 4-7: Common Core topic

54
Figure 4-8: Ebola Nigeria topic

Figure 4-9: Nigeria Contraceptives topic
In figures 4-11 and 4-12, Rahul used the language model to label all stories in the U.S. Mainstream Media collection from March 15-22 in both 2016 and 2017.

Rahul raised some interesting points when looking at those plots:

- The labels in the US Mainstream Media Collection were much more widely distributed, compared to the topics. This is expected, since the collection does mention many topics.
- "Basketball" is popping up on March due to the college tournament.
Figure 4-12: US MSM 2017

- "Medicine and Health" pop up in 2017 but not in 2016, probably due to many news about the Affordable Care Act during that time.
- "Elections" show up in 2016 but not 2017 (probably due the the November 2016 elections).

To conclude, the model evaluation on the Media Cloud topics returned topics we would expect. For example "Education and Schools" is at the top for the Common Core Topic and "Motion Pictures" and "Awards, Decorations and Honors" are top topics for stories about the Oscars. These result made us believe that this model is good enough for the current proof of concept presented here. Despite these promising findings, further rigor evaluation is required, comparing this model to existing research and state-of-the-art methods for detecting news topics, in order to determine the model’s general success rate.

4.4 Sentiment and Subjectivity Analysis Classifiers

4.4.1 Sentiment Analysis

Sentiment classification is a common application and a widely researched topic in Natural Language Processing methodologies [71, 72]. In sentiment analysis, the goal is to extract the emotional content in text and evaluate its sentiment or tone. Sentiment analysis can
be seen as a method to quantify qualitative data with a discrete sentiment score. This is challenging as sentiment is largely subjective, and many times, two people would disagree on the sentiment type and level that a specific text presents. Since computers don’t understand language the way humans do, and frequently tend to miss the underlying layer of interpretation, extracting a sentiment score has been an interesting challenge to solve for many years.

For training a sentiment analysis classifier I used the SAR14 dataset [73]. An independent score-associated review dataset of 233,600 movie reviews and their accompanying actual scores. The dataset consists of 167,378 user reviews connected to scores valued from 7 to 10, and 66,222 reviews linked to 1-4 rated ones. Using SAR14, I trained a Stochastic Gradient Descent Classifier (described below). The original dataset scores from \([0,10]\) were mapped to numbers from \([-1, 1]\) in order to fit into the panorama dataset and interface. When new labels come in, submitted by the user, they are entered in the \([-1, 1]\) scale and used for the re-training of the classifier.

4.4.2 Subjectivity Analysis

Subjectivity analysis focuses on distinguishing subjective words and texts that mark the presence of opinions and evaluations from objective words and texts, used to present factual information. The task involves the same complexities as in sentiment analysis. When performing subjectivity analysis, we want to determine whether a given text is subjective or objective. Consider the following review sentences:

- The camera is a good purchase.
- A camera is a good device for capturing photographs.

Both sentences contain opinion bearing the word good, yet, the first sentence is subjective and the second one is objective in nature. Thus, the target of subjectivity classifications is to determine whether a text is subjective or objective. However, classifying a sentence as either subjective or objective is a non-trivial task due to non-availability of training datasets. Annotated sets of subjective and objective sentences are difficult to obtain and requires lots of time consuming manual processing [74].

For training the subjectivity classifier we used the subjectivity dataset v1.0 by Pang and
Lee [75]. The dataset contains 5,000 subjective and 5,000 objective processed sentences.

This dataset comprise snippets of movie reviews from Rotten Tomatoes \(^1\) and plot summaries for movies from the Internet Movie Database\(^2\). The labels assume that all snippets from the Rotten Tomatoes pages are subjective, and all sentences from IMDb plot summaries are objective. This is mostly true, but plot summaries can occasionally contain subjective sentences that are mislabeled as objective. The labels were translated to \([-1, 1]\) scores, with \(-1\) representing an objective story and \(1\) symbols a subjective story. Due the the nature of the dataset, and the fact it only included the edge scores \((-1\) or \(1\)) the initial classifier mainly labels stories as completely objective \((-1)\) or completely subjective \((1)\) and the objectivity score distribution looks like Figure 4-13. As Panorama users submit more labels that map better to news stories and also have a finer level of granularity, the subjectivity analysis classifier is expected to improve.

4.4.3 Logistic Regression With Stochastic Gradient Descent

Both the sentiments and the subjectivity analysis classifiers are Logistic Regression Classifiers trained with Stochastic Gradient Descent.

**Logistic regression** is a statistical process for estimating the relationships among variables. For the purpose of the sentiment analysis and subjectivity analysis in Panorama, multinomial logistic regression is used. Multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems, i.e. with more than two possible discrete outcomes. That is, it is a model used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables (which may be real-valued, binary-valued, categorical-valued, etc.).

\(^1\)http://www.rottentomatoes.com/
\(^2\)http://www.imdb.com
Stochastic gradient descent (SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions. In other words, SGD tries to find minima or maxima through iteration.

SGD is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as Support Vector Machines and Logistic Regression. In machine learning, we can use stochastic gradient descent to evaluate and update the coefficients in every iteration in order to minimize the error of a model on our training data. This optimization algorithm operates by examining each training instance one at a time. The model makes a prediction for each training instance, the error is calculated and the model is updated in order to reduce the error for the next prediction. SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. It is widely used because it is known to be very efficient and easy to tune.

Since SGD has the ability to learn incrementally from mini-batches, it was a great candidate for implementing the online learning mechanism needed in Panorama. When new labels come in, they are treated as new mini-batches of labeled data. Feedback from the Panorama worker serves as a new partial-fit of the classifier. SGD is also not affected by batch size which was extremely useful for our use case, where the number of input labels available is dependent on the user’s behavior and thus not predictable.

4.5 Left-Right Leaning Sources

For determining a political orientation of a story, we used the results of a recent research done by Ethan Zuckerman, Yochai Benkler, Hal Roberts and the Media Cloud team, that tried to determine partisanship in media sources [76]. The study examined over 1.25 million stories published online between April 1, 2015 and November 7, 2016 (Election Day). Using Media Cloud, the team analyzed hyperlinking patterns, social media sharing patterns on Facebook and Twitter, and topic and language patterns in the content of the 1.25 million stories, published by 25,000 sources.
As part of the study, the authors analyzed the share of media sources stories tweeted by users who also retweeted either Clinton or Trump during the Elections. This analysis therefore reflects the attention patterns of audiences. Using this analysis, the Media Cloud team divided the sources into partisan quintiles based on relative number of tweets of election story urls by Clinton vs. Trump retweeters. These quantiles categorize each Media Cloud source into one of [Left, Center-Left, Center, Center-Right, Right].

Using those labels as a baseline, Panorama labels news stories with a political leaning score. Initially, each story receives a score between -1 to 1, based on the story’s source label in the Media Cloud API (-1 is left, 0 is center and 1 is right). As the Panorama system collects more human annotated labels, the language of the story is weighted as well in the process of determining the political leaning of a specific story. Initial source labels still count, so a story from a source labeled “Right” will not be considered “Left” but it might be considered “Center” if the story talks about a non-political topic and does not present a political partisanship in its language or subjects.

4.6 Trend Detection

In order to decide whether a story is trending or ongoing, Panorama uses already detected Trends, that are published through Twitter API and Google News. As part of the Panorama worker (explained below), every 6 hours we query the Twitter API for trending topics in the United States, as well as the Google News website for the trending topics that appear on the navigation bar.

The Twitter API "GET trends/place" [77] resource returns the top 50 trending topics for a specific WOEID (Where On Earth IDentifier). A WOEID is a unique 32-bit reference identifier, that identifies any feature on Earth. In our case, we use the WOEID of United States. Panorama inspects the available trends and uses the name of each trending topic to understand what people are talking about. Currently, Panorama only uses trends names that do not start with a hashtag (#), since they are more readable and more likely to represent a news trending topic.

The retrieved trends, from both Google and Twitter, are parsed and uploaded to a
database, together with a timestamp of when a trend was last updated, and a count of how many times a trend has appeared so far.

When a new story is retrieved and added to the Panorama database, the trending score is determined as described in algorithm 1

**Algorithm 1:** Get a story trending score

1. Retrieve all trends from the database
2. Sort trends from newest to oldest
3. for each trend do
4.   if the trend name appears in the story then
5.     calculate a trending score based on "freshness" of the trend
6.     return the trending score
7. end
8. end
9. return 1 (ongoing story)

This method returns a score between -1 to 1 of trending-ongoing, where a -1 scores
represents a fresh, trending story and 1 is given to an ongoing story. The specific trend that determined a story score is also saved to the database.

When new human-annotated labels are parsed, the original trending score of the labeled story is compared with the new score given by the user. If the story was labeled as more trending - the story text is scanned to extract named entities (using Stanford NER model [78]). Each entity that appeared more than 3 times in that story is added to the trending database. If the score is lower, the trending topic that determined that trending score is marked down, reducing that term score in the next iteration of the algorithm.

Trend detection is a popular research topic and there is a number of suggested methods in order to detect trends in a large dated text corpora, many of them based on statistical models and machine learning algorithms [79, 80, 81]. Due to the scope of this thesis, and to the fact that the panorama database does not save full text of stories, we decided not to use usual machine learning algorithms or text mining techniques. Improved trend detection is future work for Panorama, as is exploring new methods and patterns of detecting trending and emerging topics with a human in the loop interface.
4.7 Panorama Database Structure

For managing all the data behind Panorama, an SQL-type database is used. SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. A relational database system contains one or more objects called tables. The data or information for the database are stored in these tables. Tables are uniquely identified by their names and are comprised of columns and rows. Columns contain the column name, data type, and any other attributes for the column. Rows contain the records or data for the columns.

4.7.1 Panorama database tables

The Panorama database contains a number of tables, some are static and never change, and some are constantly updating. This section describes each table.

Stories

Each row in the stories table represents one story with the fields specified in tables 4.2 and 4.3.

<table>
<thead>
<tr>
<th>storyId</th>
<th>MediaId</th>
<th>collectDate</th>
<th>publishDate</th>
<th>mediaName</th>
<th>MediaUrl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique id of the story</td>
<td>Unique id of the story source</td>
<td>Date the story collected</td>
<td>Date the story was published</td>
<td>Name of media source</td>
<td>Url of media source</td>
</tr>
</tbody>
</table>

Table 4.2: Stories table part 1

<table>
<thead>
<tr>
<th>title</th>
<th>leftRight</th>
<th>posNeg</th>
<th>trend</th>
<th>objective</th>
<th>image</th>
<th>url</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story title</td>
<td>The left right score given by the classier</td>
<td>The sentiment analysis score</td>
<td>The trending score</td>
<td>The objectivity score</td>
<td>Url of story image if exists</td>
<td>Story url</td>
</tr>
</tbody>
</table>

Table 4.3: Stories table part 2
Descriptors

This is a static table that never changes, it contains all 600 descriptors that the multi-label classifier might use. It only contains one column, called "id". This is a unique string with a descriptor name, for example: "abortion", "air pollution", "commuting".

DescriptorsResults

The DescriptorsResults table contains all the results generated by the multi-label topic classifier. Each row represents one results and contains the columns specified in table 4.4.

<table>
<thead>
<tr>
<th>id</th>
<th>descriptorId</th>
<th>storyId</th>
<th>score</th>
<th>createdAt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique id for that result</td>
<td>The descriptor id as it appears in the Descriptors table</td>
<td>The story id, that matches the ids in the Stories table</td>
<td>A number between 0-1, representing the relevance of the descriptor to the story, as returned from the classifier</td>
<td>The date and time of this result creation</td>
</tr>
</tbody>
</table>

Table 4.4: DescriptorsResults table

Connections

The Connections table represents the connections between the various descriptors, it is re-calculated by the Panorama Worker, every 6 hours, using an SQL query. Table 4.5 includes the connections columns.

<table>
<thead>
<tr>
<th>id</th>
<th>origin</th>
<th>dest</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique id for that connection</td>
<td>The descriptor id of the first descriptor in the connection (for example &quot;abortion&quot;)</td>
<td>The descriptor id of the second descriptor in the connection (for example &quot;babies&quot;)</td>
<td>An integer representing the number of stories that include both the origin and the dest descriptors</td>
</tr>
</tbody>
</table>

Table 4.5: Connections table
Trends

The Trends table is updated by the Panorama Worker every 6 hours. This Table contains all trends that were collected by the worker, and for each trend, collecting the most recent date it was mentioned.

Labels

The Labels table contains all labels that were submitted by Panorama users. Columns specified in table 4.6 and 4.7

<table>
<thead>
<tr>
<th>id</th>
<th>storyId</th>
<th>leftRight</th>
<th>posNeg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique id for that label</td>
<td>The id of the story that was labeled</td>
<td>The left right score given</td>
<td>The positive/ negative score</td>
</tr>
</tbody>
</table>

Table 4.6: Labels table part 1

<table>
<thead>
<tr>
<th>trend</th>
<th>objective</th>
<th>isUsed</th>
<th>sessionId</th>
</tr>
</thead>
<tbody>
<tr>
<td>The trending score</td>
<td>The objectivity score</td>
<td>Whether this label was already used for retraining or not</td>
<td>Unique id of the session this label was submitted at</td>
</tr>
</tbody>
</table>

Table 4.7: Labels table part 2

4.8 Data Aggregation Architecture

The Panorama system relies on a worker script, separate from the interface architecture. The worker code is written in Python and available on Github. It is deployed using a cron job that runs every 6 hours. The worker workflow is described below in algorithms

3https://github.com/jasrub/panorama-worker
Algorithm 2: Update all the classifiers

1. Collect all human annotated labels from the database

2. for each label do
3. if label is different from the original story label then
4. data is sent to each classifier’s retrain() function
5. end
6. end
7. Retrieve a new list of trends from Google News and Twitter API and push these into the trends database table

Algorithm 3: Aggregate new news stories

1. Query Media Cloud and Superglue for all stories published within the past 2 days
2. Check the database and only keep stories that were not yet uploaded
3. for each new story do
4. for each classifier do
5. Feed story to predict() function the classifier
6. Annotate story with classifier score
7. end
8. Upload annotated story to the database
9. end

The worker is built out of different modules - Trends module, Sentiment module, Subjectivity module, Political module and Topics module. Each module (besides the Topics module) has both a retrain() and a predict() functions, that are called as needed by the classifier updating pipeline or the data aggregation pipeline.

4.9 Interface Architecture

The interface is built with React.js and a node.js Express server. It has two pieces of code, the backend (called "panorama-api") and the frontend (called "panorama-site").
The panorama-api is interacting with the database, querying it and pushing new labels to it. This is a Node.js Express server, with POST and GET endpoints to perform various actions. It is connected to the database using the Sequelize model. The server has 3 endpoints:

- **/story** - to retrieve all the stories in a specific time frame and filtered stories if necessary, this endpoint also contains a function that fetched an image that matches a story title from the Google image search API and adds it to the story row in the database.

- **/descriptor** - this endpoint retrieves all the descriptors and their matching stories from the database. It is called whenever the filters are changed to only get stories and descriptors that match the filters.

- **/label** - this endpoint only has one POST request, that adds new labels to the database as they come in.

The frontend part renders the interface itself in the user’s browser. It is built with React.js [82] and contains various components that are described in chapter 5. Some of the components contain actions that send POST and GET requests to the panorama-api.
Chapter 5

User Experience and User Interface

Panorama presents a novel interface for exploring thousands of news stories, as well as a tool to annotate those stories. Human curated labels input by users are delivered to the machine learning classifiers to train on. This section provides an overview of the user interface and describes in detail how a user explores the visualization elements and the day’s news stories and how she could label stories when she disagrees with the automatic labels generated by the machine learning classifiers. The Panorama interface is a single page interface that includes a number of dynamic components. Figure 5-1 shows an annotated image of the Panorama interface that specifies the various components described below.

5.1 Panorama Interface Components

Topic list and topics search bar

Each story in Panorama is labeled with different "topics" by the multi label classifier described in chapter 4. The topic list component appears on the left side of the interface. On the landing page this list include the "hot topics" which are the topics that appeared in most stories in the past 24 hours time frame. The topics are rendered sorted by their order of stories count (how many stories are labeled with this topic). Each topic is shown inside
Figure 5-1: Panorama interface components: 1 - Topic list 2 - Sources 3 - Stories 4 - Filters

- a Gray rectangle, the width of the rectangle represents the number of stories that were labeled with that topic, so the wider the rectangle, the more stories were related to that topic. Clicking on a topic shows stories that are relevant to that topic in the Stories component. It will also replace the topics in the topic list with topics that relate to the selected topic, i.e. topics that also appear on stories labeled with the selected topic. For example, if "Basketball" is selected, the list will likely include "Games", "Sports" and "Baseball". Once selected, the given topic rectangle will be highlighted in purple and the Filters component

Figure 5-2: Panorama - topic list
(described below) will redraw a purple area chart plotting the distribution of that topic over the labels extracted by the classifiers. The sources component also changes when a topic is selected, showing a bar graph that corresponds to the percentage of stories from each source that were labeled with the chosen topic. At the bottom of the topic list there is a search box. At any given point the user can search for topics that are not in the "hot" or "related" topics lists. When typing three letters or more, all topics that match the string in the search box appear under the search box. The search results topics are also sized by the number of matching stories, the user can visually compare how "big" is the topic she is interested in.

**Sources**

The sources component appears on the bottom left side of the interface, under the topic list. On the landing page, all sources are rendered at full height. In this state, the sources are all equally sized bars, colored orange. Clicking on a source in that state colors it in
purple and draws a same colored area chart distribution graph over the classifiers labels shown in the filters component, representing the distribution of stories from that source over each of the filters.

![Figure 5-5: Panorama - selected source](image)

When a topic is selected, the bars in the source components change their height to match the selected topic, visualizing the percentage of stories in each source that reported about the chosen topics. Hovering on a topic bar shows a tool-tip with the current percentage number.

![Figure 5-6: Panorama - sources component](image)

Clicking on a source in this state, draws the distribution over the filters (the combined distribution of both the selected topic and the selected source) as well as filter the stories presented in the stories component, to only present stories from the selected source. Re-clicking the selected source or clicking the "Sources" title will de-select that source.

User testing suggested users were not happy with the layout of this component. They enjoyed the fact that they could see what sources the news stories are coming from, but the font of the sources names was too small and hard to read. Further iterations of the interface design should address this issue and find another solution for presenting the
sources component while also making the sources names more readable.

Filters

The filters component is rendered on the right side of the interface. Each one of the filters represents one of the classifiers described in section 4. Each filter is a range slider. Initially, the full range is selected. Over each slider there is a Gray area chart, representing the distribution of all the stories over the different values of each classifier. When a topic or source are selected, a purple area chart is drawn on top of the Gray area chart, showing the distribution of stories matching that specific selection.

![Filter By](image)

Figure 5-7: Panorama - filters component

When the sliders are moved, a call to the database is triggered, and the stories are filtered, presenting only the stories that match the selected values in the sliders. Moving the sliders and filtering the stories also changes the rendering of the topics list, since "Politics and Government" might be the largest topic when looking at all stories, but if we only count the positive stories, or the left leaning stories, another topic might emerge as the largest topic.

If stories in the stories component are presented, changing the values in the sliders will filter the presented stories and there will be an animation of stories removed and stories added to the screen.
Stories

The stories component is displayed at the center of the screen, and is the largest components in the interface. Initially, this component is empty and it presents an invitation for the user to explore the interface and different controls. As soon as a topic in the topic list is clicked, stories appear in the stories component. The component first renders 12 stories, but the user can click at the bottom to show more. Each story is rendered in a square, and 66% of the stories presented will randomly have a picture in the background to create an interesting, not overwhelming display. The title of the story (or of the television show name, in case the story is a video from Superglue) will appear in the center of the square.

Clicking on a story will push all the stories down and present an embedly frame [83] on top of them. The frame will render the selected story and will also have the title of the story above it, as a link to the original web page. Clicking a story will also open the Story Labeling Component.

![Stories component](image1)

![A story is selected](image2)

Figure 5-8: Panorama stories component
Story Labeling

The story labeling component appears on the right side of the interface, above the filters components, when a story is clicked. This is the "human-in-the-loop" part of the interface. This component shows the user how the specific story that she is currently reading was labeled by each of the classifiers. The top part of the Story Labeling component is a box with 4 sliders. The sliders have the same labels and domains as the sliders in the Filters component, but these are single valued sliders. Initially, the value of each slider is the value that the story has in the database for that specific category (the value given by the ML classifiers). Under the sliders box there is also a short list of the other topics that the story was labeled with by the multi-label classifier. The topics are sorted by relevancy to the story, their size represent that relevancy as well. This list enables the user to explore other topics that are displayed in the story and might be interesting for her.

![Figure 5-9: Panorama - story labeling](image)

The Story Labeling component is built to encourage the user to submit her opinion regarding the story she is currently reading. For example, if the story was labeled very positive by the algorithm, but the user thinks it is actually a negative story, all she has to do is move the labeling slider to the negative point and click the "Label Now" button. That action would send the new labels to the database, and the Panorama worker will consider
them when re-training the classifiers. That way the user helps improving the classifiers for the next use, as well as for other users of the system.
Chapter 6

Evaluation

This section describes how the Panorama Interface was evaluated. It also discusses further evaluation points out studies we would like to perform on Panorama to truly understand the system's benefits and limitations.

6.1 Experiment Design

6.1.1 User Study

To evaluate the user experience and usability of Panorama, we used a service called "WhatUsersDo" [84]. The service supplies videos of users using the interface, while speaking their thoughts. Thus, we obtained feedback directly from users with different backgrounds and needs and got a qualitative analysis and understanding of the system in the eyes of a first-time user. Each user in the study received a series of tasks. After each task, the users mark whether they think they succeeded and how easy or hard the task was. We performed this test with 8 users, all from the United States, over 18 years old. The study instructions are specified in table 6.1.
6.1.2 User Statistics

Besides the qualitative results retrieved through the "WhatUsersDo" service, Panorama also has mechanisms to track user sessions using the Google Analytics platform [85]. Events, such as what descriptors and stories are clicked, how a user changed the filters, and what labels were submitted are registered, as well as how long a user session is.

6.2 Evaluation Results

Quantitative results of the user study can be seen on Figure 6-2.

At the end, users also answered the question "Name one or two things you like and one or two things you would change about this web site. have any final thoughts, comments, or suggestions about this web site?"

Overall, the feedback on this question was positive. "Great job, giving the reader the ability to find news on his terms." "I liked the way the site gave me the choice which topic and media source I could use, even to compare the various slants by the various sources. I liked the way I could choose the slant, how positive the story, objectivity, and trending." "Overall, the webpage has a great layout and I will be visiting it again soon." "I like the premise of this site. It would be nice to be able to have all these news stories, based on your source preferences all in one place easily accessible." "I really liked the in depth personal
A friend recommended a website called "Panorama," which is a site that collects thousands of news stories from all mainstream US media sources and analyzes them with machine learning algorithms. "Panorama" allows you to search, tag, and update information about news articles as well.

Almost all of the users complained about the font size in the sources layout. This will be considered for the next version of Panorama and we will need to find an elegant solution for plotting all 24 media sources we use, while leaving space on the screen but making is easier to read and navigate. "I liked the 'bookshelf' look to it, but it was just too small and hard to read unless I scrolled over it." "The news sources are very small and hard to see. The print needs to be larger." "I would also consider rotating the sources to make them easier to read. " "I did not like the font size on the sources as I found it too small to read easily"

Insights from Google Analytics evaluation: Between April 22, 2017 and May 6, 2017, a period of 2 weeks, the website counted 101 sessions by 43 users. An average session duration was 07:13 minutes, which is very long for a website and might suggest users were engaged and enjoyed the interaction. The events distribution during that time period can be seen in figure 6-3.

<table>
<thead>
<tr>
<th>Task (Instr. 3)</th>
<th>What avenue do you use to obtain your news?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task (Instr. 4)</td>
<td>Please take 10-15 seconds and look at the home page without clicking anywhere. Describe your initial thoughts and what you think is the goal of this Web site.</td>
</tr>
<tr>
<td>Task (Instr. 3)</td>
<td>What avenue do you use to obtain your news?</td>
</tr>
<tr>
<td>Task (Instr. 5)</td>
<td>Find a New York Times Story about Politics and Government</td>
</tr>
<tr>
<td>Task (Instr. 6)</td>
<td>What stories are trending within Finance?</td>
</tr>
<tr>
<td>Task (Instr. 7)</td>
<td>What do you think is the purpose of the filters within the home page?</td>
</tr>
<tr>
<td>Task (Instr. 8)</td>
<td>How much does Fox News talk about basketball?</td>
</tr>
<tr>
<td>Task (Instr. 9)</td>
<td>How long do you usually spend reading news each day?</td>
</tr>
</tbody>
</table>

Table 6.1: User study instructions

filtering options, so not only could I find topics of news that I was interested, but also the tone of the story could be filtered down as well. That is pretty cool. I really could see myself using that often."

79
To conclude, the qualitative study mainly tested the usability and clarity of the Panorama interface. All users were happy with the idea of gathering news from various sources and being able to filter them and select their topics of interest and compare between sources. Not all users understood the filters functionality or used it appropriately. For instance, we can see in the results that instruction 5, the task of finding a NYT story about politics and government, took on average 130 seconds, which was the longest average in the study results. In general, spending more than two minutes to perform a simple task on a web
The Google Analytics results suggest that clicking a topic was the most common method of interaction. Users have also clicked on stories and changed the filters at almost the same amount. Yet, not many labels were submitted and this might be due to confusion, or maybe most users agreed with the current labels or they didn’t feel encouraged enough to submit their opinion.

### 6.4 Further Evaluation

To better understand Panorama and the benefits within it, further evaluation is required. This kind of evaluation will include:

**Larger user study**

A larger user study is required in order to better understand how users interact with the interface. The architecture to conduct such study is already built into the interface. We’ll need to recruit many users to gather insightful statistics about usability. A further survey addressed to engaged users, asking them if they found stories they were interested in and stories they would not usually read, and weather they learned something new about the
distribution of news topics and sources, will give more insights.

Another interesting test would be to learn whether people spend more or less time on reading a news story when they are asked to label it. For this purpose we could perform a study in a lab setting. Such study would ask the subjects to read five stories on a "standard" news aggregator such as Google News followed by reading five stories on Panorama, and then will measure the average time it takes this for this user to read a story in each of the platforms.

A further qualitative test of Panorama would also include surveying professionals that are usually engaged with news, such as journalists and academics, and inquire weather Panorama could be used as a tool them.

Evaluating re-trained classifiers against baseline classifiers

In order to evaluate the interactive machine learning system, we could perform a quantitative test that will assess whether the language models were indeed improved with the human annotated labels. Such evaluation would be conducted after a large user study, when many labels were submitted. We would then take a carefully selected, manually labeled dataset of news stories, feeding them to each one of the originally trained classifiers and the re-trained classifiers, comparing their precision and recall scores.
Chapter 7

Conclusion and Future Work

7.1 Future Work

Other Language Models

The work on Panorama presented a proof of concept for using machine learning language models to categorize and predict the features of a news story, which is an unstructured text document. These predictions were used to sort and visualize the news stories as a whole. The language models were built so they can be re-trained and improved with human labeled annotations, collected by an interactive interface. The specific language models used here (sentiment, subjectivity, political leaning, trendiness and topics extraction) were selected for the sake of the experiment, and were based on a previously done research and freely available labeled data sets that were used as a baseline. Those language models and their parameters could vary, and a rigorous evaluation of the currently used algorithms is suggested. Additionally, new language models could be easily added and tested against existing models. For example, adding a classifier for controversiality of a story or a classifier that predicts whether a text presents a rumor or a fact. The Panorama system architecture is built in a modular way that allows easy addition of new language models and classifiers.
**Personalization**

In its current state, the Panorama interface does not require login. It doesn’t save any information about a specific user and all the users get exactly the same experience. A personalization dimension should be explored in the future, and might encourage users to input more labels, since they would benefit from it directly. A personalized interface would initially ask the user for her preferences and would suggest those topics of interest on the landing page. When a story is read, the user could mention whether she wanted to see more stories like that or less stories like that, which would be a good input label about stories similarity, as well as a good method to help the user explore the topics and stories she is most interested in. When labels are then entered, the labels are read by the system as relative to the user. A personalized interface could also suggest stories that other users with the same “taste” liked and labeled, or recommend popular filters configurations. There are definitely many possibilities to explore in the space of personalized interfaces for this concept.

**Apply To Other Corpora**

The work described in this thesis was applied to news stories data. The same idea and implementation, of using interactive data visualizations for interactive machine learning interfaces could also be explored in other domains and datasets. I can imagine Panorama being applied to a library collection, enabling users to easily explore million of books to find the ones that they would like most. It could also be applied to medical records, to be used by doctors and researchers to explore previously examined conditions and improve the use of machine learning algorithms in healthcare and wellbeing fields. Other ideas would include financial and stock market reports, to help predict future changes, music and entertainment collections and more. As machine learning tools become more accessible, computation powers become cheaper, and the understanding of good human/computer partnerships is developed, I am intrigued to see what new applications can incorporate computer and human abilities in a vision of obtaining the best results.
7.2 Overall Achievements

This thesis describes Panorama, a tool that allows humans and machines to collaborate and achieve a better understanding and analysis of news. This work included the development of a set of statistical language models for analyzing news stories, as well as the design of a web-based user interface for exploring those news stories and comparing various aspects of them. Panorama allows users to view, sort, filter and organize a large set of news stories from U.S. mainstream media sources. It supports exploration and serendipity in news consumption, by not selecting or highlighting anything in advance while putting the power in the hands of the user. Panorama also encourages users to submit their opinion and "criticize" the performance of the machine learning algorithms. This feedback is used to automatically improve the underlying mechanisms and bringing the human back into the learning loop.

The Panorama interface was evaluated to assess its ability to engage, inform, and facilitate the navigation of users through the news stories they want to read. These evaluations suggested that users are interested in having the power to customize their news and take their own choices, and that the Panorama interface was clear and enabled most users to do so. It also suggested that some changes to the interface are required to further amplify its power and functionalities. Further evaluations are needed to assess the effects of Panorama on news consumption in various contexts, as well as assessing the interactive machine learning mechanism, and the quality of the labels submitted by users.

The concept was applied to news stories which are currently curated by editors and algorithms who define our social dialogue and create a feed that can misinform as well as it can lead. Panorama was created with the thought that we can do better. While human editors have their own implicit biases and agendas and machine intelligence is often opaque and mystical. Here, design and interaction clarifies the process, transforming the hidden workings of a machine into a useful human and human partner, a conversation instead of a consultation.

The principles suggested in this thesis have applications in fields other than news, such as social media, music collections, healthcare and more. Being able to combine hu-
man and machine abilities to get the best capabilities of each in processing and exploring information is a valuable tool in our world that is becoming so rich with data.
Bibliography


[28] Jigsaw. If you are reading this, we might be in the same news bubble. https://medium.com/jigsaw/if-you-are-reading-this-we-might-be-in-the-same-news-bubble-cb697270c698#.w493f4i4h, March 2016.


