VizByWiki: Mining Data Visualizations from the Web to Enrich News Articles

Allen Yilun Lin[1], Joshua Ford[2], Eytan Adar[3], and Brent Hecht[1]
[1]Northwestern University; [2]Apple Inc. (work done at the University of Minnesota); [3] University of Michigan
allen.lin@eecs.northwestern.edu, ford0420@umn.edu, eadar@umich.edu, bhecht@northwestern.edu

ABSTRACT

Data visualizations in news articles (e.g., maps, line graphs, bar charts) greatly enrich the content of news articles and result in well-established improvements to reader comprehension. However, existing systems that generate news data visualizations either require substantial manual effort or are limited to very specific types of data visualizations, thereby greatly restricting the number of news articles that can be enhanced. To address this issue, we define a new problem: given a news article, retrieve relevant visualizations that *already exist on the web*. We show that this problem is tractable through a new system, *VizByWiki*, that mines contextually relevant data visualizations from Wikimedia Commons, the central file repository for Wikipedia. Using a novel ground truth dataset, we show that VizByWiki can successfully augment as many as 48% of popular online news articles with news visualizations. We also demonstrate that VizByWiki can automatically rank visualizations according to their usefulness with reasonable accuracy (nDCG@5 of 0.82). To facilitate further advances on our "news visualization retrieval problem", we release our ground truth dataset and make our system and its source code publicly available.

KEYWORDS: news articles; Wikimedia Commons; user-generated content; data visualizations; peer production; Wikipedia

1 INTRODUCTION

Data visualizations have become an increasingly prominent part of the news landscape [2, 10, 26]. Indeed, *The New York Times*, *The Washington Post*, *The Wall Street Journal* and *The Guardian* all now operate entire teams that design and publish data visualizations [33]. Recent research suggests that this trend is quite beneficial for the reader: empirical evidence has shown that data visualizations can make complex relationships in news articles easier to comprehend and can provide critical context for news narratives [33]. More generally, coupling text with data visualizations has been shown to improve understanding and recall over either text or visualizations alone [5,8,15] (in accordance with the well-established dual-coding theory from cognitive science [31]).

While news data visualizations are highly beneficial and in great demand, creating them requires time, money and expertise. This means that local news organizations usually cannot afford to make data visualizations, and even large national news outlets are only able to create them for a tiny fraction of articles. Researchers from information visualization and data-driven journalism [2] have developed many tools to help automate this process. However, as we discuss below, these systems either still require substantial human intervention (e.g., [23, 30, 31, 41]) or only work with very specific topical domains (e.g., financial reports [11]) or visualization types (e.g., maps [8]).

In this paper, we propose an alternative and tractable approach to automatically add contextually-relevant data visualizations to news articles with no human intervention. Moreover, unlike existing techniques, our approach has very few constraints with respect to topical domain or visualization type.

Our approach is based on a novel insight: there is often no need to create visualizations from the ground up because *large numbers of data visualizations already exist in Wikimedia Commons*. Wikimedia Commons, or "the Commons" [40], is the media repository used by Wikipedia editors and has over 41 million images, a non-trivial percentage of which are data visualizations created to support Wikipedia articles. Moreover, the visualizations in the Commons have redistribution-friendly licenses, making the Commons a particularly appealing resource for news publishers. To the best of our knowledge, this paper is the first to recognize and leverage the rich corpus of data visualizations that exists in the Commons. As we discuss below, we hope our research can support further inquiry into the value of this important repository, both within the news visualization context and beyond it.

In more formal and general terms, this paper defines a new problem that we call the *news data visualization retrieval problem*. Given an arbitrary news article, the goal of the news data visualization retrieval problem is to automatically retrieve relevant, pre-existing data visualizations to support the article from a given repository (in our case, the Commons).

WWW 2018, April 23-27, 2018, Lyon, France.

ACM ISBN 978-1-4503-5639-8/18/04.
DOI: https://doi.org/10.1145/3178876.3186135

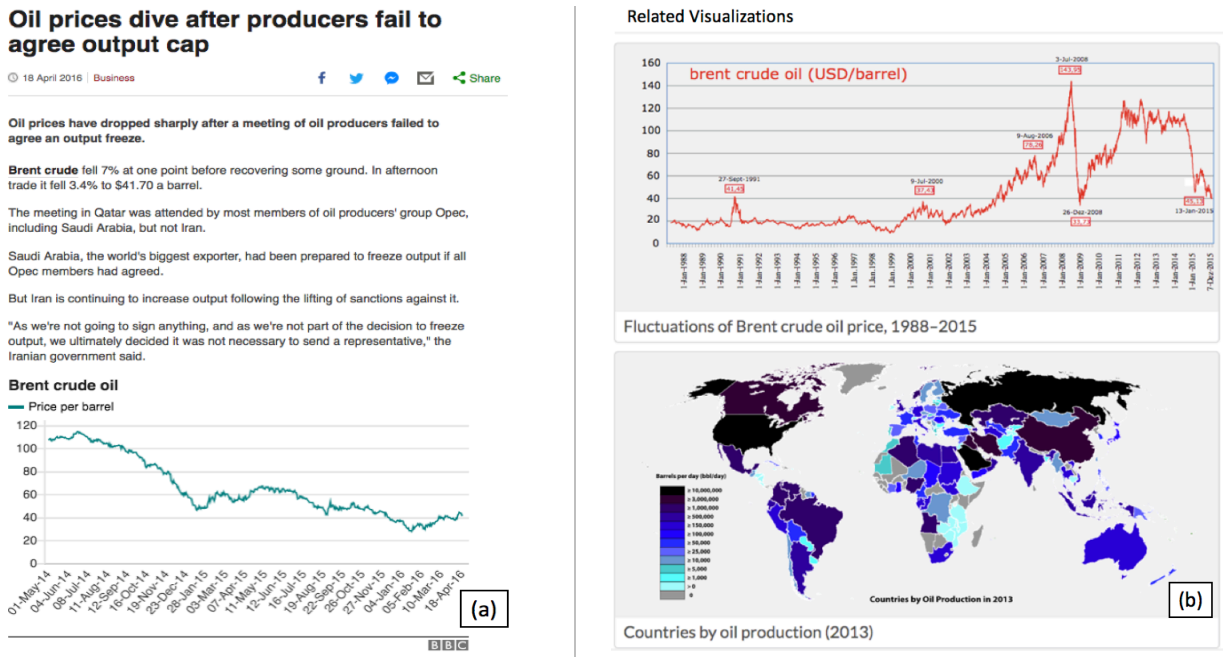

Figure 1: (a) A BBC article about the oil price drop after producers failed to agree on an output freeze and (b) the data visualizations retrieved by VizByWiki. In (b), the retrieved data visualizations are ranked in descending order according to readers' perceived usefulness.

We demonstrate that the news data visualization retrieval problem is *tractable* through a new, end-to-end system called *VizByWiki*[1]. As an example, consider Figure 1, which shows an article about a drop in oil prices that includes a line chart created by journalists (Figure 1a). For this article, VizByWiki mined the Commons and retrieved a line chart and a map and ranked them according to their usefulness to this article (Figure 1b). Compared to the data visualizations in the original article, the line chart presents the same variables (Brent Crude oil price over time, although for a longer time range). Additionally, the thematic map was rated as useful to the article by study participants, but was not included by the news publisher.

To formally evaluate VizByWiki and to learn the factors that predict useful data visualizations, we designed a crowdsourcing task to collect human ratings on the usefulness of the visualizations retrieved by VizByWiki (which resulted in the usefulness score for the map in Figure 1). We demonstrate that VizByWiki can enrich around 50% of popular online news articles with at least one somewhat useful data visualization. Moreover, using this ground truth dataset, we gained an initial understanding of the factors that determine useful data visualizations for news articles and trained VizByWiki to rank visualizations according to this understanding. As we show below, VizByWiki's ranking accuracy is good, with an nDCG@5 of 0.82.

Overall, this paper both contributes a **new problem** (the *news data visualization retrieval problem*) and demonstrates that the problem is tractable with an **end-to-end system** contribution (VizByWiki). We also make two other contributions whose impact may generalize beyond the news visualization context. First, this paper helps to demonstrate the **tremendous potential of Wikimedia Commons**, a resource that – unlike its sister project Wikipedia – remains largely untapped by the computing research community. Second, a key component of VizByWiki involves **automatically distinguishing visualizations from non-visualizations**, a new challenge that could be relevant to other domains (e.g., image classification). In this paper, we show that this challenge can be addressed with a F1-score of 0.91 by simply using a pre-trained Convolutional Neural Network (CNN).

The structure of this paper follows best practices in the systems research genre (e.g., [8, 11, 14, 32]) by first providing an overview of our system's components and then outlining the novel challenges faced in each component and describing how the challenges were addressed. Prior to this, however, we begin below by highlighting related work.

2 RELATED WORK

In this section, we review the three research areas that most directly motivated this work: 1) automated text illustration, 2) other systems that generate news data visualizations, and 3) methods for data visualization classification.

[1] Link to the system demo and source code repository: http://www.psagroup.org/projects/vizbywiki.

2.1 Automated Text Illustration

One area of the literature that provided key motivation for this work is text illustration. Text illustration is a constrained image retrieval problem that focuses on retrieving illustrative images for long texts. Some of this work has focused specifically on illustrating news articles with multimedia files. For instance, Li and Hai [17] illustrate news articles with images from Flickr. NewsMap [19] on fusing image search results from multiple short queries that are generated from news articles. Similarly, Delgado and Joao's system [5] constructed a visual story of news articles by finding sequence of images. BreakingNews [28] leveraged CNN to learn to match the original images and the text of the news article..

Some text illustration systems also use images on Wikipedia, but focus on a small-scale, self-curated sample of these images (e.g., [37]). The only work that considers all Wikipedia images is by Agrawal et al. [1], who designed a book illustration system to enrich textbooks for developing countries. Their method essentially ranks Wikipedia images according to scores computed from the token overlap between the keywords of the text and the descriptions of the image. Due to its similarity to our problem (using Wikipedia images for long text), we implemented their algorithm and used it as a baseline for comparison in the evaluation section, showing that we outperform this approach by a solid margin.

However, the *news data visualization retrieval problem* is also fundamentally different from text illustration problem. For example, consider a news article about air pollution in Beijing. In the text illustration problem, the ideal solution would be a photo depicting smog in Beijing. However, in the news data visualization scenario an appropriate image would be a bar chart showing the number of days in each month that have a PM2.5 reading greater than the safe level. More generally, the news visualization retrieval problem is interested in adding new information to contextualize a story, not finding a photo that will depict exactly what is written in the text. Similarly, visual features play a much different role in the data visualization problem; data visualizations of the same type could share very similar visual features but cover substantially different topics and data. Our methodological choices reflect our considerations of these important differences, and is one reason our approach outperforms the baseline method for text illustration.

2.2 Generating Data Visualizations for News

Several research projects in the information visualization and data-driven journalism domains have sought to automate the ground-up generation of news data visualizations. However, these systems either (1) still require a substantial degree of manual effort or (2) focus on a narrow domain within news articles. An example of the former case is the MuckRaker system [23], which provides a user interface to help find and visualize structured data from databases that is relevant to a given news article. Research systems that automate the creation of specific categories of news visualizations include Contextifier [11], which produces annotated stock visualizations for financial news articles. However, the system is limited to financial news and only generates line charts. NewsViews [8] is built to automatically generate geovisualizations for news articles through a pipeline that involves identifying toponyms and topics, finding relevant tabular datasets and creating a thematic map. Like Contextifier, NewsViews focuses only on a specific type of data visualization (thematic maps) and a specific type of article (those with strong geographic elements). NewsViews is further limited by the small, manually-curated structured datasets from which the system generates maps. In comparison, VizByWiki does not require human intervention and operates without the limitations on the types of data visualizations, the types of news articles, and the diversities of curated datasets.

2.3 Classifying Data Visualizations

As described below, a necessary step in VizByWiki is differentiating between data visualizations (e.g., bar charts and pie charts) and other images (e.g., photos, engineering diagrams). This problem is related to the problem of identifying different visualization mark types, e.g., separating bar charts from pie charts. This is a task addressed in several papers that try to recover the data behind statistical charts. For instance, ReVision [32] classifies visualization types using a combination of textual features from OCR and low-level visual features which capture prominent patterns in the images. More recently, deep learning has been used in this problem space. For example, ChartSense [14] includes a mark type classifier that was trained from scratch using the GoogLeNet architecture. Similarly, Heer et al. [27] built a visualization type classifier by fine-tuning a pre-trained CNN and, using the same fine-tuning technique, FigureSeer [35] successfully trained a classifier to distinguish different types of results figures in research papers.

While VizByWiki faced a different problem than the research described above (i.e. determining whether an image is a data visualization vs. distinguishing between types of visualizations), previous works' success using pre-trained CNNs provided key methodological guidance for our approach.

3 SYSTEM OVERVIEW

In this section, we first provide an overview of VizByWiki's high-level user experience. We then describe VizByWiki's behind-the-scenes system architecture. Finally, we discuss the different types of datasets that are used to build VizByWiki.

3.1 User Experience

The primary audience for VizByWiki is the millions of people who read news online. For this audience, we have built a working browser plug-in (demonstrated as a web application in the URL on page 2) that implements all the techniques below. The plug-in processes unstructured text from a news article and presents users with data visualizations that are ranked by their usefulness alongside the news article (as shown in Figure 1b). A potential secondary audience for VizByWiki is those in

the data-driven journalism community [26]. For this audience, VizByWiki could be used as an exploratory tool to inform the design of customized data visualizations (as well as perhaps in the journalistic discovery process).

3.2 System Architecture

In this section, we provide a high-level overview of the system architecture of VizByWiki. As shown in Figure 2, VizByWiki consists of a three-stage pipeline: 1) topic filtering, 2) data visualization identification and 3) data visualization ranking. In Stage 1, VizByWiki uses Wikipedia articles that are relevant to the news article, and that contain Wikimedia Commons images as a proxy to filter out topically irrelevant images. To find relevant Wikipedia articles, VizByWiki first applies entity linking techniques to identify Wikipedia concepts/articles that are mentioned in the news article text. The resulting Wikipedia article set is then expanded by finding the most semantically related Wikipedia articles using semantic relatedness (SR) measures. All images in the expanded set of Wikipedia articles are extracted and delivered to Stage 2.

In Stage 2, the extracted images are filtered to isolate data visualizations from non-data visualizations. VizByWiki employs a two-step filtering approach that consists of both straightforward heuristics and CNN-based transfer learning techniques. The output of Stage 2 – a set of topically relevant data visualizations – is then processed by the data visualization ranker in Stage 3. The ranker was trained using a new ground truth dataset that we collected. The dataset consists of human judgments describing which data visualizations are useful to which news articles.

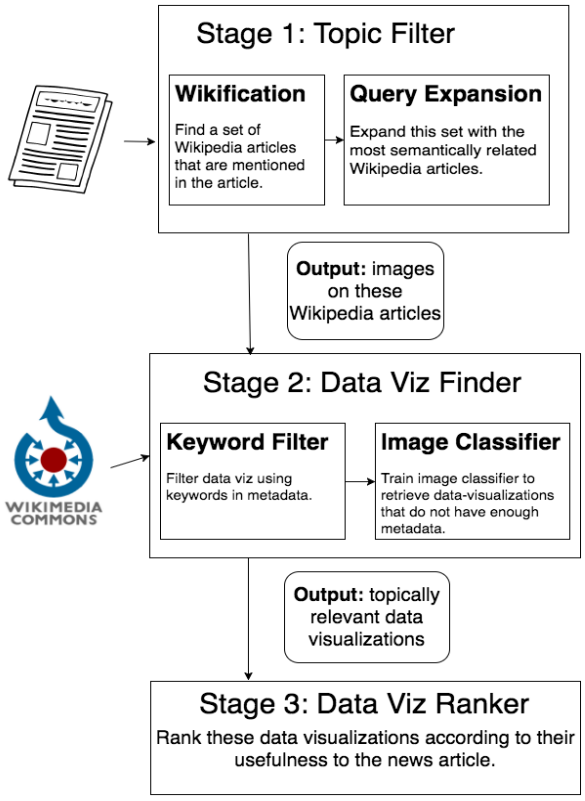

Figure 2. Overview of VizByWiki's architecture

3.3 Dataset

3.3.1 Wikimedia Commons and English Wikipedia

VizByWiki augments news articles with data visualizations from Wikimedia Commons. Wikimedia Commons is the central media files repository for all the Wikipedia language editions and is the largest repository of freely licensed educational content in the world [40]. We chose Wikimedia Commons over other popular media repositories (e.g., Flickr) because it contains a large number of data visualizations due to its encyclopedic focus. Using a highly accurate data visualization classifier developed as part of VizByWiki (see below), we estimated that the Commons contains roughly three million data visualizations. We were also attracted to the Commons' licensing regime, as this mitigates any real-world legal barriers to any commercial use of VizByWiki. As discussed below, the Commons proved to be a powerful repository of data visualizations that we believe can be leveraged in contexts beyond this project.

An obstacle to directly leveraging Wikimedia Commons is its poor metadata. The Wikimedia Commons community itself has stated that images are "described only by casual notation, making it difficult to fully explore and use this remarkable resource" and even this "casual notation" does not exist for all images [40]. Moreover, while the Commons began a long-term project to standardize its metadata in 2017, this process is far from finished [40].

To address this obstacle, we used the English Wikipedia to augment the knowledge that we have about Commons images. The Commons is the dominant image repository for most Wikimedia projects, including Wikipedia [42]. As such, English Wikipedia articles and the text around Commons images that appear in these articles can provide natural semantic embeddings for Commons images. We note that this may be a useful approach to seed the metadata standardization process in the full Commons as well. However, a side effect of this approach is that it does limit our pool of potential data visualizations to those that appear in at least one English Wikipedia article.

To process the full English Wikipedia, we leveraged WikiBrain [34], a software framework that processes Wikipedia XML dumps and provides access to a range of Wikipedia-based algorithms (including the semantic relatedness algorithms we used). For our studies, we used the June 23, 2017 English Wikipedia dump.

3.3.2 News Articles Datasets

To perform realistic experiments on VizByWiki, we sampled two datasets of popular online news articles from major news outlets. The first news article dataset (which we call the *ad hoc* dataset) was originally collected during April 2016 (for an earlier project) and consists of 40 popular online news articles sampled arbitrarily from the home pages of various large news

outlets including *CNN*, *Fox News*, *BBC*, and *The New York Times*. Some of these news articles are accompanied with data visualizations designed by experts. We used this dataset primary for early feasibility testing.

Our core dataset (called *uniform*) was collected during July 2017 and consists of 60 articles sampled from the different topical categories provided by popular news outlets. These articles were sampled through the RSS feeds of *Fox News* and *CNN,* both of which are organized into topics including "World", "U.S.", "Business", "Politics", "Technology", "Health", "Entertainment", and "Travel". We randomly sampled the same number of articles from each category. Different from the previous dataset, these popular online articles were selected without consideration of whether they already contained visualizations. The purpose of this dataset is to evaluate the general utility of the system in an ecologically valid fashion, i.e. how well it could retrieve data visualizations for arbitrary news content.

In some evaluations, we also combined these two datasets into a single 100-article dataset. We call this dataset *combined*.

4 STAGE 1: TOPIC FILTERING

The first stage in the VizByWiki pipeline is identifying topically relevant Wikimedia Commons images for an input news article. Here, we used Wikipedia articles that contain the Wikimedia Common images as proxies for the topic of the images. VizByWiki deploys a two-step process that leverages both wikification [4, 24] and query expansion techniques to identify the appropriate Wikipedia articles.

4.1 Wikification

Wikification involves disambiguating named entities in unstructured text to Wikipedia articles [24]. For example, the system can recognize that "travel ban" in the recent news article refers to "Executive Order 13769" (and linked to the associated Wikipedia page) and not two separate words ("travel" and "ban"). We use a technique developed by Noraset et al. [25], which uses hybrid rule-based named entity recognition to discover terms and a learned model to disambiguate their corresponding Wikipedia entities.

The output of the wikification process contains many entities that are trivially related to the main topic of the article (e.g., countries that were mentioned in the news article but not included in the list of banned countries). To filter out these entities, we computed the semantic relatedness (SR) between these "wikified" entities and the news article content and only kept entities that are highly related to the article content. Specifically, we leveraged WikiBrain's implementation of the Explicit Semantic Analysis semantic relatedness algorithm (ESA) [7] which maps words into Wikipedia-concept-based embeddings and computes semantic relatedness as the cosine similarity of the embeddings. We only kept wikified entities that have had an ESA SR greater than 0.8 (out of 1) with the news content. In WikiBrain, an SR score of 0.8 effectively means that this score is at 80th percentile of all SR scores [39].

Table 1. Performance of Stage 1: Topic Filtering

	Avg. # of Wiki articles per news	Avg. # of images extracted per news article
Wikification	6.3	56.0
After query expansion	10.3	69.6

4.2 Query Expansion

The entities that are output from the SR filter are used as "seed queries" for retrieving Wikimedia Commons images. However, query expansion is also necessary due to Wikipedia's organization and its relationship to the articles on which data visualizations about specific topics appear. For instance, content about a concept is often contained not only in the "main article" about the concept (e.g., the "United States" article) but also in "sub-articles" (e.g., the articles "History of the United States", "Geography of the United States", "American Literature" and so on) [21]. This issue has important implications for our problem. Consider again a news article that covers the Trump administration's changes to U.S. travel policy. The term "immigration policy" in the article might be correctly disambiguated to the Wikipedia article "Immigration Policy". However, this linkage will miss the Wikipedia article "Immigration Policy of Donald Trump", which is a sub-article of "Immigration Policy" and contains useful statistical charts related to foreign-born workers in the US labor force.

To address this issue, VizByWiki expands the queries to include articles that are highly semantically related to wikified entities using ESA. Similar to the criteria above, to ensure that these expanded entities are highly relevant, we only include them if they have an SR > 0.8 with the news article.

Finally, the output of Stage 1 consists of Wikimedia Commons images that are extracted from the Wikipedia articles identified from the above two steps.

4.3 Evaluation

We validated the feasibility of our topic filtering approach using the *combined* dataset. Table 1 reports that on average, our approach could "wikify" 6.3 Wikipedia entities for each news article (after the SR filter). Table 1 also shows that query expansion with semantic relatedness successfully increased this number to 10.3. From these Wikipedia articles, Stage 1 extracted an average of 69.6 unique image candidates for each news article (6473 unique images in total for the *combined* dataset). The results in Table 1 demonstrate that our topic filtering approach is a viable approach to retrieve many image candidates for the later stages of the system (although it does not provide a guarantee that this is the *best* such approach, a topic to which we return in Discussion).

5 STAGE 2: IDENTIFICATION

The goal Stage 2 is to identify the visualization "needles" out of the large "haystack" of diverse images from Stage 1.

5.1 Problem Definition

Prior literature [11] has identified that common types of news data visualizations include maps, line graphs, bar graphs, bubble charts, scatterplots, tree maps and pie charts. Less common forms such as area graphs and Venn diagrams are also broadly viewed as data visualizations [27, 32]. In VizByWiki, we assume that the definition of data visualizations subsumes all of the above types.

As mentioned above, information visualization researchers have examined the problem of distinguishing different types of data visualizations from one another. However, the problem we face here – separating data visualizations from non-data visualizations – presents two major challenges that make it distinct from prior work. First and foremost, unlike our problem, research on visualization type classification usually starts with a corpus containing only data visualizations [14, 27, 32, 35]. By contrast, Wikimedia Commons contains much more diverse types of images, including many non-data visualization images that share visual similarities with data visualizations (e.g., engineering diagrams, photos of paper maps, logos). Second, due to the crowdsourced nature of Wikimedia Commons, even visualizations of the same type can be visually heterogeneous, which increases the difficulty of classification. Datasets used in the type classification problem do not contain this degree of heterogeneity; they are often manually curated by researchers [32], designed in the same fashion from the same tool [27], or carefully generated to specific standards by professionals [35].

To address both of these challenges, VizByWiki contains a two-step data visualization identifier that leverages both textual and visual features. We explain each step in detail below.

5.1 Keyword Filtering

In the first step of our data visualization identification process, a naive keyword filter eliminates obvious non-data visualizations using rule-based heuristics. The filter uses text metadata from both Wikipedia and Wikimedia Commons. Metadata considered includes image captions from Wikipedia, file descriptions and category tags from Wikimedia Commons, and machine-generated EXIF metadata. To develop our heuristics, one researcher went through a series of example images and identified keywords that indicate obvious non-visualizations. These included "photograph," "picture," "image," "featured," "photo," "portrait," "road sign," and "coats of arms." We also excluded images that contain camera EXIF information (e.g., a camera make/model), which clearly indicate an image from a digital camera rather than a data visualization. On the *combined* dataset, the keyword filter identified 5718 images as obvious non-data visualizations out of the 6473 images from Stage 1's output. This left us with 755 candidate images that were potentially data visualizations.

5.2 Image Classifier

In the second step of our data visualization identification process, we trained an image classifier using visual features. While keyword filtering effectively screens out many *obvious* non-data visualizations, the output of the keyword filter still contains many images that are not data visualizations. This is primarily the result of three issues: (1) *sparseness*: many images on the Commons have limited metadata, (2) *errors*: the metadata that is available can be inaccurate, and (3) *coverage*: it is intractable to develop a complete set of filter keywords.

As such, to increase precision, we used a pre-trained convolutional neural network (CNN) to design an image classifier that differentiates between data visualizations and other images. CNNs have been shown to be effective in the data visualization classification tasks described in related work (e.g., [14, 27, 35]), but they also require enormous numbers of ground truth images for training. One way to address this is to leverage the power of transfer learning with a pre-trained CNN, and we adopted this approach in VizByWiki. Specifically, we used a pre-trained CNN as a feature extractor: we leveraged the output from the second-to-last CNN layer as a vector representation of each image, and fed these representations into a traditional classifier. This approach has been successfully employed on a wide range of image recognition tasks [29].

In our implementation, we used the InceptionV3 CNN pre-trained on ImageNet [36] and a support vector machine (SVM) classifier. Because the features from the second-to-last layer of the pre-trained CNN are sparse and high-dimensional (1024 dimensions), feature engineering was necessary before training on our relatively small ground truth dataset (detailed below). We applied Principle Component Analysis and used the top-20 principle components as features (which account for about 51% variation). These 20 features were normalized to aid in the training process.

To obtain a ground truth dataset that accurately represents the underlying data distribution of this problem, we labeled all 755 images that were output from our keyword filtering step. Since manually differentiating data visualizations from other images is a relatively unambiguous task (under the concrete definition of news data visualizations enumerated at the beginning of this section), one researcher manually sorted our images into a data visualization class (455 images) and a non-data visualization class (300 images). This dataset was then split into a development set (50%), which was used to train and tune the hyperparameters of the SVM, and an evaluation set (50%), which was used to evaluate classifier performance. We used grid search to tune a variety of hyperparameters.

The best performing SVM classifier was found to employ a radial basis function (RBF) kernel with $\gamma = 0.01$ and $C = 100$. Table 2 shows the results of this classifier on our test dataset. We were able to achieve an average F1 score of 0.91 and

Table 2. Performance of image classifier

Class	Precision	Recall	F1-score
non-dataviz	0.89	0.88	0.89
dataviz	0.93	0.91	0.91
avg	0.91	0.91	0.91

roughly equally high F1 scores for both the non-data visualization class (0.89) and the data visualization class (0.91). These are important results for two reasons. First, they represent a more-than-adequate overall accuracy for our VizByWiki prototype, allowing us to continue to the Stage 3 ranking task. Second, they indicate that the visual features learned for detecting objects in natural scene images (one of the main ImageNet tasks) are useful for identifying data visualizations (which are mostly computer-generated with very different visual characteristics), a finding that deserves more exploration.

6 STAGE 3: RANKING

In Stage 1, VizByWiki extracts images relevant to an input news article from Wikimedia Commons and in Stage 2, VizByWiki filters out images that are not data visualizations. The goal of Stage 3 – the final stage – is to rank the data visualizations output from Stage 2 according to their usefulness to the reader (as in Figure 1b). In this section, we first discuss how we formulated the Stage 3 problem into a "learning to rank" problem. We then document how we collected a novel visualization usefulness ground truth dataset through crowdsourcing. Lastly, we use this dataset to conduct two important evaluations: one for the overall system's general feasibility and the other specifically for the ranker's performance.

6.1 Problem Formulation

We formulated Stage 3 as a learning to rank problem similar to the one that is typical for search engines: given a news article, our goal was to rank a set of data visualizations by their usefulness to the news article. Due to the novelty of this problem, we had to construct our own ground truth dataset. To build this dataset, we used the data visualizations output from our work in Stage 2 and paired them with their corresponding news articles. We manually corrected all classification mistakes in order to ensure a data-visualization-only dataset, allowing us to focus purely on the ranking task. This dataset consists of 572 {*news article*, *candidate data visualization*} pairs (Note that one data visualization can be paired with multiple articles).

For each pair, we generated both textual and visual features. Specifically, the textual features are as follows:

- **content-caption**: The semantic relatedness score (computed by Explicit Semantic Analysis) between the news article content and the visualization caption written by the editors of the Wikipedia article in which the image is used (the article identified in the Stage 1). Note that even though metadata for images is sparse and can be inaccurate in the Commons, almost all images have a caption when they are included in Wikipedia.
- **content-WPtitle**: The semantic relatedness score between the news article content and the title of the Wikipedia article that contains the candidate image
- **title-caption**: The semantic relatedness score between the news article title and the visualization caption.

The visual features are much simpler:

- **CNNembed**: These are the same features that are used to train the image classifier in Stage 2. They are the top 20 principal components from the 1024-dimension features extracted from a pre-trained CNN.

For our ranking algorithm, we used the popular RankSVM [13], which employs a pairwise method that is trained to minimize the number of inversions. We implemented the RankSVM as a linear kernel SVM using the Python package scikit-learn.

6.2 Collecting Ground Truth Ratings

For each {*news article*, *candidate data visualization*} pair, we used crowdsourcing to collect ground truth usefulness ratings. This dataset allowed us to assess the feasibility of the news data visualization retrieval problem (Section 6.3) and to learn to rank the data visualizations for each news article (Section 6.4).

2.2.1 Task UI. The crowdsourcing platform we used was Amazon Mechanical Turk (MTurk). Figure 3 shows the UI of the task (re-scaled for clarity). Upon accepting our task, a crowdworker ("Turker") was shown a task tutorial and an example is provided. The Turker was then randomly assigned to one of the 100 news articles in the *combined* corpus.

After reading the article, the Turker was presented with all the candidate data visualizations that are extracted by VizByWiki for the article (output of Stage 2) and was asked to rate each visualization on a scale of 0-3 according to how useful the data visualization is. Each data visualization was accompanied with its original Wikipedia article caption and could be clicked to zoom in if the Turker wanted to examine its details.

Usefulness was assessed on a four-point scale: 0 = "not useful"; 1 = "somewhat useful"; 2 = "useful"; 3 = "very useful"). A useful visualization was defined as one that "helped explain or provide context" to the article. We considered evaluating the retrieved visualizations on various lower-level characteristics

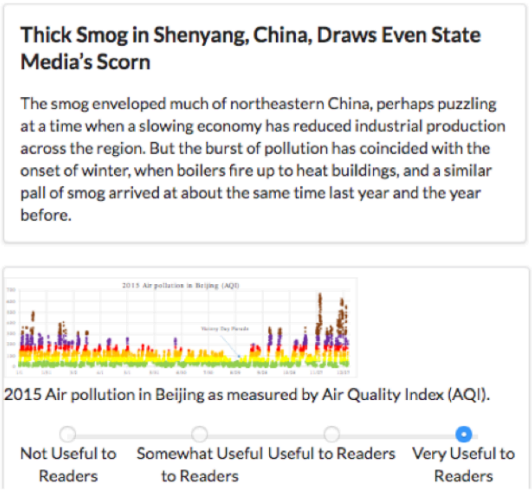

Figure 3. MTurk task UI for usefulness ratings collection. News article attenuated due to length.

from the visualization domain, e.g., expressiveness [22] and interestingness [8] – rather than usefulness. However, we determined that usefulness, as defined above, would more directly capture overall user experience at this stage of the exploration of the news data visualization problem.

2.2.1 Improving the quality of crowdsourced data. As is well-known in the human computation domain, crowdsourced data is subject to quality issues such as *spam*, *errors,* and *biases* [12]. For instance, unless precautions are taken, some crowdworkers will fill in random answers without reading the questions in order to make as much money as possible in a short period of time. As such, we implemented the following strategies in our task to improve crowdsourcing quality:

1) Following Chang et al. [3], we ensured that in the final dataset, no one worker did more than 5% of all the tasks. This simple technique effectively eliminates large-scale spam.
2) To minimize the effect of personal biases and unintentional errors, we rely on redundancy [12]. For each pair of {*news article*, *candidate data visualization*}, we collected 4 ratings and used the median rating as the final rating.
3) We added verification questions as suggested by Kittur et al. [16]. After workers read the news article and before they started rating the data visualizations, workers were required to answer a multiple-choice question about an obvious fact in the article.

In total, for 572 pairs of {*news article*, *candidate data visualization*}, we collected 2288 ratings (4*572). To facilitate further advancement on the news data visualization retrieval problem, we are releasing our ground truth data (see system URL above).

6.3 Evaluation 1: General Feasibility of the News Data Visualization Retrieval Problem

With the ground truth data, an important question to address even before training the ranker regards the feasibility of the broader news data visualization retrieval problem. In other words, can Wikimedia Commons provide useful visualizations for a non-trivial number of popular news articles?

To answer this question, we used two metrics: 1) for a given news article, how many "good" data visualizations could be retrieved by VizByWiki and 2) how many news articles could be augmented by at least one "good" data visualization. We used two definitions for "good" data visualizations: 1) data visualizations with Turkers' median ranking greater or equal to 1 (i.e. *somewhat useful*), 2) data visualizations with Turkers' median ranking greater or equal to 2 (i.e. *useful*).

Table 3 reports the feasibility evaluation of VizByWiki. Most importantly, for our core news dataset *uniform*, VizByWiki could retrieve at least one **somewhat useful** data visualization for 48.3% of articles and could retrieve at least one **useful** data visualization for 21.7% of articles. Recall that our *uniform* dataset contains popular online news articles that are randomly and uniformly sampled from diverse topics. Hence, our results demonstrate that our approach to news visualization retrieval and the use of the VizByWiki system specifically *could result in anywhere from one-fifth to one-half of popular online news articles being enhanced by at least one data visualization*. We note that we observed roughly the same results for the *ad hoc* dataset as well.

Table 3. Feasibility evaluation of VizByWiki.

Metric	*Uniform*	*Ad hoc*
% of articles with ≥ 1 **somewhat useful** dataviz	48.3%	52.5%
% of articles with ≥ 1 **useful** dataviz	21.7%	27.5%
avg. # **somewhat useful** dataviz per article	4.6(5.3)	4.2(3.9)
avg. # **useful** dataviz per article	3.5(4.0)	2.7(1.7)

Note: Standard deviations shown in parentheses where relevant.

To understand the articles for which VizByWiki did not retrieve at least one **somewhat useful** data visualization, one researcher carefully read over 20 such articles. We found that 1) most of these articles did not explicitly reference any structured data in the text and 2) some of these articles covered very recent breaking news in which data might need to be gathered or updated quickly (e.g., an earthquake, election). We return to both of these points below.

Looking at the number of data visualizations that were retrieved for each article, VizByWiki was able to retrieve an average of 4.6 **somewhat useful** visualizations and 3.5 **useful** visualizations for our core dataset *uniform* (including all the zeros for articles for which no visualization could be retrieved). A similar trend can be observed for the *ad hoc* dataset. However, the actual number of good data visualizations varies significantly across articles. As such, it is reasonable to conclude that although our approach likely is feasible for large-scale visualization enhancement, the performance of our approach is better for some articles than for others. These results also suggest that for articles for which VizByWiki can produce useful visualizations, ranking is important as there is often a non-trivial number of visualizations per article. We address this ranking problem in the next sub-section.

6.4 Evaluation 2: Performance of the Ranker

For our ranking experiment, we used the *combined* dataset. We used 50% of the dataset as a development set (training and hyperparameter tuning), and held out 50% for evaluation.

We assessed our data visualization ranker using the conventional search engine evaluation approach involving the *nDCG@k* metric. To put our ranker's performance into context, we additionally implemented a ranking algorithm from Agrawal et al. [1], which involved ranking Wikipedia images for their relevance to a particular section from a textbook. This method essentially relies on token overlap between the keywords of the text and the descriptions of the image. The Agrawal et al. approach provides a useful baseline to help understand our ranker's performance.

Table 4 shows the performance of our ranker trained on different feature sets and compared to the baseline method by Agrawal et al. We computed *nDCG@k* (where k = 3, 5 and 7) for

Table 4. The performance of the supervised ranker using different features and compared to the baseline method in [1].

Features	nDCG@3	nDCG@5	nDCG@7
baseline [1]	0.69 (0.30)	0.74 (0.25)	0.78 (0.21)
all textual features	**0.77 (0.32)**	**0.82 (0.23)**	**0.84 (0.19)**
all textual features all visual features	0.69 (0.31)	0.75 (0.25)	0.78 (0.22)
content-caption title-caption	0.69 (0.29)	0.75 (0.24)	0.79 (0.21)
content-WPtitle title-caption	0.73 (0.26)	0.81 (0.19)	0.83 (0.17)
content-WPtitle content-caption	**0.79 (0.26)**	**0.82 (0.17)**	**0.85 (0.19)**

Note: The nDCG is the mean nDCG across all queries (i.e. news articles). Standard deviations are in brackets.

each news article and the mean and standard deviation of *nDCG@k* is what is shown in Table 4. The results in row 2 show that our ranker trained on textual features alone outperforms the baseline from prior work by a solid margin. For further context, we additionally see that this version of the ranker's performance (e.g., *nDCG@5* = 0.82) is comparable to past research in the web search domain that also involved defining new problems (e.g., [6, 38]). The results in row 3 are also quite informative. They show that adding visual features *decreases* ranking quality. Our hypothesis here is that unlike is the case for many other image retrieval problems, our images can be visually similar while being semantically quite different. That is, data visualizations of the same type (e.g., bar charts) look roughly the same, but almost always cover substantially different topics.

Focusing our attention on our textual features, we further investigated different permutations of these features. The combination of **content-WPtitle** (semantic relatedness between news content and Wikipedia article) and **content-caption** (semantic relatedness between the news content and image caption) works the best. A ranker trained with these two features achieved a 0.82 nDCG@5, which is similar to the performance of the ranker trained on all textual features. The **title-caption** feature (SR between title and the caption), however, appears to be less effective. As such, for reasons of both performance and parsimony, we use the model trained with just **content-WPtitle** and **content-caption** in the final VizByWiki system. With this model, for instance, the system was able to give a top rank to the visualization in Figure 3 for the article in Figure 3 (as indicated by the score in Figure 3).

6 DISCUSSION

The above evaluations showed that VizByWiki is able to retrieve useful visualizations for up to approximately half of popular online news articles of diverse types and is able to rank them with reasonable quality. However, it is important to point out that the system has several notable limitations.

First and foremost, VizByWiki is limited by the quality of data in Wikimedia Commons. Despite being the largest repository of its type, the Commons suffers from the same metadata standardization issues that are common to all peer-production systems [9]. Fortunately, with the recently announced multi-year project aiming to standardize data in Wikimedia Commons [40], chances are that metadata quality will improve, making VizByWiki more effective in the future.

Second, VizByWiki sometimes recommends visualizations with older data. Relatedly, it also sometimes fails to retrieve data visualizations for news that has very recently broken. There are two potential causes here. First, VizByWiki is using a static snapshot of the Commons' images, and this may have resulted in us serving older versions of visualizations than currently exist in the Commons. This problem could be addressed with a larger-scale deployment using real time Commons and Wikipedia data. Second, it may be that there is a lag between an event occurring and Wikipedia editors updating their visualizations to include the new data. Future work should examine the lag time for visualization generation, as has been done for text (which found lag time be relatively small [15]).

Third, we observed that VizByWiki works better on some articles than on others. Future work should seek to answer questions such as: What characteristics make a news article suitable for data visualizations? What is the availability of different types and different topics data visualizations on Wikimedia Commons and on the Web more generally?

Finally, crowdsourced visualizations might not conform to specific aesthetic requirements from publishers. However, there is a promising opportunity to chain VizByWiki with other pipelines that reverse-engineer data visualizations (e.g., [14, 27, 35]) to support a complete process of finding data visualization, extracting data and redesigning the graphics.

7 CONCLUSION

To address the challenge of automatically generating large numbers of data visualizations for news articles, this paper defined the **news data visualization retrieval problem,** which involves mining data visualizations from the web to enhance news articles. We showed that this problem was tractable by designing an **end-to-end system, VizByWiki**, which mines a powerful-yet-untapped corpus of data visualizations: Wikimedia Commons. We evaluated VizByWiki using popular online news articles of different types. We found that the system could retrieve useful visualizations for many popular articles and that it could achieve satisfying ranking quality. To facilitate further progress on the news data visualization system, we are releasing a demo of the system, our ground truth data, and our source code (see URL on p. 2).

ACKNOWLEDGEMENTS

This work was funded in part by the U.S. National Science Foundation (IIS-1702440, IIS-1707319, CAREER IIS-1707296, and IIS-1421438).

REFERENCES

[1] Agrawal, R. et al. 2011. Enriching textbooks with images. *Proceedings of the 20th ACM international conference on Information and knowledge management* (2011), 1847–1856.

[2] Baack, S. 2011. A new style of news reporting: Wikileaks and data-driven journalism. *Cyborg Subjects*. (2011), 10.

[3] Chang, S. et al. 2015. Got Many Labels?: Deriving Topic Labels from Multiple Sources for Social Media Posts Using Crowdsourcing and Ensemble Learning. *Proceedings of the 24th International Conference on World Wide Web* (New York, NY, USA, 2015), 397–406.

[4] Cheng, X. and Roth, D. 2013. Relational inference for wikification. *Urbana*. 51, 61801 (2013), 16–58.

[5] Delgado, D. et al. 2010. Assisted News Reading with Automated Illustration. *Proceedings of the 18th ACM International Conference on Multimedia* (New York, NY, USA, 2010), 1647–1650.

[6] Ensan, F. and Bagheri, E. 2017. Document Retrieval Model Through Semantic Linking. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2017), 181–190.

[7] Gabrilovich, E. and Markovitch, S. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *IJCAI* (2007), 1606–1611.

[8] Gao, T. et al. 2014. NewsViews: an automated pipeline for creating custom geovisualizations for news. (2014), 3005–3014.

[9] Hall, A. et al. 2017. Freedom Versus Standardization: Structured Data Generation in a Peer Production Community. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), 6352–6362.

[10] Howard, A.B. 2014. The Art and Science of Data-Driven Journalism. (2014).

[11] Hullman, J. et al. 2013. Contextifier: Automatic Generation of Annotated Stock Visualizations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), 2707–2716.

[12] Ipeirotis, P.G. et al. 2010. Quality Management on Amazon Mechanical Turk. *Proceedings of the ACM SIGKDD Workshop on Human Computation* (New York, NY, USA, 2010), 64–67.

[13] Joachims, T. 2002. Optimizing Search Engines Using Clickthrough Data. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2002), 133–142.

[14] Jung, D. et al. 2017. ChartSense: Interactive Data Extraction from Chart Images. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), 6706–6717.

[15] Keegan, B. et al. 2013. Hot Off the Wiki: Structures and Dynamics of Wikipedia's Coverage of Breaking News Events. *American Behavioral Scientist*. 57, 5 (May 2013), 595–622. DOI:https://doi.org/10.1177/0002764212469367.

[16] Kittur, A. et al. 2008. Crowdsourcing User Studies with Mechanical Turk. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), 453–456.

[17] Li, W. and Zhuge, H. 2014. Summarising news with texts and pictures. *Semantics, Knowledge and Grids (SKG), 2014 10th International Conference on* (2014), 100–107.

[18] Li, Z. et al. 2016. Multimedia News Summarization in Search. *ACM Trans. Intell. Syst. Technol.* 7, 3 (Feb. 2016), 33:1–33:20. DOI:https://doi.org/10.1145/2822907.

[19] Li, Z. et al. 2011. News contextualization with geographic and visual information. *Proceedings of the 19th ACM international conference on Multimedia* (2011), 133–142.

[20] Li, Z. 2017. Understanding-Oriented Multimedia News Summarization. *Understanding-Oriented Multimedia Content Analysis*. Springer, Singapore. 131–153.

[21] Lin, Y. et al. 2017. Problematizing and Addressing the Article-as-Concept Assumption in Wikipedia. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (New York, NY, USA, 2017), 2052–2067.

[22] Mackinlay, J. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.* 5, 2 (Apr. 1986), 110–141. DOI:https://doi.org/10.1145/22949.22950.

[23] Marcus, A. et al. 2013. Data In Context: Aiding News Consumers while Taming Dataspaces. *DBCrowd 2013*. 47, (2013).

[24] Mihalcea, R. and Csomai, A. 2007. Wikify!: Linking Documents to Encyclopedic Knowledge. *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management* (New York, NY, USA, 2007), 233–242.

[25] Noraset, T. et al. 2014. WebSAIL Wikifier at ERD 2014. *Proceedings of the First International Workshop on Entity Recognition & Disambiguation* (New York, NY, USA, 2014), 119–124.

[26] Parasie, S. and Dagiral, E. 2013. Data-driven journalism and the public good:"Computer-assisted-reporters" and "programmer-journalists" in Chicago. *New media & society*. 15, 6 (2013), 853–871.

[27] Poco, J. and Heer, J. 2017. Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images. *Computer Graphics Forum*. 36, 3 (Jun. 2017), 353–363. DOI:https://doi.org/10.1111/cgf.13193.

[28] Ramisa, A. et al. 2016. Breakingnews: Article annotation by image and text processing. *arXiv preprint arXiv:1603.07141*. (2016).

[29] Razavian, A.S. et al. 2014. CNN Features off-the-shelf: an Astounding Baseline for Recognition. *arXiv:1403.6382 [cs]*. (Mar. 2014).

[30] Ren, D. et al. 2014. iVisDesigner: Expressive Interactive Design of Information Visualizations. *IEEE Transactions on Visualization and Computer Graphics*. 20, 12 (Dec. 2014), 2092–2101. DOI:https://doi.org/10.1109/TVCG.2014.2346291.

[31] Satyanarayan, A. and Heer, J. 2014. Lyra: An Interactive Visualization Design Environment. *Computer Graphics Forum*. 33, 3 (Jun. 2014), 351–360. DOI:https://doi.org/10.1111/cgf.12391.

[32] Savva, M. et al. 2011. Revision: Automated classification, analysis and redesign of chart images. *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), 393–402.

[33] Segel, E. and Heer, J. 2010. Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics*. 16, 6 (2010), 1139–1148.

[34] Sen, S. et al. 2014. WikiBrain: Democratizing Computation on Wikipedia. *Proceedings of The International Symposium on Open Collaboration* (New York, NY, USA, 2014), 27:1–27:10.

[35] Siegel, N. et al. 2016. FigureSeer: Parsing Result-Figures in Research Papers. *Computer Vision – ECCV 2016* (Oct. 2016), 664–680.

[36] Szegedy, C. et al. 2016. Rethinking the Inception Architecture for Computer Vision. (2016), 2818–2826.

[37] Tsikrika, T. et al. 2011. Overview of the Wikipedia Image Retrieval Task at ImageCLEF 2011. *CLEF (Notebook Papers/Labs/Workshop)* (2011).

[38] Wang, P. et al. 2017. Concept Embedded Convolutional Semantic Model for Question Retrieval. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2017), 395–403.

[39] WikiBrain: Advanced SR usage.: *https://shilad.github.io/wikibrain/tutorial/advancedsr.html*. Accessed: 2017-10-31.

[40] Wikimedia Foundation 2017. Wikimedia Foundation receives $3 million grant from Alfred P. Sloan Foundation to make freely licensed images accessible and reusable across the web. Retrieved from https://blog.wikimedia.org/2017/01/09/sloan-foundation-structured-data/

[41] Wongsuphasawat, K. et al. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics*. 22, 1 (Jan. 2016), 649–658. DOI:https://doi.org/10.1109/TVCG.2015.2467191.

[42] 2017. Help:Adding image. *Wikipedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=Help:Adding_image&oldid=764170156