UNIVERSITY OF BERGEN

Department of Information Science and Media Studies

MASTER'S THESIS

Developing and Comparing Similarity Functions for the News Recommender Domain Using Human Judgments

Author: Sebastian Øverhaug Larsen

Supervisor: Assoc. Prof. Dr. Christoph Trattner

June 15, 2020

Abstract

Similar item recommendations—a common starting point in various domains—provide users with similar items based on a reference item. These rely on *similarity functions* that are usually designed for a specific domain, i.e. recipes or movies. In this work, similarity functions were designed for the news domain using human judgements of similarity to guide predictive models. Human judgements were collected through a user study, in which participants judged the similarity of ten pairs of news articles. These judgements were then benchmarked against various similarity functions and used to train different machine learning models that can be used as similarity functions to compare news articles. It was found that the investigated similarity functions that work well in other domains overall correlate weakly to the human judgements of similarity, but that text-based similarity shows promise given the right metrics. In addition, the results from the current study within the news domain are compared to the results from another, wherein the focus was on the recipe and movie domains. Here, it was found that the different *types of features* have different degrees of importance in each domain and that similar metrics perform differently depending on the domain, according to correlation analyses against the provided human judgements. Overall, it is found that different domains call for different types of features and metrics, but that there exists some homogeneity.

Acknowledgment

I would like to thank everyone who supported me. I am thankful for the enlightening discussions and constructive criticisms I received throughout this master's thesis.

I am indebted to my supervisor Assoc. Prof. Dr. Christoph Trattner. I was inspired by his work, and he gave me the chance to undertake this research. On any day of the week, he provided me with invaluable advice and patient guidance.

I am very thankful for the grant I received from the DARS lab (https://dars.uib.no/), which alleviated the financial aspects of this research. Without it, this research would not have been possible.

Sebastian Øverhaug Larsen

Bergen, Norway, June 2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The news industry has undergone a significant transformation since the inception of the Web. News outlets can now publish or update news content instantaneously, and readers have instant access to it. However, the abundance of news content available can make it challenging for readers to read what they want when they want it. In addition, the news domain is a highly volatile environment with articles'[1] relevance changing rapidly, and users' interests changing dynamically [25]. The use of human judgments in recommender systems is not a new concept, however a relatively unexplored one. The primary benefits of this approach are the potential to learn in what way users perceive items to be similar, and to understand how to recommend items while achieving a minimal discrepancy between the recommended items, and the rating of similarity as judged by users. For news, this has scarcely been explored, while in other domains, researchers have started leveraging this approach to better understand how to recommend items [47, 46]. In one approach, the goal was to understand which specifically designed algorithms best represented the perceived similarity [47], and in the other to understand the parameters of a regression-based recommendation algorithm [46]. In the news domain, a study was conducted to understand how humans judge the similarity between news articles based on news titles [45]. In Trattner and Jannach [46], the approach showed its viability in the recipe and movie domains, and they suggested it to be used as a blueprint in other domains.

[1] For simplicity, an article can refer to an online news blog or article.

1.2 Problem

This thesis is a consequence of a problem that is continuously being addressed in the field of news recommender systems—what is the best approach to recommending similar news articles? To further understand the domain and how we can recommend similar news articles, this thesis undertakes an approach where existing metrics are explored for the news domain and compared across domains. Additionally, it explores features rarely used in earlier news recommender scenarios. These metrics and features are combined to create *similarity functions*, and benchmarked against human judgements of similarity. The problem statement of this thesis is thus as follows:

Given a reference news article and a set of other potential similar news articles, which similarity functions and features should be used to compute the most similar articles for the given reference article?

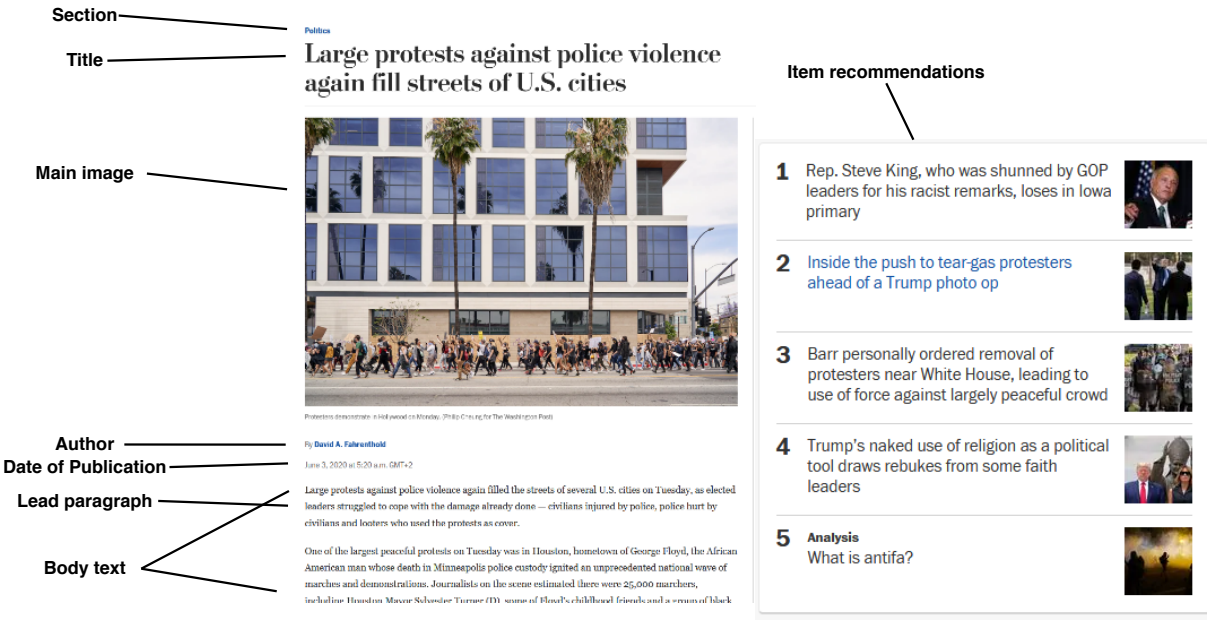

Figure 1.1: Illustration of the problem at hand - which feature do readers use to determine similarity between articles?

1.3 Objectives

The primary goal of this master's thesis is to understand which metrics and features best represent human judgments of similarity. To this end, the following research questions are addressed:

- **RQ1:** Which *types of features*, and which *specific features* best determine the similarity between items as perceived by users? In Section 4.2, analyses are conducted to understand the correlation between users' perception of similarity and the scores provided by similarity functions.
- **RQ2:** Which *combination of features* is best suited for predicting user-perceived similarity levels? In Section 4.3, an offline experiment is conducted based on the data obtained in the study. In this experiment, different machine learning models are constructed, and their prediction accuracy compared.
- **RQ3:** How do we compare to the recipe and movie domains? In Section 4.4, the results from the current study are compared against the work of Trattner and Jannach [46].

1.4 Contribution

A goal of this master's thesis is to learn similarity functions to recommend similar news articles with a minimal discrepancy to human's perception of similarity. In other domains, studies have been conducted that revealed the benefit of using human's perception of similarity to achieve this [46, 47]. As such, this thesis further explores this approach by extending it to another domain, and the contributions are therefore as follows:

- To conduct the study, data from the Washington Post was processed to be more usable in the context of the approach. Therefore, a data processing pipeline was developed to process the data, including converting the data to CSV-format, downloading images from the news items, as well as to compute the similarity between the items.
- The thesis provides a better understanding of how readers perceive similarity between news, in terms of (i) what information cues are reported as important, as well as (ii) how the various information cues correlate to the ratings provided by the user study

participants. (iii) The results show that the importance of the different information cues reported by the users are not always in line with computed correlations. (iv) In addition, it reveals that information cues rarely observed in earlier news recommendation scenarios can be of value given the right metrics.

- Insight is provided into the predictive performances of the various information cues available in news content. It shows that there are clear distinctions in terms of what makes for a good indicator of similarity according to users.

- Lastly, it provides an extended insight into the novel approach proposed by Trattner and Jannach [46], by providing a comparison of the results of the current study against theirs. The comparison analysis further emphasizes the differences in how users perceive similarity across domains.

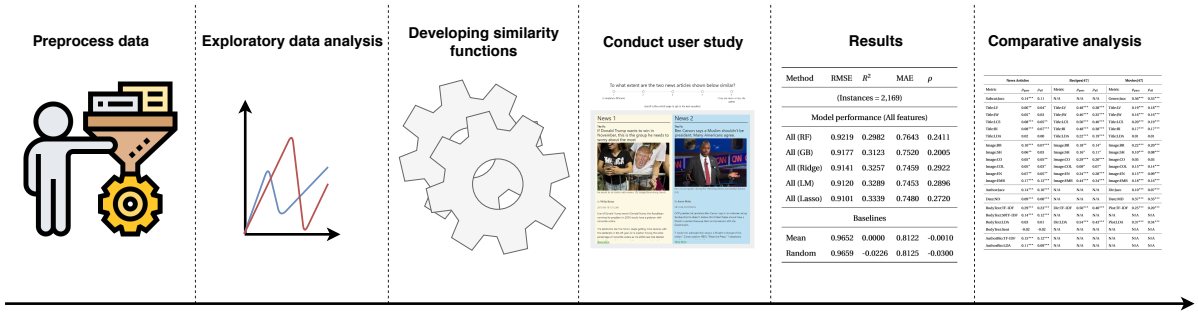

Figure 1.2: Schematic illustration of the thesis.

1.5 Relevance of this Work

- Understanding what makes news articles similar is an important aspect for a news recommender system.

- While many product or service providers utilize recommender systems, many online news outlets can be hesitant to do so due to lack of trust in the systems. This study provides better insight into how a recommender system for the news domain should be designed.

- Further explores Trattner and Jannach's [46] approach, which they see as a blueprint for further research into this line of research.

1.6 Thesis Outline

This master's thesis is split into five Chapters. This introduction Chapter is followed by the background (Chapter 2), which reviews work relevant to this thesis. The background gives a brief overview of how similar item recommendations can be computed, work that has been conducted to understand the news domain, and an overview of work conducted using human judgments as an optimum standard. Chapter 3 describes the data and methods used in this thesis. It provides insight into the structure of the data, how it was processed, how it was improved for use in this thesis, and describes the methods of computing similarity and conducting the user study to collect human judgments. Chapter 4 presents and discusses the results of the study. It presents the results of the correlations between the similarity functions and the human judgments, and the performance of the predictive models. Additionally, the results are compared to the results of Trattner and Jannach [46] in the recipe and movie domains. Finally, Chapter 5 discusses the conclusions of the study, limitations of this master's thesis, proposes future research directions, and describes the tools used in this thesis. Additionally, the Appendix provides further insight into the data and methods used in the current study. It also presents the submission to the ACM RecSys Conference[2] which is based on this research, and the author of this thesis was the second co-author.

[2] https://recsys.acm.org/

Chapter 2

Background

This Chapter attempts to give an overview of previous work relevant to the context of this thesis and is split into five sections. Section 2.1 describes the problem of recommending similar items and describes common approaches. Section 2.2 discusses the news domain in particular and the challenges that are present here. Section 2.3 sheds light on some of the approaches that have been conducted in news recommender scenarios. Section 2.4 gives an overview of the features and methods that have been observed in earlier news recommender scenarios. Finally, Section 2.5 describes related work where human judgments were used to recommend items, especially in regards to the work of Trattner and Jannach [46].

2.1 Similar Item Recommendations

At its core, a recommender system is a system that essentially attempts to support the decision-making of users. It attempts to do this by providing item suggestions, based on various data, such as preferences, demographics or items a user has interacted with in the past. What the system provides is generally denoted as an *item* or a *document*, but it can be a product a user can buy (i.e. a book), or a service (i.e. on-demand movie) [31].

The sheer amount of items and services offered online can be too difficult for the human mind to process efficiently. Within recommender systems, there are four main approaches used in building a system that can alleviate users: *collaborative filtering* (CF), *content-based* (CB), *knowledge-based* (KB), and *hybrid-based* (H) approaches [22]. CF recommends items by identifying users with similar preferences to that of a given user. CB recommends items

by identifying other, *unseen*, or *novel* items similar to those a given user has interacted with or specified that they prefer, in the past [31, 7, 22]. KB-approaches are based on domain expertise to map user preferences, and hybrid-approaches are based on a combination of CF and CB. CB-approaches employ features that are domain-specific (e.g. recipe ingredients in recipes) to assess the similarity between different items [46]. The use of features is formalized in various *similarity functions* [47]. Since these item-based approaches are based on existing documents, they do not suffer from cold-start problems as much as approaches that are based on user activity [11].

A common approach is to derive *vectors* from items a user has liked in the past, and from items found within the system. *Term frequency-inverse document frequency* (TF-IDF) is a *vector space model* commonly used to create such vectors: $TF-IDF(t,d,D) = tf(t,d) * idf(t,D)$, where $tf(t,d)$ denotes the number of times a term appears in a document, and $idf(t,D)$ denotes the number of documents a term appears. Subsequently, the similarity between the vectors of liked and unseen items can be computed using Cosine similarity: $sim = \frac{A*B}{||A||||B||}$. [4].

In a simpler approach, a set of keywords can be derived from an item [22]. For example, a book recommender could compute the similarity between $book1 = fantasy, epic, bloody$, and $book2 = fantasy, young, dragons$, using the *Jaccard coefficient* as follows: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ [37], where A denotes $book1$ and B denotes $book2$. Depending on the task, there are various similarity metrics available, such as Dice coefficient [22], the Levenshtein distance (also called the *edit distance*), LDA (Latent Dirichlet Allocation), etc. TF-IDF is one of the most commonly used methods in information-retrieval scenarios. Although it has been outperformed by other measures such as BM25 [34], it is still used regularly [46, 4]. Furthermore, in similar item computations, Cosine similarity has been used to predict rating values [46].

2.2 The News Context

The news domain has been found to be a more volatile domain than most others [25]. This is because interest in different topics can vary greatly among readers, and can change over short or long-term periods for any reader. For example, users can be interested in different topics during a weekend than during weekdays [25]. While users in domains such as movies tend to

dislike "too-obvious" recommendations [47], news readers could be interested in learning more about the topic of a given news article [25]. Furthermore, news articles tend to decay fast in interest, but it is nonetheless suggested that "old articles" should not be blindly filtered out [25].

Many approaches have been proposed to address the challenge of *recency* or *freshness* of a news article. There exist three stages in which this can be addressed: pre-filtering, recency modeling, or post-filtering [25]. Pre-filtering refers to filtering out news found to be outdated before computing predictions or ranking items. recency modeling, the most common approach, involves incorporating the recency factor into the algorithms that compute the predictions. Lastly, post-filtering happens after the main process. recency modeling is the most common approach, likely due to its distinct advantage where the different factors in a similarity computation can be balanced more easily [25]. Pon et al. [39] proposed a recency modeling approach, where recency was considered along with a "multiple topic tracking" technique, targeted at users with several interests. Short term topic interests were accounted for by computing the similarity of the recently consumed news articles. In another approach, the recency of an article was considered as an item feature, and recent news articles were given a higher weight in the ranking process [25].

Much research has been conducted to understand the effect of emotion in the news. It has been suggested that negativity in news has a great impact on the reader——indeed, the emotional reaction of a reader lasts longer than readers are able to remember details in a media story [44]. Beyond this, emotion has been leveraged for use in numerous information processing theories and models, i.e. for motivated reasoning [44]. Thus, Soroka et al. [44] posited that there exists good reasons to believe that emotion and affect are central in political reasoning and can be important to consider to understand the source of people's information, i.e. political news content. To better understand the emotions behind political news, one can use sentiment analysis, often colloquially called opinion mining along with i.e. subjectivity analysis. Soroka et al. [44] define sentiment as a "... broad construct comprising attitudes, opinions, and emotions, where (1) attitudes refer to positive or negative evaluations, (2) opinions refer to judgments and beliefs, and (3) emotions refer to feelings." Thus, the aim of sentiment analysis is to detect these three aspects, and it can be applied to texts to infer the attitude, opinion, or affective state of the writer. To the knowledge of the author, sentiment analysis has not yet been used to find similar news articles given a news article, but has been

leveraged in i.e. product recommendation by opinion mining product reviews [13].

2.3 News Recommender Systems

News recommender systems primarily focus on textual representations of articles. They are usually geared towards the utilization of an article's body text or title, and other textual features such as the author are often ignored [4, 25]. While images are often used in some domains (e.g., recipes and movies [46]), they are used much less frequently in news [25]. Moreover, an article's date of publication is also used less frequently than the body text or the title [25, 30], despite novelty being reported as a particularly important aspect in news recommender scenarios [25]. Both image and date (i.e. release date, date of publication) features have been noted to be of particular importance for cross-domain comparisons [46].

A common approach in news recommender scenarios is to use topic models to derive latent topics from texts, through methods such as Latent Dirichlet Allocation (LDA)[33, 29, 16], and Probabilistic Latent Semantic Indexing (PLSI) [29]. For example, Li et al. [29] employed a two-stage approach, where the first stage involved using both LDA and PLSI in separate experiments to cluster topically-similar news articles together. In the second stage, different methods were applied to refine the recommendations, i.e. by assigning recency scores to news articles. Here, recency was considered after the main process and is in such cases called *post-filtering* [25].

In a different approach, TF-IDF has been just in conjunction with the K-Nearest Neighbor algorithm to recommend short-term interest news articles to individual users [3]. Here, news articles were converted to TF-IDF vectors, and Cosine similarity was used to measure the similarity of two vectors. K-Nearest Neighbor was then used to identify articles that belong to the same *threads of events*, and that a user already knows. In addition, long-term interests were identified by using a Naïve Bayesian classifier, which had been shown to perform competitively with more complex algorithms.

TF-IDF in combination with Cosine similarity is a traditional method of recommending news articles [9, 19]. Indeed, it is one of the most common methods to find approaches of various kinds leverage and has been used as a benchmark to test other methods against [6, 9, 43]. In two approaches, experiments were conducted to understand whether TF-IDF

suffers from processing very long documents [6, 43]. Articles are written in an inverted pyramid style, meaning that the most important information is found at the start of an article [6]. From this, Bogers and Van Den Bosch [6] posited that constraining the length of articles may boost TF-IDF performance. The results showed that there is indeed a drop-off, however insignificant, in performance as texts grew longer, which was in line with the findings of Singhal et al. [43]. Another challenge present in using TF-IDF is that it does not capture the meaning of words. One approach attempted to solve this by developing a new method based on TF-IDF, called *Synset frequency-inverse document frequency* (SF-IDF) [9]. Instead of counting how often a term appears, synset frequency counts the number of times a word appears that is interchangeable with another without losing its meaning. Similarly, Goossen et al. [19] proposed another TF-IDF-based approach called *Concept frequency-inverse document frequency* (CF-IDF). Here, term frequency is replaced by counting the number of times a concept appears, i.e. "google". Both SF-IDF and CF-IDF were found to out-perform the traditional TF-IDF method.

2.4 Features Used in Similar News Recommendation

Earlier news recommender approaches are found to primarily focus on textual representations of news articles, and usually ignore media such as images. Furthermore, the approaches are usually geared towards utilizing the body text, title, or all text of the news articles, and ignore most other textual features such as the author [4, 25].

Table 2.1: Features and methods used in earlier content-based similar news recommendation scenarios. Features and methods used in the current study are marked with bold.

Feature	**Description & Relevant Articles**
Title	Okapi BM25, Language model Jelinek-Mercer (LM-JM), Language model Dirichlet prior (LM-DIR), **Cosine similarity** [34]; **TF-IDF** [48]; Dependency structure language model (DSLM) [40]
Body text	Okapi BM25, Language model Jelinek-Mercer (LM-JM), Language model Dirichlet prior (LM-DIR), **Cosine similarity** [34]
Abstract	Okapi BM25, Language model Jelinek-Mercer (LM-JM), Language model Dirichlet prior (LM-DIR) [34]
All text	**TF-IDF** & K-Nearest Neighbor [3, 4, 21]; **Cosine Similarity**, Naïve Bayes [4]; Overlap Coefficient [10]; Probabilistic Latent Semantic Indexing (PLSI) [29]; **Latent Dirichlet Allocation** [33, 29, 16]; Fisher Kernel Function (PLSA) [32]; Dependency structure language model (DSLM) [40]
Image labels	Image-label overlap similarity [30]
Date of publication	Pre-filtering [15, 12, 27]; **recency modeling** [39, 14, 18, 28, 2, 16, 36]

Table 2.1 presents an overview of features and some of the metrics used in earlier news recommender scenarios. In previous work, it was found that short descriptions of news articles, such as title and abstract, are too compressed to represent the news articles' information [34]. For example, Yuanhua et al. [34] found that the main text (i.e. body text) of news articles is better suited for finding similar articles.

2.5 Human Perception of Similarities (Similarity Functions)

Tintarev and Masthoff [45] conducted a study to better understand similarity of news. As part of the study, they investigated how humans judge the similarity of news articles, based only

on headlines (i.e. titles). In this experiment, the participants were shown nine pairs of news articles. For each pair, the participants were asked three questions regarding the similarity, in which they were to answer on a seven-point Likert scale. These questions related to (i) how related the articles are, (ii) if an acquaintance is interested in article A, how sure are they that their acquaintance is interested in article B, and (iii) how much new information might article B provide given that you have read article A. The article headlines were obtained from Google news[1], and from various different categories, such as Entertainment, Science and Technology, Sports, and more. Their experiment primarily showed that users are more often than not able to identify identical articles with different headlines.

Yao and Harper [47] conducted a study in which they collected more than 22,000 human judgments of movie pairs. They used different CB and CF methods to measure whether similar item recommendations were able to match the human judgments of similarity. Their study involved an algorithm-centric and a user-centric research question: RQ-ALG - "Which related item algorithms best match user perceptions of relatedness and recommendation quality?", and RQ-UX - "How should related item algorithms be designed to improve the user experience?", respectively. Their work contrasts previous work, which mostly entails optimizing input to a collaborative filtering algorithm, or optimizing business outcomes with click-through rates [17, 47]. Their user study was divided into two parts: a survey in which they asked the participants questions relating to the manner in which MovieLens recommends movies; and a survey in which participants were shown pairs of movies, and were asked to what extent the movies are similar, and whether they would recommend the second movie to someone who likes the first.

In answering RQ-ALG, Yao and Harper [47] found that content-based algorithms are the superior approach to match user expectations. Furthermore, they found that free text works better than tags. One of their key findings was that there is a trade-off between item similarity and user relevance; users do not necessarily want the most similar items. They believe that *related item recommenders* should be content-based. Regarding RQ-UX, Yao and Harper [47] found that *related item recommendation* plays an important role in a recommender system. Study participants rated related item recommendations to be more important than an overall recommendation or per-genre recommendation.

[1] https://news.google.com

Trattner and Jannach [46] conducted a study where they employed a novel approach to train and validate similarity functions. Their study was based on using human judgments of item similarity as ground truths for (i) how similar two items are, and (ii) what makes two items similar. In previous work, human judgments have been used for similar item generation in other domains, though primarily in the music domain. However, datasets generated from user studies in such work were mainly used to ascertain what makes two items similar and were focused on evaluating already existing approaches. Trattner and Jannach's [46] aim was to systematically train similarity functions in order to understand which features and metrics correlate with human estimates. In the music domain, user studies have been focused on asking participants for broad assessments of similarity (i.e. how similar are these two songs), and on asking participants to disregard particular song features prior to their assessment (i.e. how similar two songs are besides common instruments) [1, 24]. Trattner and Jannach [46], however, specifically asks participants which features were important in their similarity assessment.

Initially, Trattner and Jannach [46] conducted user studies on the platform Amazon Mechanical Turk[2]. Participants were asked to assess how similar two objects are, and to which degree the different features (i.e. title and image) played a role in their assessment. The data was fed to 17 similarity functions of different metrics and features, and they conducted offline evaluations of how well the models perform. In both domains, they found that a combination of all predictor variables (features), using Ridge regression, was the model that led to the highest accuracy [46]. Furthermore, they conducted additional user studies to validate the models in an online setting. Their goal was to validate that recommendations generated by combined similarity functions are also *perceived* by users to be similar, more so than recommendations based on individual cues. As part of their research questions, they wanted to discover whether high prediction accuracy (offline) led to a high perceived item similarity (online). Researchers often find that offline evaluations do not provide a real-world view of perceived similarity [17, 42].

Trattner and Jannach [46] conclusively states that their work demonstrates the feasibility of relying on human-generated judgments fed to similarity functions. However, they found that taking the human judgments under consideration is also a necessity, since "... experts can err and because self-assessments by users regarding the relative importance of certain

[2] https://www.mturk.com/

factors might be misleading." Offline evaluations showed great promise, and their validations through user studies further emphasized the feasibility of the approach, as well as suggests that offline evaluations can be viable in such a setting. They believe that their study can be used as a blueprint for further research into domains other than recipes and movies.

2.6 Summary of Previous Work and Key Differences

In many recommendation scenarios, standard methods such as TF-IDF, or the Jaccard coefficient, are still in use today. This is no different in the news domain, but many approaches rely on a modified version of i.e. TF-IDF. Due to the volatility of the news domain, many researchers find that we need to know the meaning of words to understand the similarity. Here, many approaches have been proposed, such as Concept-frequency or Synset-frequency instead of Term-frequency in the TF-IDF method. These approaches often share the same characteristics but leverage different resources. Here, the common TF-IDF method is used as normal with the body text of news articles, as well as in a method where the length of body texts are constrained. This is so that it is comparative to previous work in using human judgments [46], and to attempt to capture the most important information of a story, which is found in the beginning [6].

The only earlier work in news recommendation scenarios found to leverage human judgments is the work of Tintarev and Masthoff [45]. The key difference between their work and the current study is that they leverage only the headlines of articles, and were not concerned with which features make the most important factors for users. Their main goal was to better understand similarity in news, not which features make news similar.

Topic-modeling approaches such as LDA are popularly used in news recommendation scenarios (see Table 2.1). Most commonly, it is used as a means to cluster similar news articles together, often as part of a multi-stage approach. Here, LDA is paired with Cosine similarity to compute the similarity of pairs of news articles instead of grouping articles. Additionally, most approaches leverage only the title and, or the body text of news articles. Neither the author nor the date of publication features are often used relative to the title or body text. Here, all features presented to readers of the Washington Post are leveraged as a feature, paired with a measurement of similarity. This includes the author biography, which is seen as a description

of the author. recency modeling is found to be the most common approach to incorporate the date of publication of news articles. This approach is also used here, where a linear function is used to calculate the distance in days between two news articles.

Previous work shows that sentiment has been used to i.e. capture bias in news, mine opinions from news headlines, or boost item predictions. The various applications of sentiment, along with Soroka et al.'s [44] belief that sentiment is a strong indicator of readers' perception of political news, leads to the application of it in a similarity function later presented in the current study.

The work of Trattner and Jannach [46], and Yao and Harper [47] share similarities in that they both explore different algorithms' capabilities of approximating users' perception of similarity. However, differently from Yao and Harper[47], Trattner and Jannach [46] automatically learn different item features' different importance weights, instead of evaluating existing approaches in this area. Additionally, Trattner and Jannach [46] validate that their best-performing method from the offline evaluations also leads to a high similarity perception by users, by conducting additional user studies.

The key difference of the current study to Trattner and Jannch [46] is the domain in which the study is set in. While their work is based in the recipe and movie domains, this study is based in the news domain. Features available in the news domain play different roles than those available in the recipe and movie domains. Part of the goal in the current study is thus to understand how the metrics they developed for the recipe and movie domains perform when they are developed for the news domain. Additionally, their study involved a final step of validating their results with additional user studies. Here, the participants rated the similarity of pairs of items generated by their strongest predictive model. This step is not within the scope of the current study since the primary focus is on understanding the strength of existing metrics in the news domain.

Chapter 3

Methods

This Chapter describes the data and methods used in the current study, and is split into four sections. Section 3.1 provides an overview of steps taken in processing the dataset used in the current study. The process of entity engineering is then described in Section 3.2, where JSON-objects were transformed to more representative entities. Section 3.3 describes the statistics of the resulting dataset and the sample dataset later used to conduct the user study. Section 3.4 provides an overview of the developed similarity functions. Lastly, Section 3.5 describes the process of collecting human judgments through a user study.

This study uses the 2017-version of the TREC Washington Post Corpus [1], a JSON-formatted file comprised of 595,037 news articles. Each news article contains several JSON-objects, including a JSON-array (*article* in Table 3.1) which contains i.e. the body text of an article. The news articles contain HTML tags, including embeddings such as image, video, and tweets.

[1] https://trec.nist.gov/data/wapost/ - Note that since the start of this thesis, the dataset has been updated with articles from 2017 until 2019.

Table 3.1: Features available in the TREC Washington Post Corpus.

Feature	**Description**
Title	The title of the news article
Byline	Author of the news article
Date of publication	Date published
Kicker	Section header
Article	Article split into paragraphs
Links	Links to embedded images and multimedia

3.1 Dataprocessing

Figure 3.1 illustrates the processing of the data, up until the point at which a desired category is set. As the figure illustrates, several steps were taken to ensure quality in the dataset, and to make it more usable in the context of the user study. Thus, the process involved converting the dataset to CSV-format since it is faster to process. To this end, the first step was to design a data processing pipeline to convert the dataset, to preserve the structure of the data, as well as to enrichen it.

From reviewing data, an image-link found in the same JSON-object as the full title of an article were found to be the "main image" of an article. These were then added as a separate feature in each news article during conversion to CSV. In the event that an image could not be found alongside the title, the image-property previously described was left empty. In a different step, all image links found for each item, as well as the rest of the images found in each object, were stored in a separate CSV file. Each image was given the current article ID as filename, and suffixed with the order in which they were found in the news article. For future research purposes, all images found in each article were then downloaded, resulting in 655,533 images, and the main image could be identified by its suffix. Furthermore, some images were found to be corrupted after downloading. Since articles that did not have a main image according to this strategy, and corrupted images were removed, all articles during the sampling stage had a functional, main image.

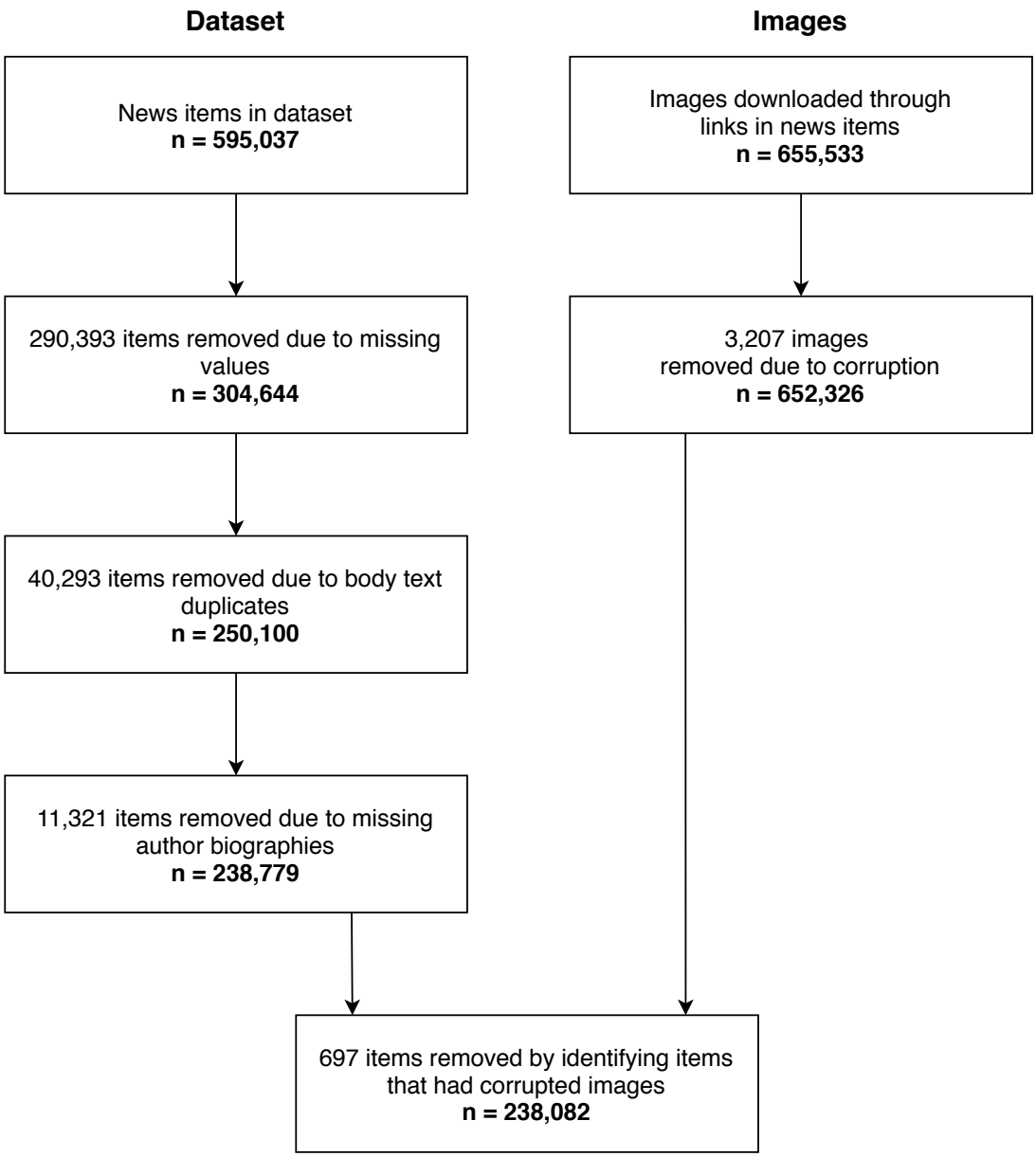

Figure 3.1: General process of removing incomplete data from the dataset.

HTML tags were removed for each object within the JSON-array to remove redundant styling However, since the structure of the news articles needed to be preserved, the objects were spaced using <*br*><*br*>, which results in two newlines in HTML. This mostly means that, while the structure in terms of headings and paragraphs is preserved, the texts no longer contains bold or italic faces. Additionally, media embeddings found in these objects were manually reconstructed by identifying them with the *type* property of each object. Instead of the format provided by i.e. Twitter or Vimeo, they were given a basic format without any aesthetic modifications.

The process resulted in a dataset 42.3% of the original size (from 6,99 GB down to 2,96 GB). Section 3.2 further describes how the dataset was enriched. See Table 3.2 for a complete overview of the features available in the processed dataset.

Table 3.2: Features available in the processed TREC Washington Post Corpus dataset.

Feature	**Description**
title	The title of the news article
author	Author of the news article
author_bio	The author's bio
date	Date published
time	Time of day published
id	The article's assigned ID
text	All text available from the article's body text
category	The general section of the Washington Post the article belongs to
subcategory	The original section of the article
article_url	The URL of the article
image_url	The URL of the title-image
type	The type of the article
subtype	The subtype of the article

3.2 Entity Engineering

This section describes the work that was done in engineering entities that better represent the components of an article in the dataset. The entities are created in such a way that they can be identified by their respective *categories* or *subcategories*. Additional entities were constructed to better represent the basic structure of the news articles, similarly to how they are presented on the Washington Post.

The original dataset does not contain properties that describe the general category of a news article, i.e. an article being a political or sports article. Instead, they contain properties annotated as a *kicker* which reflects the subsection of the Washington Post the news article was published in. Thus, to be able to use data from a specific category, the 163

unique subsections were manually mapped to their respective, general category, by reviewing the various sections found on the Washington Post's website. A list of each subsection, i.e. subcategory, observed in the dataset, and the category they were mapped to is presented in the Appendix on Table 3. This process is also further described later in Section 3.2.

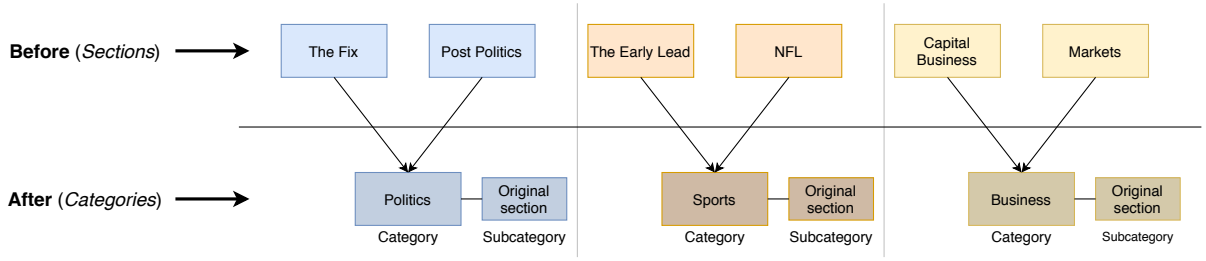

Figure 3.2: Example of categories before and after modification.

The *published date* entity was split into a *date* and *time* entity. The date entity, originally in UNIX-format, was converted to YYYY-MM-DD, a variation of the ISO 8601 format. By converting to YYYY-MM-DD, we can use individual parts of a *date* as a metric, i.e. a day, month, or year.

It was discovered that the *author* was sometimes missing, usually when the article in question is a *compilation*. Such articles always included a paragraph at the end stating "Compiled by", followed by one or multiple authors. These authors were extracted and set as authors of the appropriate articles.

Subtype was added as an entity. Subtype can have two values; "compilation" or "standalone". Subtype was set to "compilation" in the case of finding missing authors by the process previously described. If an article contains authors, or authors could not be extracted, then the subtype was set as "standalone".

Author biography was added as an entity since it is part of the Washington Post's article format. These entities describe the author's focus area, i.e. there are journalists who focus on specific categories or topics, e.g. "Peter Stevenson covers national politics for The Fix"[2], and journalists who work in a general capacity, e.g. "Lindsey Bever is a general assignment reporter for The Washington Post"[3]. Additionally, in cases of multiple authors of an article, there are

[2] https://www.washingtonpost.com/news/the-fix/wp/2016/09/08/does-body-language-really-give-trump-insight-into-intelligence-operatives-thoughts

[3] https://www.washingtonpost.com/news/morning-mix/wp/2014/11/03/how-brittany-maynard-may-change-the-right-to-die-debate-after-death

articles that do not include an author biography for each author. These were identified, and the corresponding articles ignored.

The image URL of an article was added as an entity by identifying article-objects of type *fullcaption*. These were found to contain the full title of a news article as well as the primary image URL, that is to say the image that is paired with the title at the top of an article.

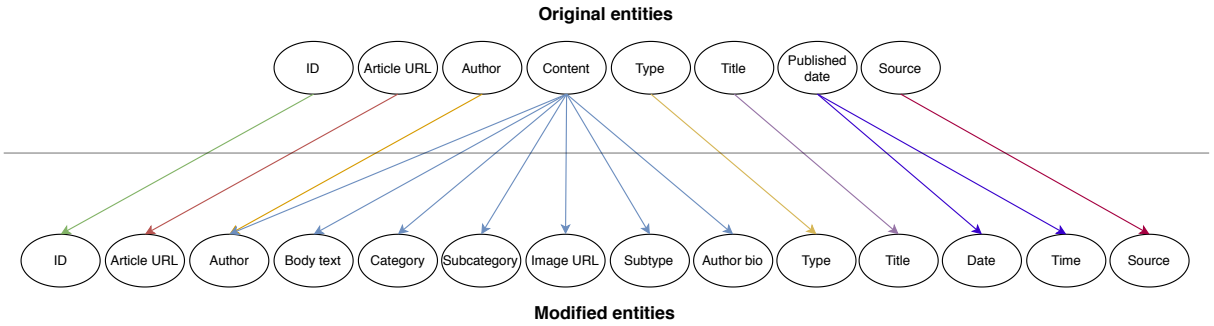

Figure 3.3: Before and after entity engineering.

3.3 Exploratory Data Analysis

This Section provides an overview of the general statistics of the processed dataset, and of the sample later used in the user study. It also provides a reasoning for why "Politics" was the chosen category for this study. Data presented here are without missing values, without duplicates by body text, and without articles found to have a corrupted main image, as illustrated in Figure 3.1.

3.3.1 Overview of the Processed Dataset

Figure 3.4 presents the category distribution in the data. "Sports" and "Politics" stand out as the largest in terms of number articles, with the latter having more than twice as many news articles as the third largest, "D.C., Md. & Va.", which is the Washington Post's local news category. In the original dataset, the items can only be categorically separated by non-descriptive names such as "The Fix" (Politics), "The Early Lead" (Sports), and "Act Four" (Opinions). By mapping these to their respective, general categories, it enables us to choose a category to continue with as we wish.

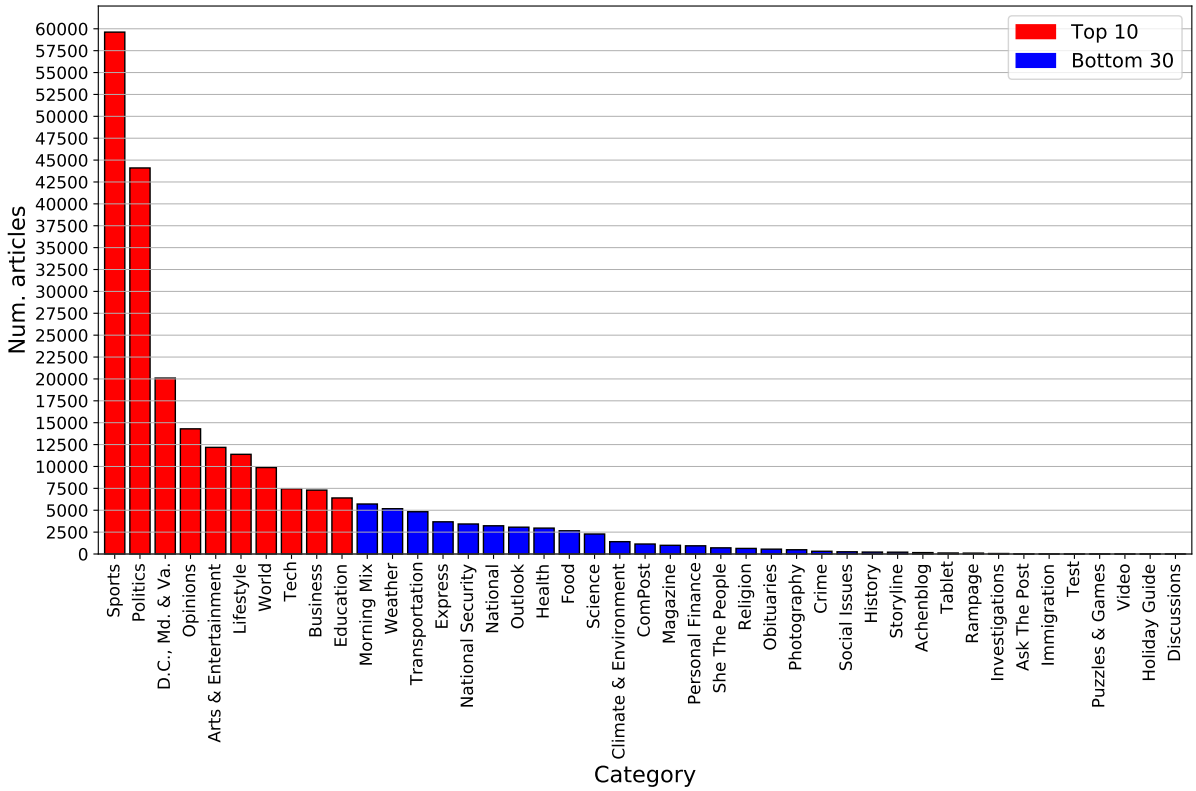

Figure 3.4: Category distribution in the processed TREC Washington Post Corpus.

Figure 3.5 illustrates the distribution of articles published between January 2012 until August 2018. Looking at the number of articles published over time, we see that it increases as the weeks progress, and declines as the weekends approach. We can also see a steady increase in number of articles published as the years progress.

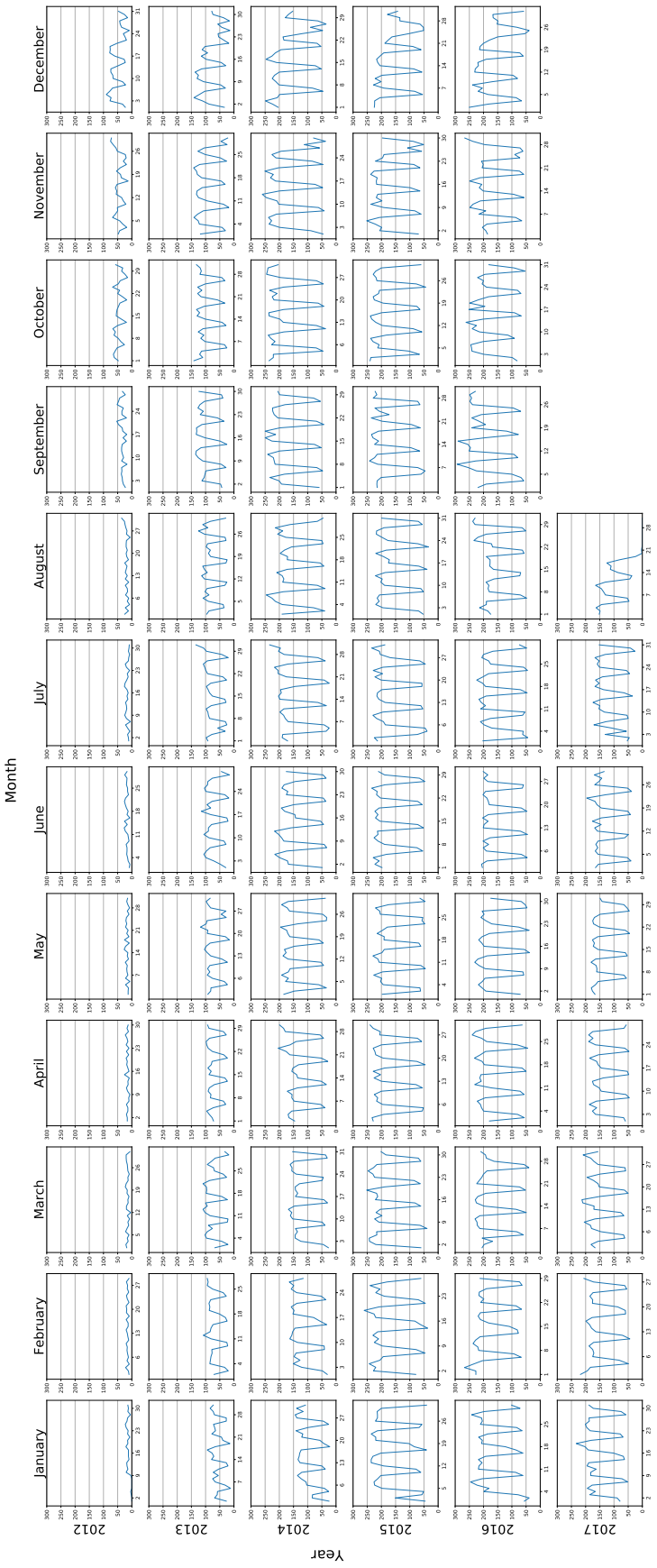

Figure 3.5: Date of publication distribution of articles from January 2012 until August 2018. In each sub-graph, the x-axis marks Mondays in the given month.

3.3.2 Choosing a Category

Figures 3.6, 3.7, and 3.8 presents the subcategories within the three largest categories "Politics", and "Sports", and "D.C., Md. & Va.", respectively. In choosing a category, it was important that (i) it contains enough articles so that an evenly distributed sample can be obtained, and (ii) the content is as little localized as possible, meaning that the content does not require local knowledge to understand it. During mapping of the subcategories, several seemingly local subcategories were found to belong to a national category, i.e. "Washington Nationals" in "Sports", a section of the Washington Post about baseball teams in Washington. On the other hand, news articles from the "Politics" category seem to generally be about either strictly national or international politics, as none of the subcategories present in Figure 3.6 are focused on local politics. Local political news articles are instead found in the local category presented in Figure 3.8, i.e. "Maryland Politics". While sports teams certainly have fans from more than just its place of origin, it is nonetheless argued that "Sports" can require more local knowledge than "Politics", given Washington Post's method of sectioning these. Thus, national and international political news articles are chosen as the point of focus in the current study.

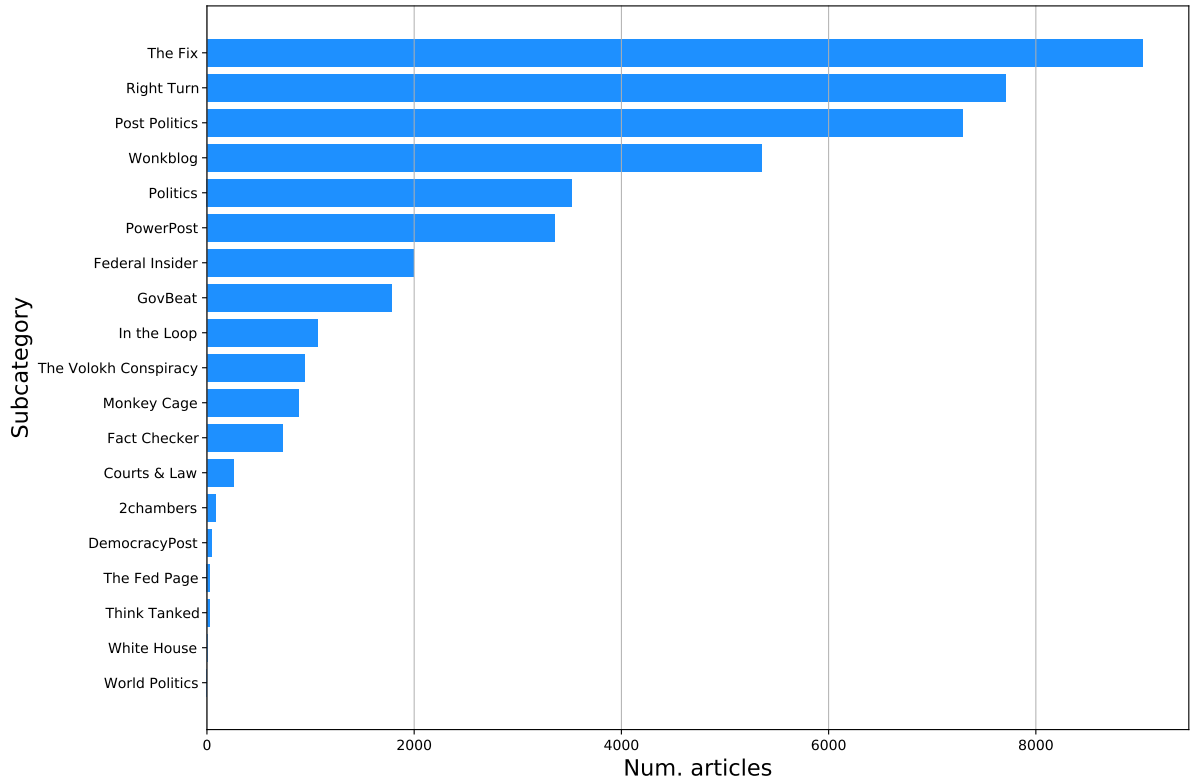

Figure 3.6: Number of articles for each subcategory in the "Politics category.

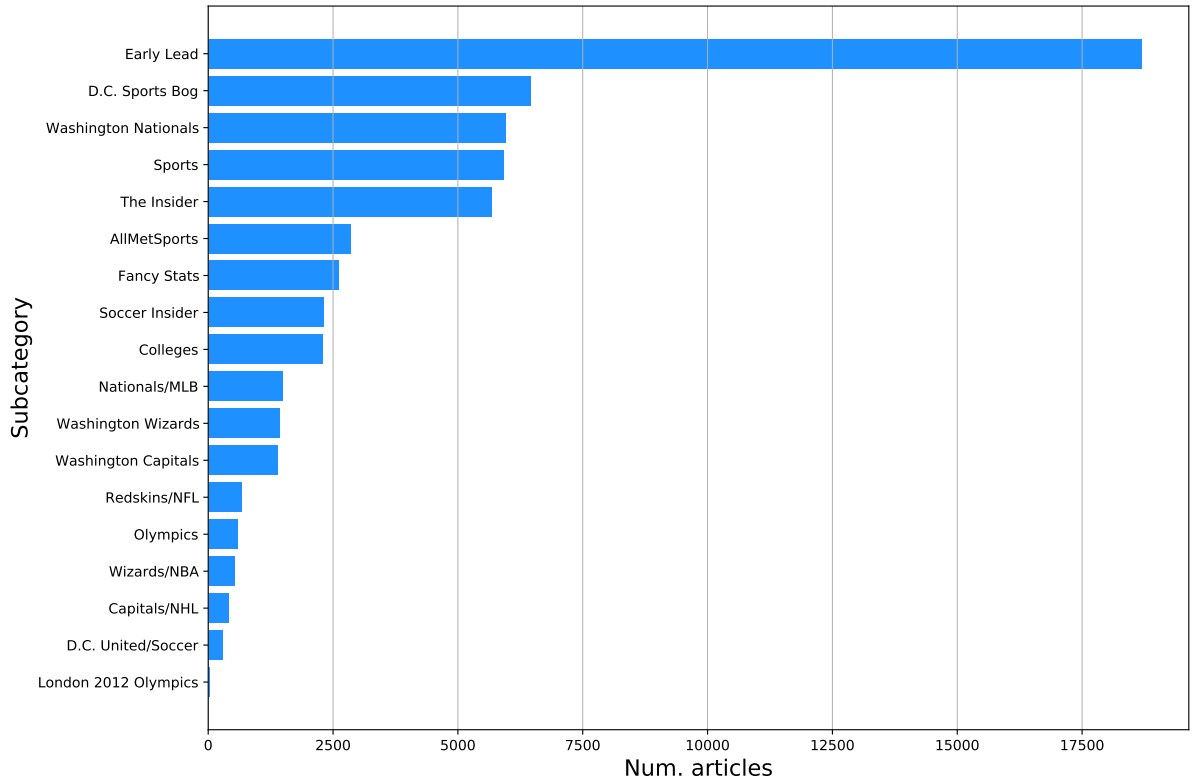

Figure 3.7: Number of articles for each subcategory in the "Sports" category.

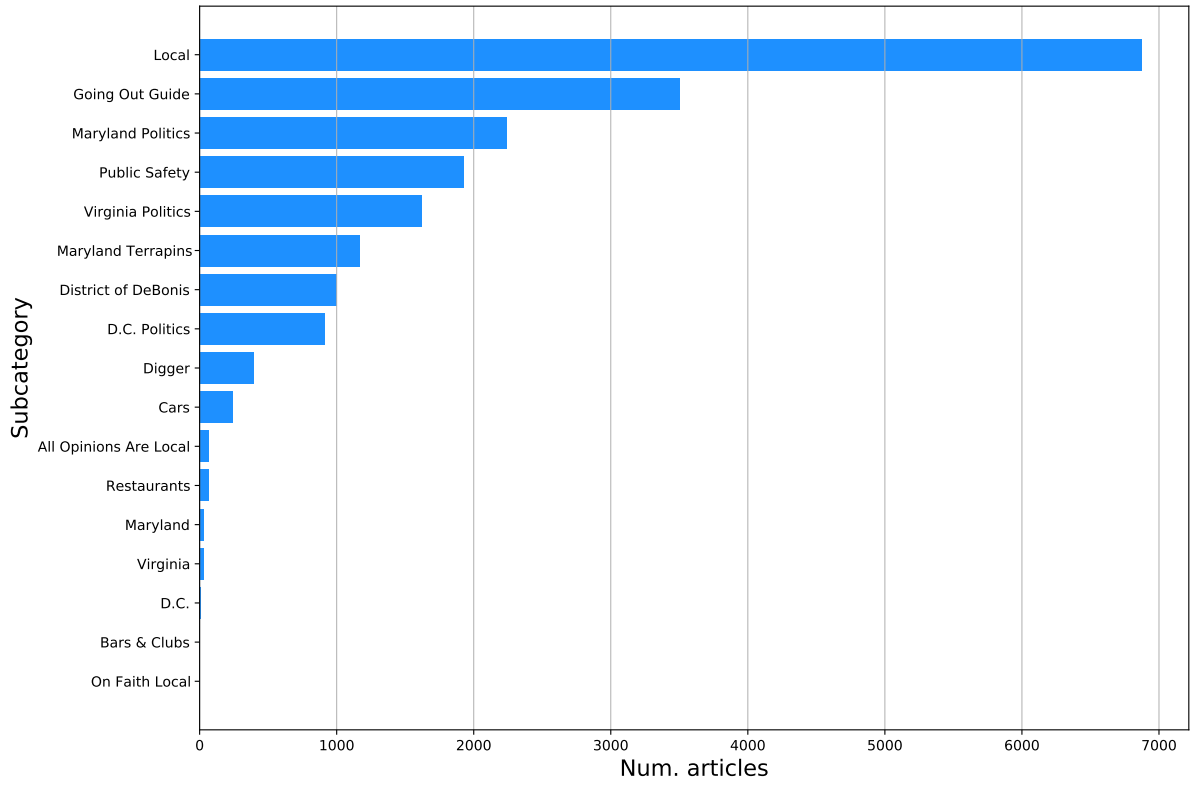

Figure 3.8: Number of articles for each subcategory in the "D.C., Md. & Va." category.

3.3.3 Statistics of Sample Used in the Study

As described in Section 3.3.2, "Politics" was the chosen category to leverage in the current study. To obtain an evenly distributed sample, 400 news articles were sampled from each year, resulting in 2400 news articles ($400 * 6 = 2400$). This was found necessary due to the uneven distribution as presented in Figure 3.9, which shows the number of political articles in each year. Additional general statistics of the sample dataset can be found in the Appendix on Table 1.

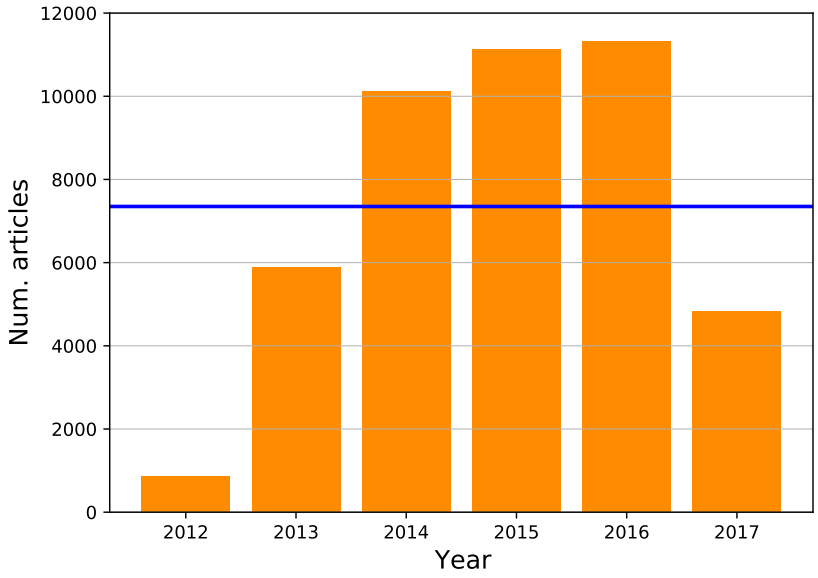

Figure 3.9: Number of political articles in each year. Blue line denotes the average. Note that the year 2017 ends at August.

- **Subcategory**

  Figure 3.10 presents the number of news articles for each subcategory in the sample dataset. From reviewing the subcategories on the Washington Post's website, most of these subcategories appear quite different from one another. For instance, the largest subcategory "The Fix" is a daily blog designed to be a 5-minute read on everything the reader needs to know about politics on the given day. The second-largest subcategory "2chambers", is about news and insights on political campaigns in the U.S. "Politics" are news articles that are not specified as belonging to a specific section, i.e. subcategory, of the Washington Post. During the category mapping process described in Section 3.1, these news articles were given "Politics" as both their category and subcategory.

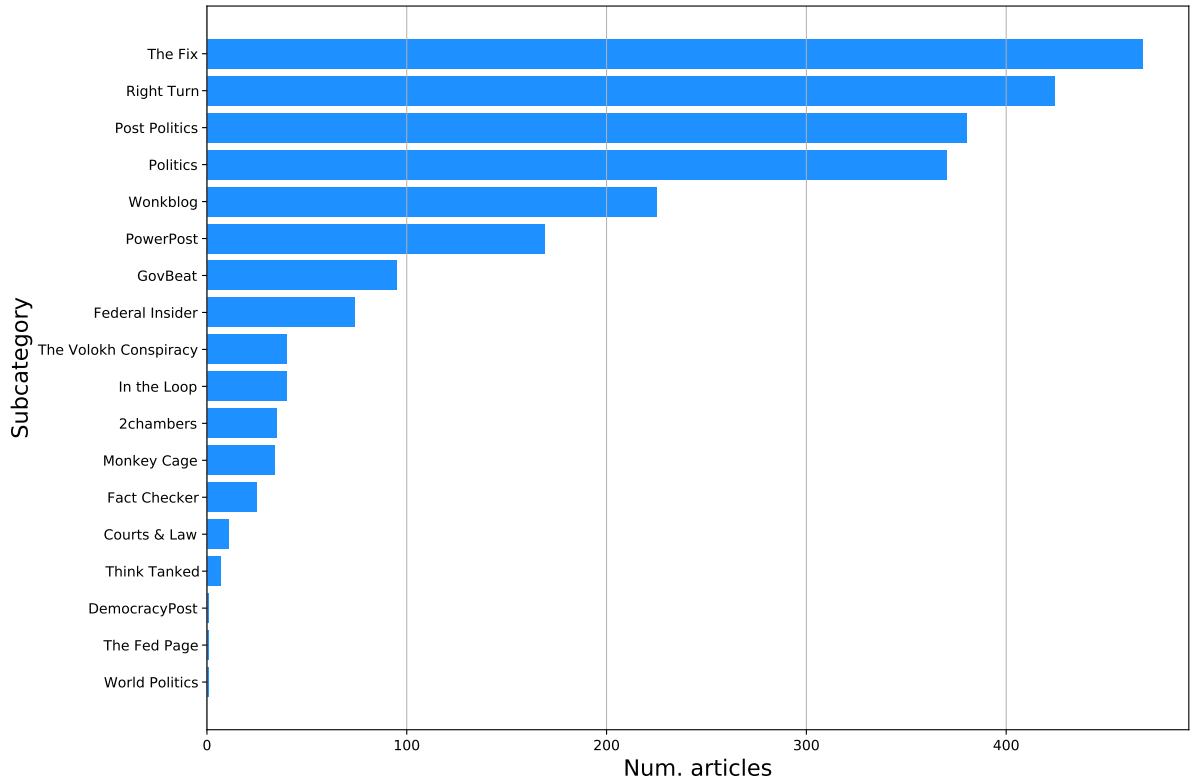

Figure 3.10: Number of articles for each subcategory in the sample dataset.

- **Title, body text, author biography, and author**

  Figure 3.11 presents the mean length of textual features title, body text, author biography, and the mean number of authors, for each year present in the sample dataset. Overall, the length of features and the number of authors involved in news articles increase for each year. This is interesting because the number of items published increases for each year in a similar fashion, as shown in Figure 3.9.

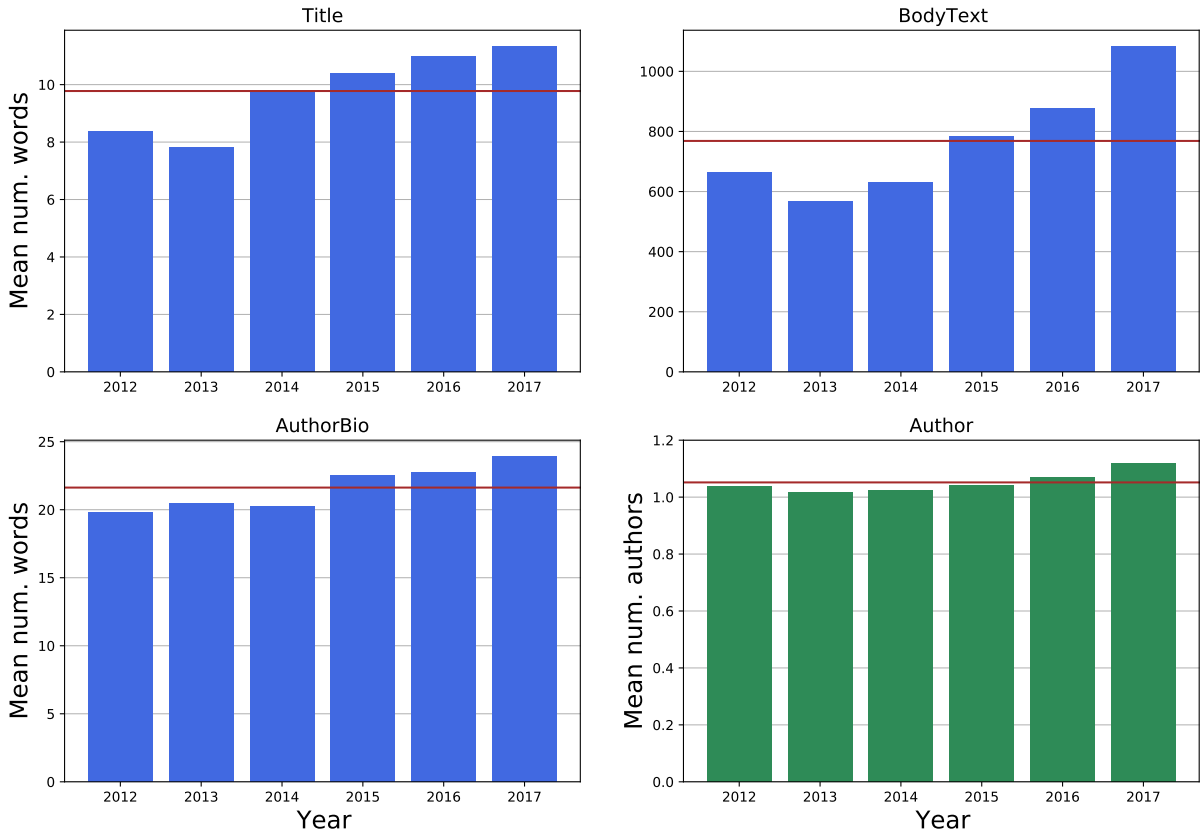

Figure 3.11: Average length of political news article titles, body texts, author biographies, and the mean number of authors for each year in the sample dataset. The red line denotes the mean across all years.

- **Date of publication**

  Figure 3.12 presents the distribution of articles between January 2012 and August 2017 in the sample dataset. We see that the distribution is similar to the distribution shown for all articles in Figure 3.5. Comparing the distribution of articles for all political articles against the sample dataset, the standard deviation is 0.17 for the percentage of articles sampled from each month for each year. This means that we have a fairly even distributed sample dataset, in terms of date of publication.

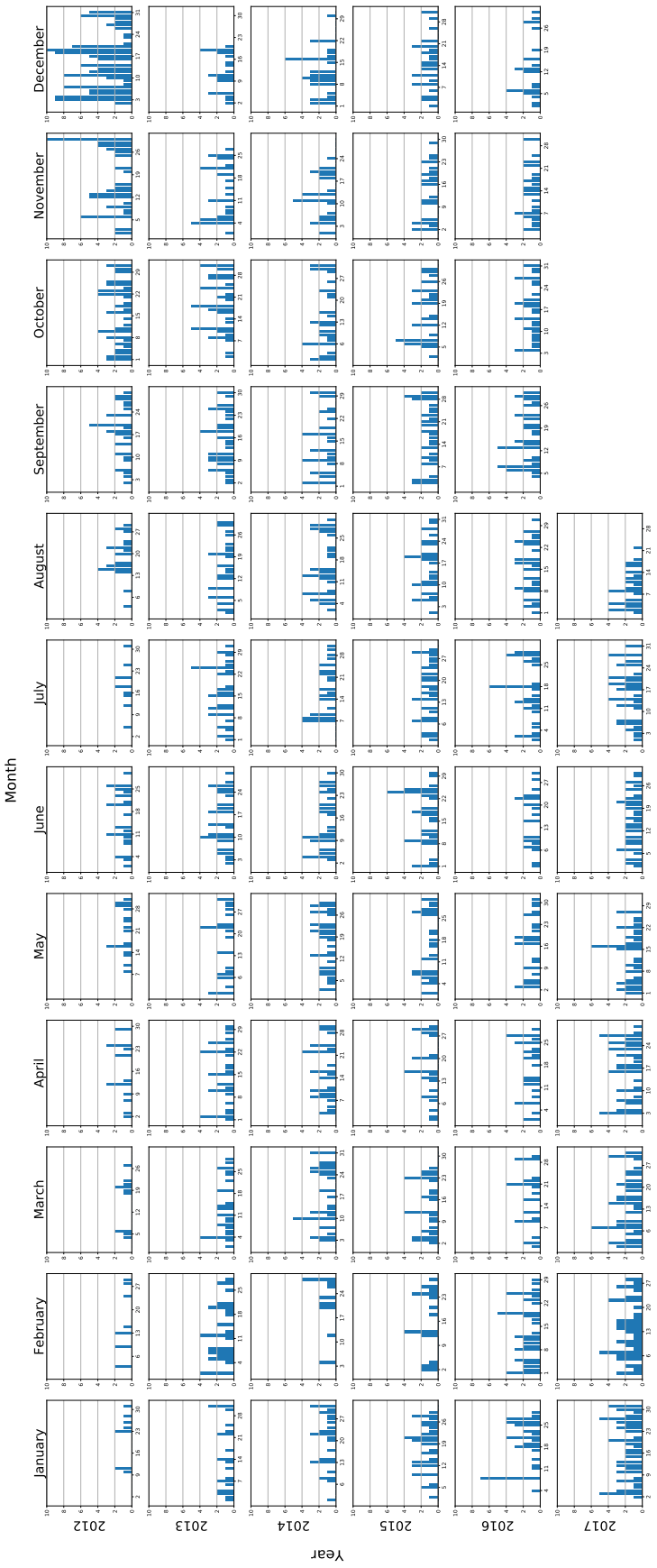

Figure 3.12: Date of publication distribution of news articles in the sample dataset, from January 2012 until August 2018. Each x-axis is marked by Mondays in the given month.

3.3.4 Summary of Exploratory Data Analysis

Generally, we see that the data evolves over the years. We see that the text of news articles become longer, and more authors are involved for every year that passes by. We also see that the majority of the news articles are published in the middle of a week. Additionally, we also see that the Washington Post is a very diverse newspaper, having many sections (here *subcategories*) wherein they publish news content. With the size of the newspaper, this naturally diversifies the dataset. As the second-largest, and arguably the least localized category, political news articles are used for this study. The sample drawn from the "Politics" category shows that the dataset is still quite diverse in terms of subcategories, but that the news articles are about national or international events. Additionally, the sample is evenly distributed over the years and months in terms of date of publication.

3.4 Learning the Similarity Function

As the previous work in Section 2 suggests, earlier news recommender scenarios most commonly use the title and the body text of news articles to determine similarity. in the current study, the approach is based on leveraging a variety of features to learn a similarity function that can consider multiple aspects in parallel. A set of 20 similarity functions, each based on one of the 7 features selected for this study were designed to learn this combined similarity function. The design of the combined similarity function is the product of finding an optimal combination of these single-aspect similarity functions, which achieves a minimal discrepancy between the user-provided similarity judgments and the predictions of the model [46].

This Section describes the process of learning the combined similarity function in the news domain. First, the 20 similarity functions are described in detail in Section 3.4.1. Then, the process of collecting the human judgments through a user study, as well as an overview of the resulting dataset, is described in Section 3.5.

3.4.1 Catalog of Similarity Functions

A goal of this study is to understand the relative strength of the similarity functions employed in the work of Trattner and Jannach [46] in a different domain. Thus, many of the similarity functions are carried over from their work within the recipe and movie domains. Naturally, the features available in the different domains are not strictly the same. However, many of the features share similar roles, i.e. movie domain's "director" and news domain's "author". Therefore, such features use the same similarity metrics.

The similarity functions shown in Table 3.3 are based on the six news article features subcategory, title, image, author, date, body text, and author biography. In previous work within recommender systems in the news domain, standard IR methods such as TF-IDF encodings on the body text of articles is used often. In this work, several ways of computing the similarity of two news articles are used that the author was unable to observe in previous work. For example, since this study uses the Washington Post Corpus, the presentation of articles to the study participants is based on the Washington Post's online format, which includes author biographies at the end of each news article. Here, the author biography is used in two different similarity functions.

Table 3.3: Similarity functions, each comprised of a feature and a metric. $*$ - Metrics also used in Trattner and Jannach [46].

Name	**Metric**	**Explanation**
Subcat:JACC	$sim(n_i, n_j) = 1 - \frac{Subcat(n_i) \bigcap Subcat(n_j)}{Subcat(n_i) \bigcup Subcat(n_j)}$	Subcategory Jaccard-based similarity
Title:LV*	$sim(n_i, n_j) = 1 - \lvert dist_{LV}(n_i, n_j) \rvert$	Title Levenshtein distance-based similarity
Title:JW*	$sim(n_i, n_j) = 1 - \lvert dist_{JW}(n_i, n_j) \rvert$	Title Jaro-Winkler distance-based similarity

Title:LCS*	$sim(n_i, n_j) = 1 - \lvert dist_{LCS}(n_i, n_j)\rvert$	Title longest common subsequence distance-based similarity
Title:BI*	$sim(n_i, n_j) = 1 - \lvert dist_{BI}(n_i, n_j)\rvert$	Title bi-gram distance-based similarity
Title:LDA*	$sim(n_i, n_j) = \frac{LDA(Title(n_i)) * LDA(Title(n_j))}{\lvert\lvert LDA(Title(n_i))\rvert\rvert \lvert\lvert LDA(Title(n_j))\rvert\rvert}$	Title LDA cosine-based similarity
Image:BR*	$sim(n_i, n_j) = 1 - \lvert BR(n_i) - BR(n_j)\rvert$	Image brightness distance-based similarity
Image:SH*	$sim(n_i, n_j) = 1 - \lvert SH(n_i) - SH(n_j)\rvert$	Image sharpness distance-based similarity
Image:CO*	$sim(n_i, n_j) = 1 - \lvert CO(n_i) - CO(n_j)\rvert$	Image contrast distance-based similarity
Image:COL*	$sim(n_i, n_j) = 1 - \lvert COL(n_i) - COL(n_j)\rvert$	Image colorfulness distance-based similarity
Image:EN*	$sim(n_i, n_j) = 1 - \lvert EN(n_i) - EN(n_j)\rvert$	Image entropy distance-based similarity
Image:EMB*	$sim(n_i, n_j) = \frac{EMB(n_i) * EMB(n_j)}{\lvert\lvert EMB(n_i)\rvert\rvert \lvert\lvert EMB(n_j)\rvert\rvert}$	Image embedding cosine-based similarity
Author:JACC	$sim(n_i, n_j) = 1 - \frac{Author(n_i) \bigcap Author(n_j)}{Author(n_i) \bigcup Author(n_j)}$	Author Jaccard-based similarity
Date:ND*	$sim(n_i, n_j) = 1 - \lvert dist_{days}(n_i, n_j)\rvert$	Date published distance-based similarity (unit = days)

BodyText:TF-IDF	$sim(n_i, n_j) = \frac{TF-IDF(Text(n_i)) * TF-IDF(Text(n_j))}{\|\|TF-IDF(Text(n_i))\|\|\|\|TF-IDF(Text(n_j))\|\|}$	All article body text cosine-based similarity
BodyText:50TF-IDF	$sim(n_i, n_j) = \frac{TF-IDF(Text(n_i)) * TF-IDF(Text(n_j))}{\|\|TF-IDF(Text(n_i))\|\|\|\|TF-IDF(Text(n_j))\|\|}$	First 50 words in article body text cosine-based similarity
BodyText:LDA	$sim(n_i, n_j) = \frac{LDA(Text(n_i)) * LDA(Text(n_j))}{\|\|LDA(Text(n_i))\|\|\|\|LDA(Text(n_j))\|\|}$	All article body text LDA cosine-based similarity
BodyText:Senti	$sim(n_i, n_j) = 1 - \|SENTI(n_i) - SENTI(n_j)\|$	Article body text sentiment distance-based similarity
AuthorBio:TF-IDF	$sim(n_i, n_j) = \frac{TF-IDF(Bio(n_i)) * TF-IDF(Bio(n_j))}{\|\|TF-IDF(Bio(n_i))\|\|\|\|TF-IDF(Bio(n_j))\|\|}$	Author bio cosine-based similarity
AuthorBio:LDA	$sim(n_i, n_j) = \frac{LDA(Bio(n_i)) * LDA(Bio(n_j))}{\|\|LDA(Bio(n_i))\|\|\|\|LDA(Bio(n_j))\|\|}$	Author bio LDA cosine-based similarity

Title-based similarity consists of four string similarity metrics and one topic similarity metric. The string metrics use the Levenshtein distance metric (LV) [49], the Jaro-Winkler method (JW) [23], and lastly the bi-gram distance method (BI) [26]. The similarity is determined by calculating the distance (*dist*) for two news articles n_i and n_j as follows:

$$sim(n_i, n_j) = 1 - |dist(n_i, n_j)| \text{ [46]} \tag{3.1}$$

The last metric uses Latent Dirichlet Allocation (LDA) for topic-modeling [5]. Following Trattner and Jannach's parameters, the number of topics was set to 100 [46]. Two news articles can thus be compared as follows, given two weight vectors $LDA(n_i)$ and $LDA(n_j)$, and Cosine similarity:

$$sim(n_i, n_j) = cos(LDA(Title(n_i)), LDA(Title(n_j))) \text{ [46]} \tag{3.2}$$

Image-based similarity consists of six similarity metrics. Five of them are low-level metrics based on brightness, sharpness, contrast, colorfulness, and shannon entropy [38, 46].

The last of the six similarity metrics is more complex and is based on convolutional neural networks (CNNs) and image embeddings [46]. In earlier recommender scenarios, both of these feature spaces—low-level image features and CNN features—have been useful in different recommendation scenarios [46]. The images are resized to a maximum size of $500x500$ pixels for each similarity function.

- Image: *brightness* (BR)

  The brightness of an image is the subjective visual perception of the energy output of a light source [38]. The average brightness can be computed by using default parameters and the NTSC weighting scheme as follows:

$$avg\_brightness = \frac{1}{N}\sum_{x,y} Y_{xy}\text{ , with} \tag{3.3}$$

$$Y_{xy} = (0.299 * R_{xy} + 0.587 * G_{xy} + 0.114 * B_{xy})\text{[46].}$$

  In the luminance algorithm, Y_{xy} denotes the luminance value, and N the size of the image. R, G, and B correspond to the RGB color space channels of pixels x, y [46].

- Image: *sharpness* (SH)

  The sharpness of an image can be computed by using the Laplacian L of an image, then divided by the locale average luminance (μ_{xy}) around pixel (x, y):

$$avg\_sharpness = \sum_{x,y} \frac{L(x,y)}{\mu_{xy}}\text{, with} \tag{3.4}$$

$$L(x,y) = \frac{\partial^2 I_{xy}}{\partial x^2} + \frac{\partial^2 I_{xy}}{\partial y^2}\text{[46],}$$

  where I_{xy} denotes the intensity of a pixel [46].

- Image: *contrast* (CO)

  The intensity of each pixel in an image can be used to compute the relative difference luminance, i.e. the contrast. The root-mean-square contrast (RMS contrast) approach is defined as follows:

$$avg\_contrast = \frac{1}{N}\sum_{x,y}(I_{xy} - \overline{I})\text{[46].} \tag{3.5}$$

  I_{xy} denotes the intensity of a pixel, $\overline{I}$ the arithmetic mean of the pixel intensity, and N the number of pixels [46].

- Image: *colorfulness* (COL)

  The colorfulness of an image can be computed by using the individual color distance of the pixels in an image [38]. To do this, the image needs to be transferred to an sRGB color space, using:

$$rg_{xy} = R_{xy} - G_{xy}\,, \tag{3.6}$$

$$yb_{xy} = 1/2(R_{xy} + G_{xy}) - B_{xy}\,, \tag{3.7}$$

  where R_{xy}, G_{xy}, and B_{xy} are the color channels of the pixels. The colorfulness can then be measured as follows:

$$\begin{aligned} &avg\_colorfulness = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb}, \text{with} \\ &\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2}\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2}\,[46], \end{aligned} \tag{3.8}$$

  where σ, μ, and 0.3 are the standard deviation, the arithmetic mean, and a pre-defined parameter in OpenIMAJ, respectively [46].

- Image: *entropy* (EN)

  The entropy of an image can be described as the amount of information observed. In this work, the Shannon entropy is used to compare two images. First, the images are converted to gray-scale, resulting in each pixel containing exactly one intensity value. Second, the occurrence of each distinct value is counted. The entropy can then be computed as follows:

$$avg\_entropy = -\sum_{x \in [0..255]} p_x \cdot log_2(p_x)[46]. \tag{3.9}$$

  Here, p_x denotes the probability of finding the gray-scale value x among all pixels in the image [46].

- Image: *embedding* (EMB)

  The image embeddings were computed by using a pre-trained (ImageNet) VGG-16 network, which has been used in a number of recent recommendation scenarios, such as Messina et al. [35], Eksombatchai et al. [41], and Trattner and Jannach [46]. As in Trattner and Jannach [46], the first fully connected layer of the network is used as output. The output for each news image n is thus a vector $EMB(n)$ with 4096 elements. Cosine similarity can then be used to compare two news image embeddings. Using the Keras[4]

[4] https://keras.io/

framework for the computations, the images were all automatically downsampled to fit the input layer [46].

Author and subcategory-based similarity each consists of a single keyword metric, the Jaccard coefficient. They are both based on Trattner and Jannach's use of the Jaccard coefficient on the movie domain's features *director* and *genre* [46], which serve similar purposes as features in their respective domains as author and subcategory do in the news domain.

Date-based similarity consists of a linear function which computes the similarity based on how many days apart two articles were published. It can be expressed as follows:

$$sim(n_i, n_j) = 1 - |dist_{days}(n_i, n_j)| \text{ [46]}, \tag{3.10}$$

where $dist_{days}$ denotes the number of days between two date of publications.

Body text-based similarity consists of two string similarity metrics, a topic similarity metric, and a sentiment-based similarity metric. In Trattner and Jannach [46], LDA and TF-IDF were used with the feature *plot* in the movie domain, *directions* in the recipe domain. Both of these features can be regarded as the largest textual features among the movie features and is therefore comparable to news domain's feature *body text*. The assumption that LDA and TF-IDF are suitable metrics for the body text feature is further emphasized by the frequent usage observed in previous work (see Table 2.1).

The string metrics are both based on TF-IDF encodings, paired with Cosine similarity. In one, the body text of all articles are reduced to the first 50 words. This is because, in Trattner and Jannach [46], the comparable movie feature *plot* has an average of 51 words, and the recipe feature *directions* an average of 111 words. Thus, the first 50 words of a news article body text is more comparable to these features. In the other metric, the full body texts are used, which is the more common approach. To perform topic-modeling using LDA, the same approach as earlier described for the title was used. The last similarity function leverages sentiment to derive similarity, which has been suggested to be important in understanding how readers digest political news [44].

Author biography-based similarity consists of a string metric (TF-IDF) and a topic metric (LDA). Using the full length of the author biographies, the similarity was computed using the same approach as earlier described for the body text.

3.5 Collecting Human judgments

This Section describes the process that was undertaken to collect human similarity judgments on the crowdsourcing platform Amazon Mechanical Turk. The user study participants were presented with pairs of news articles, 10 pairs in total. For each pair, the user was to judge their similarity on a Likert scale of 1-5. Afterward, they answered questions regarding their background, characteristics, and approach to judging the similarity of the news articles.

3.5.1 Sampling Pairs for Human judgment

To ensure diversity in the dataset, 400 news articles were sampled from each year available, resulting in 2400 news articles ($400 * 6 = 2400$). In the next step, all pairwise similarity values were computed using the 20 similarity functions presented in Table 3.3. The overall similarity value for each pair was then calculated by using a linear combination of all 20 similarity functions using equal weights. In the last step, a biased stratified sampling strategy was employed to ensure similarity diversity [47, 46]. In this step, 2000 news article pairs were sampled between quantile Q0-Q1, 2000 between quantile Q2-Q8, and 2000 between quantile Q9-Q10. The resulting 6000 pairs could then be used for the human judgments user study.

3.5.2 Data Collection

To conduct the user study, participants from Amazon Mechanical Turk[5] were directed to a Web application designed for the study. Each user was then tasked to assess the similarity of 10 pairs of news articles, on a 5-point Likert scale. After reviewing the 10 pairs, the users were asked to consider the degree to which they used the various information cues [6] available in the news articles. Finally, the participants were asked questions regarding their age, gender, how often they read news, and how often they use online news websites. All questions can be found in the Appendix on Table 2.

To ensure that the responses on the survey were reliable, two measures were taken [20, 8]. First, only crowd-workers with a minimum of 98% HIT-rate, and who had positive evaluations in 500 HITs in the past were allowed to participate. Second, the Web application included

[5] https://www.mturk.com/

[6] Title, image, author, date of publication, body text, author biography

an attention check. The attention check appeared randomly during any of the 10 pairs a participant is shown. In the event of an attention check, the body texts of the news articles are replaced with text stating that the user must answer with a 5 on each Likert scale on the current page. Since the Washington Post is a U.S-based newspaper, the survey was restricted to U.S-residents only. The estimated time to completion for each user was 5-10 minutes, and the reimbursement was therefore set to 0.5 USD per HIT.

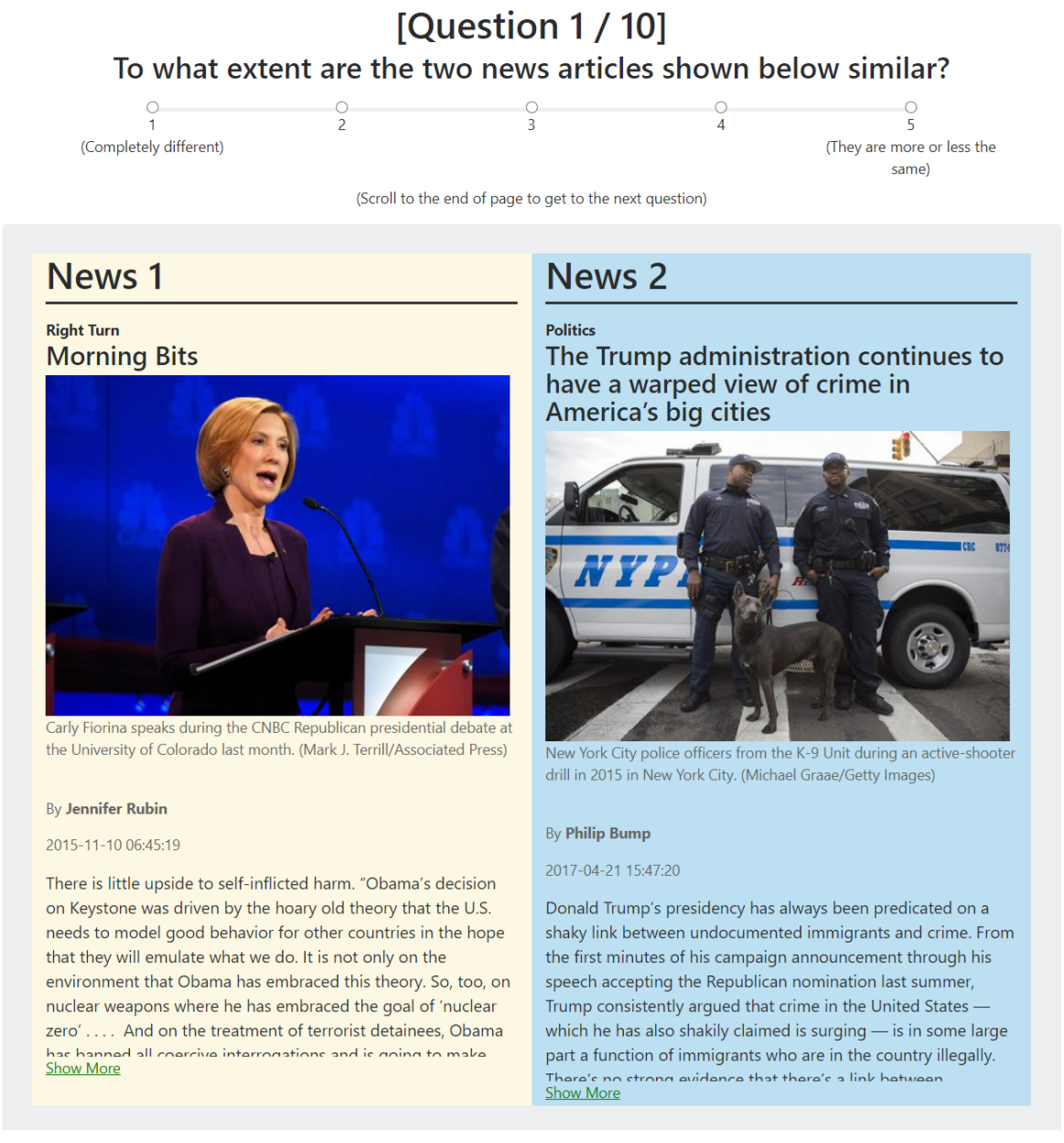

Figure 3.13: Web application for conducting user study on Amazon Mechanical Turk. Scale: 1(Completely different)-5(They are more or less the same).

3.5.3 Participants

The user study was set to recruit 400 crowd-workers. This resulted in 401 successfully completed surveys, and thus 3,609 evaluated news article pairs after removing each attention check, which accounted for one in ten. The users completed the survey at a median of 6 minutes and 35 seconds, which was slightly lower than anticipated. Surprisingly, only 241 (60%) of the users passed the attention check. This is surprising since (i) the survey was restricted to experienced crowd-workers, and (ii) the attention check appeared in the body text of the news articles, meaning it should be easily spotted. Filtering out the surveys where the attention check was not passed, we are left with 2,169 human similarity judgments.

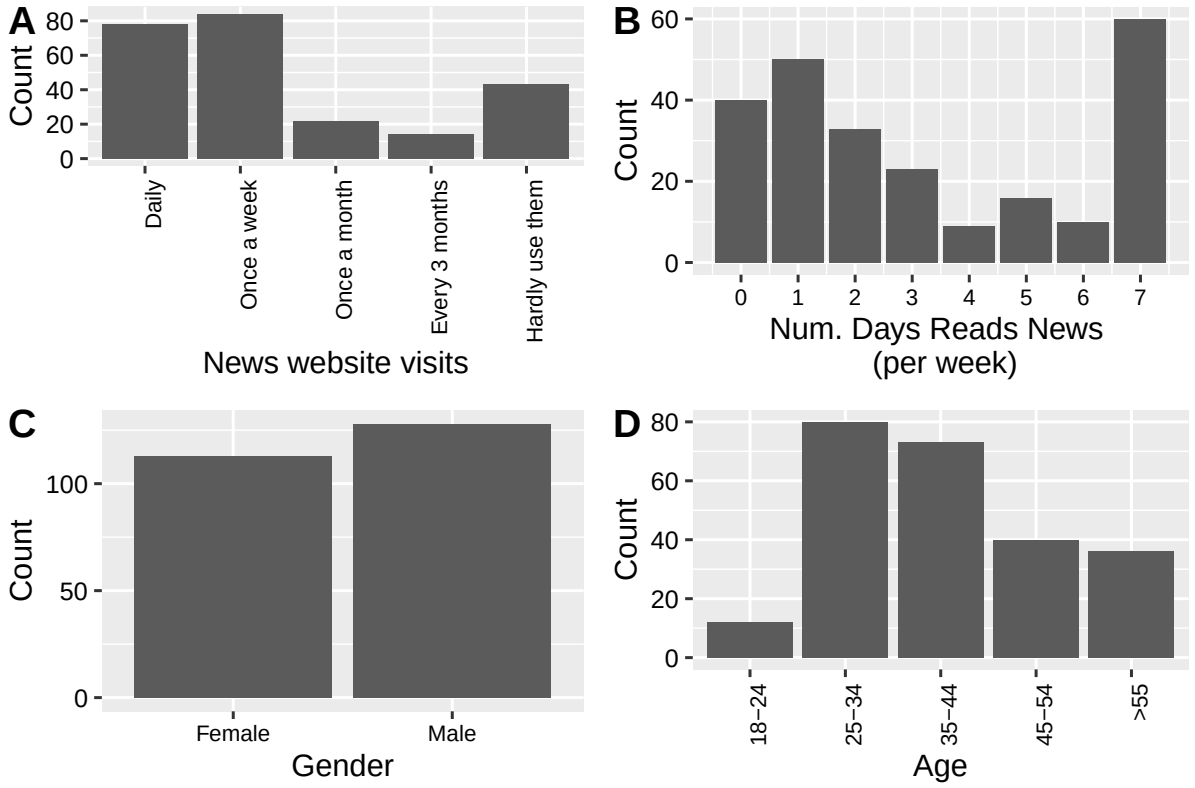

Figure 3.14: Characteristics of the user study participants who passed the attention check.

Looking at Figures A and B in Figure 3.14, we can see that the users' self-assessment of their news reading habits are quite diverse. About half of the users report that they use news websites as a news-source daily or once a week, while the other half report that they rarely use them. However, only 40 users report that they do not read news at least once a week.

We can also see that the user demographic is quite diverse, with an even distribution of males and females, and ages ranging from 25 and upwards of 55.

Chapter 4

Results

4.1 Information Cue Usage

We first review what the participants stated regarding their use of information cues when assessing the similarity of two news articles. Figure 4.1 shows that the body text and title are the most important information cues according to the assessments of the participants. These are followed by subcategory, image, date of publication, author, and author biography, respectively. Performing a one-way ANOVA[1] and a Tukey's HSD post hoc test reveals that all differences are statistically significant ($p < 0.01$)(see Figure 4.1b).

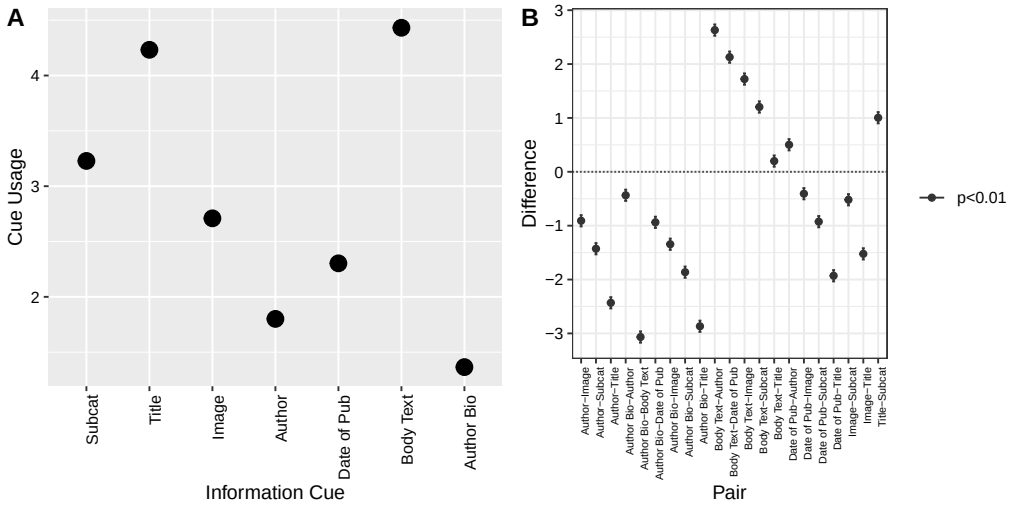

Figure 4.1: **a**: Information cue usage (means and std. errors), and **b**: pairwise comparison. Scale: 1(disagree)-5(agree)

[1] Levene's test was used to check the homogeneity of variances for all ANOVA tests.

4.2 Correlation Analysis

To address RQ1 on the relative importance of the different features, the results were analyzed to understand the correlations between the similarity judgments provided by the user study participants, against the 20 designed similarity functions. Table 4.1 shows the Spearman correlation coefficient[2] for users who passed the attention check (ρ_{pass}), and all users (ρ_{all}).

The table shows that the correlations are generally higher with the users who passed the attention check, albeit not by much. The highest correlation observed is the *BodyText:TF-IDF* metric ($\rho = 0.29$, $p < 0.001$). This is not surprising, given its common usage in earlier news recommendation scenarios [25]. *BodyText:50TF-IDF* has a much lower correlation ($\rho = 0.14$, $p < 0.001$), which is surprising since users might only have inspected an article's first 50 words (see visible text in Figure 3.13; on average 15% of full body text). The lowest correlation is found on *BodyText:Senti* ($\rho = -0.02$), which is somewhat surprising since, as previously observed, the participants reported it to be of high importance. Both LDA-based metrics are shown to have very low, insignificant correlations (*Title:LDA*, $\rho = 0.02$, *BodyText:LDA*, $\rho = 0.03$). A possible explanation is that they might have suffered from poorly optimized parameters due to insufficient latent topic information. Image embeddings (*Image:EMB*) is the strongest image-based metric, however with a modest correlation ($\rho = 0.17, p < 0.001$). The body text metric *BodyText:TF-IDF*, author metrics (*Author:Jacc*, *AuthorBio:TF-IDF*), and an article's subcategory (*Subcat:Jacc*), seem to best represent user similarity judgments.

[2]Spearman was chosen as the correlation metric since the data, i.e. user ratings, are (i) not normally distributed, and (ii) are on an ordinal scale.

Table 4.1: ρ_{pass} are correlations with users who passed the attention check. ρ_{all} denotes all users. Note: $^{*}p < 0.05$;$^{**}p < 0.01$;$^{***}p < 0.001$

Metric	ρ_{pass}	ρ_{all}
Subcat:Jacc	0.14^{***}	0.11
Title:LV	0.06^{**}	0.04^{*}
Title:JW	0.05^{*}	0.03
Title:LCS	0.08^{***}	0.05^{**}
Title:BI	0.08^{***}	0.07^{***}
Title:LDA	0.02	0.00
Image:BR	0.10^{***}	0.07^{***}
Image:SH	0.06^{**}	0.03
Image:CO	0.05^{*}	0.05^{**}
Image:COL	0.05^{*}	0.03^{*}
Image:EN	0.07^{**}	0.05^{**}
Image:EMB	0.17^{***}	0.13^{***}
Author:Jacc	0.14^{***}	0.10^{***}
Date:ND	0.09^{***}	0.08^{***}
BodyText:TF-IDF	0.29^{***}	0.23^{***}
BodyText:50TF-IDF	0.14^{***}	0.12^{***}
BodyText:LDA	0.03	0.01
BodyText:Senti	-0.02	-0.02
AuthorBio:TF-IDF	0.15^{***}	0.12^{***}
AuthorBio:LDA	0.11^{***}	0.09^{***}

The similarity functions for each feature were linearly combined to better understand the correlations between human judgments and each *type of feature*. Table 4.2 shows the human judgment correlations (first row) against each type of feature. Additionally, the correlation when all metrics are combined together is shown (*All*). Despite *BodyText:TF-IDF* being found to have the strongest, single metric correlation, the overall strongest correlation is found

when all metrics are combined together (*All*, $\rho = 0.18$, $p < 0.001$), followed by *Subcat* ($\rho = 0.14$, $p < 0.001$) and *AuthorBio* ($\rho = 0.14$, $p < 0.001$). *BodyText*'s overall low correlation is found be to due to the *BodyText:Senti* metric, which was found to have a negative correlation and halved *BodyText*'s overall correlation.

Comparing the information cue usage shown in Figure 4.1 and the correlations shown in Table 4.2 reveal several interesting findings. Title (*Title*, $\rho = 0.07$, $p < 0.001$) was reported to be of high importance by the participants, but is found to have the lowest correlations to the ratings provided by them. Similarly, the participants reported the body text to be the most important information cue, but is found to have one of the lowest correlations ($\rho = 0.13$, $p < 0.001$). On the other hand, the participants reported the author-related features (*Author* $\rho = 0.13$, $p < 0.001$, *AuthorBio*, $\rho = 0.14$, $p < 0.001$) to be of low importance, but the correlations are on a similar level to the subcategory (*Subcat*, $\rho = 0.14$, $p < 0.001$) and image (*Image*, $\rho = 0.14$, $p < 0.001$), which were reported to be of some importance.

Overall, the correlations of the similarity metrics are quite low, which indicates that the developed functions may not be optimal for the task of predicting user-perceived similarity levels. This is emphasized by the contradictions found between the information cues and the various similarity metrics, as well as between the similarity metrics and the ratings provided by the participants. However, the correlations between the different features are generally low, which indicates that there is not a high data multicollinearity present.

Table 4.2: Similarity metric correlation (Spearman) with user similarity estimates per type of feature. The metrics are linearly combined using equals weights in the linear model.
Note: $^{*}p < 0.05$;$^{**}p < 0.01$;$^{***}p < 0.001$.

	Human	Subcat	Title	Image	Author	Date	BodyText	AuthorBio	All
Human	1	0.14***	0.07***	0.14***	0.13***	0.09***	0.13***	0.14***	0.18***
Subcat	—	1	0.13***	0.37***	0.55***	0.28***	0.2***	0.53***	0.74***
Title	—	—	1	0.14***	-0.07**	0.12***	0.11***	0.02	0.30***
Image	—	—	—	1	0.18***	0.22***	0.18***	0.29***	0.70***
Author	—	—	—	—	1	0.16***	0.12***	0.63***	0.63***
Date	—	—	—	—	—	1	0.15***	0.15***	0.45***
BodyText	—	—	—	—	—	—	1	0.16***	0.32***
AuthorBio	—	—	—	—	—	—	—	1	0.61***
All	—	—	—	—	—	—	—	—	1

4.3 Learning The Similarity Function

Machine learning is applied to answer RQ2 on which combination of features is best suited for predicting user-perceived similarity levels. When a high level of disparity is observed between correlations of features, as in this case, machine learning further helps in understanding the importance of each feature for prediction [46]. Thus, the goal is to learn a model that leads to the lowest possible prediction error, with a minimal discrepancy between the predicted similarity and the human similarity judgments for the same news article pairs.

Following the blueprint of Trattner and Jannach [46], five different regression models were applied using the same approach. The models include, among others, linear regression (LM), Ridge regression (Ridge), and Lasso regression (Lasso). Though it is unlikely to be a problem here, the latter two are often considered the better to handle multicollinearities [46]. The last two models are Random Forest (RF) and Gradient Boosting (GB). The overall mean of all similarity judgments and a random predictor were used as baselines. A regression model using feature-based similarity functions can then be expressed as follows:

$$sim_H(n_i, n_j) = REG(sim_{fk}(n_i, n_j), ..., sim_{fk}(n_i, n_j)) \text{ [46]}. \tag{4.1}$$

Here, n_i and n_j denotes news article pairs from all news article pairs N. The unique human similarity judgment for a news article pair to be predicted is denoted as $sim_H(n_i, n_j)$. Lastly, REG represents an arbitrary regression method using similarity functions based on features $sim_{fk}(n_i, n_j)$ [46], as previously presented in Table 3.3. Extending upon this, a linear regression model (LM) can be expressed, where REG becomes:

$$REG = \sum_{f \in F} \beta_f * sim_f(n_i, n_j)[46], \tag{4.2}$$

where feature-based similarity functions are denoted as $sim_f(n_i, n_j)$, F represents a set of these, and β_f are the weights to be learned in the model [46].

All models were evaluated using root-mean-square-error (RMSE), R squared (R^2), mean absolute error (MAE), and Spearman correlation (p). Five-fold cross-validation was used to get an average of each performance measure. Furthermore, by applying grid search on a validation set from the training data, the optimal hyper-parameters for each model were found [46].

Table 4.3 shows that Lasso is the best performing model. The difference to the random baseline is, according to a Wilcoxon Rank-Sum test on RMSE, statistically significant ($p < 0.05$) for all models except Gradient Boosting (GB). In Trattner and Jannach [46], Ridge was found to be the best model, thus making it our point of focus going forward.

The importance of the different features used for prediction in the Ridge model, i.e. the normalized ranks of the model coefficients [46], are shown in Figure 4.2. These are determined using the "varImp" method of R's *caret* package. The results are mostly in line with the observations from Table 4.1, with *BodyText:TF-IDF* and *Image:EMB* being among the most important metrics. Interesting to note is that the title metrics *Title:BI* and *Title:LV* appear among the most important, despite having lower correlations than other metrics such as *Subcat:Jacc*, which does not appear.

Table 4.3: Performance of different learning approaches.

Method	RMSE	R^2	MAE	ρ
(Instances = 2,169)				
Model performance (All features)				
All (RF)	0.9219	0.2982	0.7643	0.2411
All (GB)	0.9177	0.3123	0.7520	0.2005
All (Ridge)	0.9141	0.3257	0.7459	0.2922
All (LM)	0.9120	0.3289	0.7453	0.2896
All (Lasso)	0.9101	0.3339	0.7480	0.2720
Baselines				
Mean	0.9652	0.0000	0.8122	-0.0010
Random	0.9659	-0.0226	0.8125	-0.0300

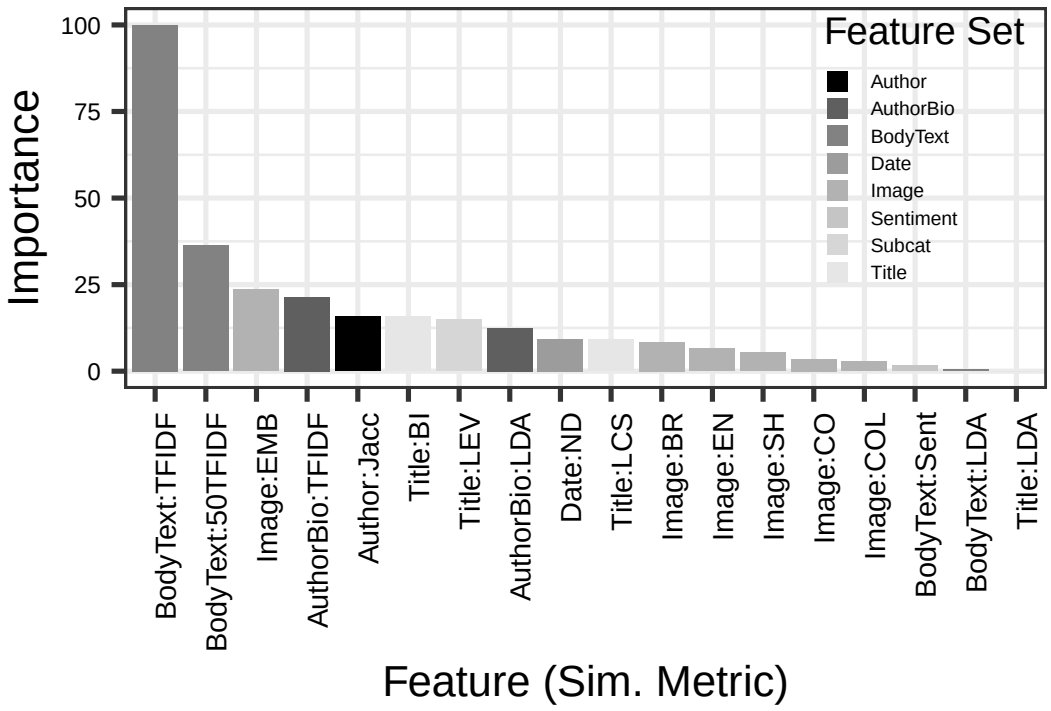

Figure 4.2: Feature importance for the Ridge regression model.

4.3.1 Considering Additional Features for Prediction

If one knows user characteristics or demographics, it can be worthwhile to incorporate these into the predictive models. Similarly to Trattner and Jannach's [46] assumption that those with more experience cooking food look more closely at the directions, we can assume that those who read news more frequently are better at assessing news similarities. Moreover, a user's age or gender can play a part in how they assess similarity.

Table 4.4: Performance of Ridge regression using additional features.

Method	RMSE	R^2	MAE	ρ
(Instances = 2,169)				
Model performance (All features)				
All (Ridge)	0.9141	0.3257	0.7459	0.2922
All (Ridge) + additional features (user characteristics)				
News website visits	0.9164	0.3207	0.7463	0.2819
Num. days reads news	0.9186	0.3215	0.7476	0.2812
Gender	0.9125	0.3314	0.7456	0.2896
Age	**0.9081**	**0.3435**	**0.7338**	**0.3130**
All additional features	0.9099	0.3412	0.7358	0.3049

Extending upon the original approach to predicting similar news articles, the user characteristics and demographics collected from the participants are incorporated to see if they affect the performance of the Ridge regression model. In this approach, the additional features are combined with the original Ridge regression model one by one, and finally including all at once. The results are presented in Table 4.4, where we see that the additional features have minimal impact on users' perception of similarity. The *age* model exhibits the best RMSE score, but it is nonetheless found to not be of statistically significant difference against the original Ridge model, according to a Wilcoxon Rank-Sum test on RMSE.

4.3.2 Considering Single Features for Prediction

Finally, Table 4.5 shows the results when a regression model is constructed for each of the 7 individual features. For features with more than one similarity function, Ridge regression is the chosen model. For features with only a single feature, Linear regression (LM) is used. Despite the *BodyText* metrics not being found to overall correlate the best, we see that it is clearly the best feature in predictive models. Furthermore, a Wilcoxon Rank-Sum test on RMSE reveal that there is not a statistically significant difference between the *BodyText* model and the *All features* model. Interestingly, *AuthorBio* appears to be a slightly better predictor than the author. We also see that *Title* is not a good predictor, suggesting that the feature is not representative of an article's information. Overall, the results indicate the body text, image, and author biography to be the best features for prediction.

Table 4.5: Ridge regression using only one information cue (feature) at a time.

Method	RMSE	R^2	MAE	ρ
(Instances = 2,169)				
All features (Ridge)	0.9141	0.3257	0.7459	0.2922
Regression model per information cue				
Subcat (LM)	0.9554	0.1406	0.7943	0.1106
Title (Ridge)	0.9618	0.0889	0.8071	0.0759
Image (Ridge)	0.9548	0.1495	0.7913	0.1590
Author (LM)	0.9568	0.1333	0.7991	0.0724
Date (LM)	0.9616	0.0911	0.8070	0.0813
BodyText (Ridge)	**0.9141**	**0.3244**	**0.7514**	**0.2847**
AuthorBio (Ridge)	0.9561	0.1414	0.7991	0.1268

4.4 Comparing the News, Recipe, and Movie Domains

Finally, RQ3 is addressed on how the news domain compares to the recipe and movie domains. We first review and discuss the differences in the reported information cue usages. Then, the metric correlations between similarity functions and human judgments are reviewed, for

the news domain as previously presented in Table 4.1, and the recipe and movie domains in Trattner and Jannach [46]. Finally, we review and discuss the differences in the performance of the Ridge model previously presented in Tables 4.3, 4.4, and 4.5 against the Ridge models from the same approaches in Trattner and Jannach [46].

4.4.1 Differences in Information Cue (Feature) Usages

Figure 4.3 presents the reported usage of the different types of information cues across the domains. The larger textual features body text in news and plot in movies are found to be of similar importance. However, the same type of feature, i.e. directions, is reported to be of the lowest importance of all in recipes. Moreover, the image is reported to be of similar, modest importance in movies and news, but very high importance in recipes. In the news and movie domains, the keyword features subcategory and genre are reported to be of modest to high importance. From this, it may seem like features that describe an item on a general level have some importance to the users. Extending upon this, it could be interesting to see the importance of a similar feature in the recipe domain, i.e. a feature that describes the origin of a recipe.

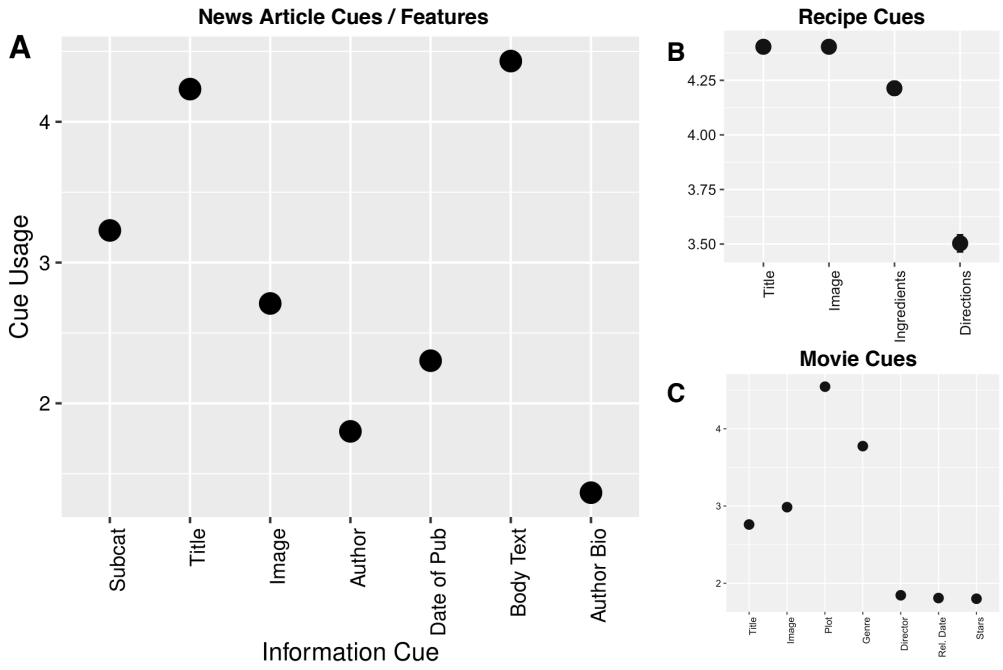

Figure 4.3: Reported information cue/feature usage (1 - did not use it; 5 - always used it) in this news study, compared to the reported usage for recipes and movies in Trattner and Jannach [46]. Graphs were adapted with permission.

4.4.2 Differences in Metric Correlations

Some types of features certainly share similarities in terms of importance across the different domains. We can see that the image of a news article or the poster (image) of a movie does not bear the same importance as the image of a recipe, which is reflected by both the correlations and the reported information cue usages. It is also evident that larger textual features, i.e. news body text, movie plot, and recipe directions, have high importance, even if users report that they do not use these information cues (as with directions in the recipe domain, where it had the lowest importance of all). Furthermore, the creator of the item (news:*author*, movies:*director*) seems to bear much importance in neither the news nor movie domain, which is shown by the low correlations as well as the low reported information cue usages (see Figure 4.3). Interesting to note is that neither the title of a news nor movie item seem to contain enough information for LDA. This is especially interesting regarding news since it suggests that a title does not describe the information of a news article very well.

Overall, the findings reveal that some of the measures proposed by Trattner and Jannach [46] for the recipe and movie domains do not translate well into the news domain. However, it also becomes all the more clear that the differences in how users perceive item similarity is quite disparate, i.e. how the image is vastly more important in the recipe domain than in news, despite users reporting otherwise. It also further emphasizes the need for multifaceted studies such as this, where it becomes apparent that users' own assessments on how they assess similarity can be misleading, which was also noted by Trattner and Jannach [46].

Table 4.6: Correlations of similarity metrics in the news, recipe, and movie domains. Data from the recipe and movie domains were obtained from Trattner and Jannach [46].
Note: ρ_{pass} are correlations with users who passed the attention check. ρ_{all} denotes all users. $^{*}p < 0.05; ^{**}p < 0.01; ^{***}p < 0.001$.

News Articles			**Recipes[46]**			**Movies[46]**		
Metric	ρ_{pass}	ρ_{all}	Metric	ρ_{pass}	ρ_{all}	Metric	ρ_{pass}	ρ_{all}
Subcat:Jacc	0.14^{***}	0.11	N/A	N/A	N/A	Genre:Jacc	0.56^{***}	0.53^{***}
Title:LV	0.06^{**}	0.04^{*}	Title:LV	0.48^{***}	0.38^{***}	Title:LV	0.19^{***}	0.18^{***}
Title:JW	0.05^{*}	0.03	Title:JW	0.46^{***}	0.35^{***}	Title:JW	0.16^{***}	0.16^{***}
Title:LCS	0.08^{***}	0.05^{**}	Title:LCS	0.50^{***}	0.40^{***}	Title:LCS	0.20^{***}	0.19^{***}
Title:BI	0.08^{***}	0.07^{***}	Title:BI	0.48^{***}	0.38^{***}	Title:BI	0.17^{***}	0.17^{***}
Title:LDA	0.02	0.00	Title:LDA	0.22^{***}	0.19^{***}	Title:LDA	0.01	0.01
Image:BR	0.10^{***}	0.07^{***}	Image:BR	0.18^{**}	0.14^{*}	Image:BR	0.22^{***}	0.20^{***}
Image:SH	0.06^{**}	0.03	Image:SH	0.16^{*}	0.11^{*}	Image:SH	0.10^{***}	0.08^{***}
Image:CO	0.05^{*}	0.05^{**}	Image:CO	0.29^{***}	0.20^{***}	Image:CO	0.03	0.03
Image:COL	0.05^{*}	0.03^{*}	Image:COL	0.09^{*}	0.07^{*}	Image:COL	0.15^{***}	0.14^{***}
Image:EN	0.07^{**}	0.05^{**}	Image:EN	0.34^{***}	0.28^{***}	Image:EN	0.15^{***}	0.09^{***}
Image:EMB	0.17^{***}	0.13^{***}	Image:EMB	0.44^{***}	0.34^{***}	Image:EMB	0.18^{***}	0.16^{***}
Author:Jacc	0.14^{***}	0.10^{***}	N/A	N/A	N/A	Dir:Jacc	0.10^{***}	0.07^{***}
Date:ND	0.09^{***}	0.08^{***}	N/A	N/A	N/A	Date:MD	0.37^{***}	0.35^{***}
BodyText:TF-IDF	0.29^{***}	0.23^{***}	Dir:TF-IDF	0.50^{***}	0.40^{***}	Plot:TF-IDF	0.25^{***}	0.20^{***}
BodyText:50TF-IDF	0.14^{***}	0.12^{***}	N/A	N/A	N/A	N/A	N/A	N/A
BodyText:LDA	0.03	0.01	Dir:LDA	0.54^{***}	0.43^{***}	Plot:LDA	0.37^{***}	0.34^{***}
BodyText:Senti	-0.02	-0.02	N/A	N/A	N/A	N/A	N/A	N/A
AuthorBio:TF-IDF	0.15^{***}	0.12^{***}	N/A	N/A	N/A	N/A	N/A	N/A
AuthorBio:LDA	0.11^{***}	0.09^{***}	N/A	N/A	N/A	N/A	N/A	N/A

4.4.3 Differences in Predictive Model Performances

Table 4.7 presents the performance of the predictive models in the news, recipe, and movie domains. In relative terms, the *different types of models* perform similarly. While Lasso is the best performing model in the news domain, Ridge is found to be the better model in the recipe and movie domains. Most notably, we see that models from the news domain are much

less accurate than those in the recipe and movie domain, further suggesting that the similarity functions do not translate well into the news domain.

Table 4.7: Results of predictive models in the news, recipe, and movie domains. Data from the recipe and movie domains were obtained from Trattner and Jannach [46].
The best performing model in each domain is marked as bold.

	News Articles ($n = 2,169$)				**Recipe** ($n = 1,539$)[46]				**Movie** ($n = 1,395$)[46]			
Method	RMSE	R^2	MAE	p	RMSE	R^2	MAE	p	RMSE	R^2	MAE	p
	Model performance (All features)											
All (RF)	0.9219	0.2982	0.7643	0.2411	0.8958	0.4734	0.6787	0.6425	0.8807	0.3543	0.7007	0.5943
All (GB)	0.9177	0.3123	0.7520	0.2005	0.8805	0.4921	0.6672	0.6390	0.8844	0.3489	0.7029	0.5897
All (Ridge)	0.9141	0.3257	0.7459	0.2922	**0.8654**	**0.5063**	**0.6651**	**0.6625**	**0.8745**	**0.3628**	**0.6926**	**0.6019**
All (LM)	0.9120	0.3289	0.7453	0.2896	0.8700	0.5022	0.6668	0.6512	0.8752	0.3616	0.6929	0.6007
All (Lasso)	**0.9101**	**0.3339**	**0.7480**	**0.2720**	0.8873	0.3574	0.7286	0.5952	0.8873	0.3574	0.7286	0.5952
	Baselines											
Mean	0.9652	0.0000	0.8122	-0.0010	1.2292	0.4995	1.0433	0.0184	1.0942	0.5001	0.9140	0.0001
Random	0.9659	-0.0226	0.8125	-0.0300	1.2290	0.0010	1.0435	0.0489	1.0948	0.0061	0.9140	0.0381

Table 4.8 presents the performance of the Ridge model in the second approach to constructing predictive models, for the news, recipe, and movie domains. Similarly to news, demographic data was found to have minimal impact on the performance of the model in the movie domain. However, in the recipe domain, the additional features proved to be useful and were of statistically significant difference to the original Ridge model ($p < 0.05$) [46]. One thing to note here is that the questions that were asked in the recipe domain were differently phrased. While the questions in both the news and movie domains asked questions in the form of "how often do you ...", the questions in the recipe domain asked questions directly pertaining to the *experience* of the user. In the news domain, it might have been a better approach to ask questions like "how familiar are you with national politics?". While the best-performing additional feature model in the news domain *age* was found to be of insignificant difference, it is interesting to see that *gender* was a better additional feature than age in both the recipe and movie domains. This might suggest that different demographics influence the predictive models in different ways, depending on the domain.

Table 4.8: Results of predictive models in the news, recipe, and movie domains when additional features are considered. Data from the recipe and movie domains were obtained from Trattner and Jannach [46].

	All (Ridge) + additional features			
Method	RMSE	R^2	MAE	ρ
	News Articles ($n = 2,169$)			
Ridge	0.9141	0.3257	0.7459	0.2922
News website visits	0.9164	0.3207	0.7463	0.2819
Num. days reads news	0.9186	0.3215	0.7476	0.2812
Age	**0.9081**	**0.3435**	**0.7338**	**0.3130**
Gender	0.9125	0.3314	0.7456	0.2896
All additional features	0.9099	0.3412	0.7358	0.3049
	Recipe ($n = 1,539$) [46]			
Ridge	0.8654	0.5063	0.6651	0.6625
Recipe Website Visits	0.8684	0.5031	0.6668	0.6558
Home Cooking	0.8648	0.5065	0.6646	0.6605
Cooking Experience	0.8631	0.5079	0.6615	0.6615
Age	0.8562	0.5170	0.6570	0.6699
Gender	0.8521	0.5203	0.6558	0.6755
All User Characteristics	**0.8393**	**0.5336**	**0.6448**	**0.6865**
	Movie ($n = 1,395$) [46]			
Ridge	0.8745	0.3628	0.6926	0.6019
Movie Website Visits	0.8757	0.3615	0.6927	0.5999
Num. Days Watches Movie	0.8754	0.3667	0.6933	0.6049
Age	0.8764	0.3613	0.6931	0.6007
Gender	0.8770	0.3604	0.6946	0.5998
All User Characteristics	**0.8732**	**0.3682**	**0.6906**	**0.6064**

The final approach involved constructing a Ridge model for each *type of feature*. The results from all three domains are presented in Table 4.9. The results generally further em-

phasizes the importance of the different types of features, and that some types of features are more important in some domains than in others. For example, the best-performing models in the news and recipe domains are long-text features, i.e. the *BodyText* in news and *Directions* in recipe. This is contrasted by the movie domain, where the *Genre* model, a keyword feature, markedly outperforms the rest, including *Plot*, movie's only long-text feature. Overall, both the movie and recipe domains include different types of features that are all representative of an item's information. They both benefit greatly from leveraging all features, which is shown by the stark contrasts in performance. On the other hand, the news domain is not better off using all features. Here, the difference between the best-performing single-feature model *BodyText* and the *All features* model is found to not be statistically significant. This certainly suggests that the similarity functions developed are either not suited for the task at hand, or that some of the features available in the news domain are not representative of an article's information.

Table 4.9: Results of predictive models in the news, recipe, and movie domains when additional features are considered. Data from the recipe and movie domains were obtained from Trattner and Jannach [46].

Method	RMSE	R^2	MAE	ρ
	News Articles ($n = 2,169$)			
Subcat	0.9554	0.1406	0.7943	0.1106
Title	0.9618	0.0889	0.8071	0.0759
Image	0.9548	0.1495	0.7913	0.1590
Author	0.9568	0.1333	0.7991	0.0724
Date	0.9616	0.0911	0.8070	0.0813
BodyText	**0.9141**	**0.3244**	**0.7514**	**0.2847**
AuthorBio	0.9561	0.1414	0.7991	0.1268
	Recipe ($n = 1,539$) [46]			
Title	1.0245	0.3079	0.8348	0.5278
Image	1.0680	0.2478	0.8706	0.4969
Ingredients	0.9449	0.4096	0.7493	0.6080
Directions	**0.9390**	**0.4190**	**0.7480**	**0.5998**
	Movie ($n = 1,395$) [46]			
Title	1.0613	0.0615	0.8939	0.2437
Image	1.0460	0.0875	0.8681	0.2939
Plot	0.9786	0.2029	0.8105	0.4476
Genre	**0.9075**	**0.3140**	**0.7299**	**0.5593**
Stars	1.0729	0.0515	0.9041	0.2201
Directors	1.0885	0.0132	0.9149	0.1040
Date	1.0158	0.1385	0.8422	0.3717

Chapter 5

Conclusions and Future Work

This Chapter concludes the thesis by summarizing the findings from the study, possible limitations of the approach, and provides possible directions going forward. Additionally, it provides a brief introduction to the tools developed for the thesis, and how the results can be reproduced.

The central theme of this thesis was to further explore the approach designed by Trattner and Jannach [46], in which they learned a combined similarity function to predict similar items by leveraging human judgments of similarity as a gold standard. To further understand this approach, the approach was extended into the news domain for this thesis. To do this, the Washington Post corpus was processed to be more usable in the context of the approach. The development of the similarity functions, most of which originated from the work of Trattner and Jannach [46], followed. After computing the similarity of pairs of news, the resulting scores were used to obtain a diverse dataset to use in the user study, where users assessed the similarity of one pair of news articles at a time. Additionally, they answered questions regarding their user characteristics and demographics. The results from the user study were then used to understand the correlations between the different similarity functions and the human judgments. Afterward, the results were used to construct predictive models in three approaches, using different degrees of data; one using all features, one including user characteristics and demographic data, and one where only a single type of feature was used for each model. After conducting the analysis, the results from the news domain was compared to the recipe and movie domains, which Trattner and Jannach [46] leveraged in their work. The findings of this master thesis can be summarized as follows:

RQ1: Which types of features, and which specific metrics best represent user perception of similarity? To answer this research question, two correlation analyses were conducted. In the first analysis, each metric, i.e. similarity function, was analyzed against the human judgments. Here, the *BodyText:TFIDF* metric was found to have the best correlation by far. Other notable correlations included *Image:EMB*, *Subcat:Jacc*, *Author:Jacc*, *AuthorBio:TFIDF*, and *AuthorBio:LDA*, which were all significant and had modest correlations. In the second analysis, each *type of feature* was analyzed against the human judgements. The analysis suggested that *All* features best represent user perception of similarity, followed by *Subcat* and *AuthorBio*. This was surprising since *BodyText* was found to have the strongest, single metric correlation in the first analysis. *BodyText*'s low overall correlation was found to be due to the *BodyText:Senti* metric, which had a negative correlation. Moreover, it was also found that *Title* only correlates a little with the human judgments, despite users reporting it to be of high importance in their similarity assessment. On the contrast, both author features (*Author*, *AuthorBio*) were found to have modest correlations, despite users reporting them to have very low importance, and there is little usage of them in earlier recommender scenarios.

RQ2: Which combination of features is best suited for predicting user-perceived similarity levels? Overall, the correlations to the human judgments and the performance of the predictive models suggested that *BodyText*, *Image*, and *AuthorBio* are the features best suited for the task. However, it was found that there is not a statistically significant difference in using all features in contrast to only *BodyText* in predictive models, which suggested a call for different methods to compute the similarity using these features. Furthermore, it was also found that considering user characteristics and demographics in the models did not boost the performance to any significant degree. These findings suggest that the main focus should be on leveraging *BodyText*, but that other features show potential given the right metric.

RQ3: How do we compare to the recipe and movie domains? Overall, computing similarity of texts (movie's plot, recipe's directions, news' body text) was found to be useful in each domain, given the right metric is used. In each domain, these were found to be among the highest correlations, as well as among the strongest predictors. Title and image were found to be less representative of the users' judgments in the news and movie domains while producing the best results for the recipe domain. The results suggest that the news domain call for different features or metrics than in the movie or recipe domains. Although all predictive models and features produced significant results ($p < 0.001$), the predictive models for news

were much less accurate than those for recipes and movies.

5.1 Limitations and Future Work

This study, like any other, is not without limitations, the largest of which is the missing step introduced in Trattner and Jannach [46], where they validated the results from the initial study by conducting additional studies using the strongest predictive model. The results presented in Section 4 might only apply to the Washington Post corpus. Extending upon this, since news decay quickly, the news might have been more relatable, thus easier for user participants to assess, if more recent news was leveraged, i.e. from an RSS feed. Additionally, the results might only apply to political news articles, which this study was focused on. Since this study is essentially an extension upon the work of Trattner and Jannach [46], it leverages many of the same methods for computing similarity. Thus, further investigations using different methods, i.e. metrics paired with features, are called for to further understand the various features present in the domain, and to understand which combination of features are best used in predictive models. One possibility is to explore modifications of TF-IDF, i.e. SF-IDF and CF-IDF, which have shown promise in previous work. Furthermore, to determine the viability of using the sentiment of articles to predict similarity, other methods of computing sentiment should be explored.

5.2 Open Science

To make this study reproducible, all code is shared freely along with the results from the study. The original Washington Post dataset can be obtained by submitting a request to TREC[1], the organization responsible for the management of it. This Section provides a brief technical introduction to the tools developed for this thesis. Having obtained a copy of the original Washington Post dataset, the tools can then be used to process the dataset the same way as described in Section 3.1, and to compute the similarity using the functions described in Section 3.4.1.

Table 5.1 presents an overview of tools (libraries) used to compute similarity. All code

[1] https://trec.nist.gov/data/wapost/

used in this thesis can be found in a Github repository[2]. The Github repository also contains the code for the Web application behind the user study, and R-script used for the correlation analysis and predictive models. Additionally, an HTML-print of the results of running the script is included.

Table 5.1: Libraries and methods used to compute similarity.

Name	**Method**	**Website**
	Python	
NGram	Bi-gram (BI)	https://pythonhosted.org/ngram/ngram.html
NLTK	Jaccard, Stopwords-removal, Stemming	https://www.nltk.org/
pyjarowinkler	Jaro-Winkler (JW)	https://pypi.org/project/pyjarowinkler/
scikit-learn	Cosine similarity, TF-IDF	https://scikit-learn.org/stable/
Levenshtein	Levenshtein (Lev)	https://pypi.org/project/python-Levenshtein/
TextBlob	Sentiment (SENTI)	https://textblob.readthedocs.io/en/dev/
pylcs	Longest-common-subsequence (LCS)	https://pypi.org/project/pylcs/
gensim	Latent Dirichlet Allocation (LDA)	https://radimrehurek.com/gensim/
keras	Image embeddings (EMB)	https://keras.io/api/applications/vgg/
	Java	
OpenIMAJ	Brightness (BR), Sharpness (SH), Colorfulness (COL), Contrast (CO)	http://openimaj.org/

The data processing pipeline described in Section 3.1 is found in the *twpc_articles.py* (*twpc -> twpc_articles.py*) module in the Github repository. The module uses an object of the class *TWPC_Helper* from the *twpc_helper.py* module, on which the user can set different parameters for the program, i.e. batch-size, or setting it to only process a single category. When a processed version of the dataset is created, the *main.py* module (*sim -> main.py*) can be used to perform both stages of sampling and to compute the similarity.

[2] https://github.com/Overhaug/HuJuRecSys - Last updated 15.06.2020.

References

[1] J.-j. Aucouturier and F. Pachet. Music Similarity Measures : What ' s the Use ? *Ismir*, 2002.

[2] M. Bieliková, M. Kompan, and D. Zeleník. Effective hierarchical vector-based news representation for personalized recommendation. *Computer Science and Information Systems*, 2012.

[3] D. Billsus and M. J. Pazzani. Personal news agent that talks, learns and explains. In *Proceedings of the International Conference on Autonomous Agents*, 1999.

[4] D. Billsus and M. J. Pazzani. User modeling for adaptive news access. *User Modelling and User-Adapted Interaction*, 2000.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[6] T. Bogers and A. Van Den Bosch. Comparing and evaluating information retrieval algorithms for news recommendation. In *RecSys'07: Proceedings of the 2007 ACM Conference on Recommender Systems*, 2007.

[7] P. B.Thorat, R. M. Goudar, and S. Barve. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, 2015.

[8] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

[9] M. Capelle, M. Moerland, F. Frasincar, and F. Hogenboom. Semantics-based news recommendation. In *ACM International Conference Proceeding Series*, 2012.

[10] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[11] Y. Deldjoo, M. Elahi, M. Quadrana, and P. Cremonesi. Toward building a content-based video recommendation system based on low-level features. In *Lecture Notes in Business Information Processing*, 2015.

[12] M. S. Desarkar and N. Shinde. Diversification in news recommendation for privacy concerned users. In *DSAA 2014 - Proceedings of the 2014 IEEE International Conference on Data Science and Advanced Analytics*, 2014.

[13] R. Dong, M. P. O'Mahony, M. Schaal, K. McCarthy, and B. Smyth. Combining similarity and sentiment in opinion mining for product recommendation. *Journal of Intelligent Information Systems*, 2016.

[14] B. Fortuna, C. Fortuna, and D. Mladenić. Real-time news recommender system. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.

[15] F. Frasincar, J. Borsje, and L. Levering. A semantic web-based approach for building personalized news services. *International Journal of e-Business Research*, 2009.

[16] F. Garcin and B. Faltings. PEN recsys: A personalized news recommender systems framework. In *ACM International Conference Proceeding Series*, 2013.

[17] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. Offline and online evaluation of news recommender systems at swissinfo.ch. In *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, 2014.

[18] A. Gershman, T. Wolfe, E. Fink, and J. Carbonell. News Personalization using Support Vector Machines. *Proceedings of the SIGIR Workshop on Enriching Information Retrieval.*, 2011.

[19] F. Goossen, W. Ijntema, F. Frasincar, F. Hogenboom, and U. Kaymak. News personalization using the CF-IDF semantic recommender. In *ACM International Conference Proceeding Series*, 2011.

[20] D. J. Hauser and N. Schwarz. Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants. *Behavior Research Methods*, 48(1):400–407, 2016.

[21] W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom. Ontology-based news recommendation. In *ACM International Conference Proceeding Series*, 2010.

[22] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: An introduction*. Cambridge University Press, 2010.

[23] M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 1989.

[24] M. C. Jones, J. S. Downie, and A. F. Ehmann. Human similarity judgments: Implications for the design of formal evaluations. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007*, 2007.

[25] M. Karimi, D. Jannach, and M. Jugovac. News recommender systems – Survey and roads ahead. *Information Processing and Management*, 54(6):1203–1227, 2018.

[26] G. Kondrak. N-gram similarity and distance. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2005.

[27] P. Lenhart and D. Herzog. Combining content-based and collaborative filtering for personalized sports news recommendations. In *CEUR Workshop Proceedings*, 2016.

[28] L. Li and H. Jiang. The research of the development transportation countermeasures on the medium and small urban based on MNL model. In *CCIE 2011 - Proceedings: 2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering*, 2011.

[29] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. SCENE: A scalable two-stage personalized news recommendation system. In *SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011.

[30] A. Lommatzsch, B. Kille, F. Hopfgartner, and L. Ramming. NewsREEL multimedia at MediaEval 2018: News recommendation with image and text content. In *CEUR Workshop Proceedings*, 2018.

[31] P. Lops, M. de Gemmis, and G. Semeraro. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*, pages 73–105. Springer, 2011.

[32] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang. Content-based collaborative filtering for news topic recommendation. *Proceedings of the National Conference on Artificial Intelligence*, 1:217–223, 2015.

[33] T. Luostarinen and O. Kohonen. Using Topic Models in Content-Based News Recommender Systems. *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, 2013.

[34] Y. Lv, T. Moon, P. Kolari, Z. Zheng, X. Wang, and Y. Chang. Learning to model relatedness for news recommendation. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, 2011.

[35] P. Messina, V. Dominguez, D. Parra, C. Trattner, and A. Soto. Content-based artwork recommendation: integrating painting metadata with neural and manually-engineered visual features. *User Modeling and User-Adapted Interaction*, 2019.

[36] N. Muralidhar, H. Rangwala, and E. H. S. Han. Recommending temporally relevant news content from implicit feedback data. In *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2016.

[37] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu. Using of jaccard coefficient for keywords similarity. In *Lecture Notes in Engineering and Computer Science*, 2013.

[38] J. S. Pedro and S. Siersdorfer. Ranking and classifying attractiveness of photos in folksonomies. In *WWW'09 - Proceedings of the 18th International World Wide Web Conference*, 2009.

[39] R. K. Pon, A. F. Cardenas, D. Buttler, and T. Critchlow. Tracking multiple topics for finding interesting articles. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.

[40] J. Qiu, L. Liao, and P. Li. News recommender system based on topic detection and tracking. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009.

[41] M. U. Real-time, C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec. Pixie: A System for Recommending 3 + Billion Items to. In *TheWebConf*, 2018.

[42] M. Rossetti, F. Stella, and M. Zanker. Contrasting offline and online results when evaluating recommendation algorithms. *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems*, pages 31–34, 2016.

[43] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document length normalization. *Information Processing and Management*, 1996.

[44] S. Soroka, L. Young, and M. Balmas. Bad News or Mad News? Sentiment Scoring of Negativity, Fear, and Anger in News Content. *Annals of the American Academy of Political and Social Science*, 2015.

[45] N. Tintarev and J. Masthoff. Similarity for news recommender systems. *{...} of the AH'06 Workshop on Recommender Systems {...}*, 2006.

[46] C. Trattner and D. Jannach. Learning to recommend similar items from human judgments. *User Modeling and User-Adapted Interaction*, 30(1), 2020.

[47] Y. Yao and F. Maxwell Harper. Judging similarity: A user-centric study of related item recommendations. In *RecSys 2018 - 12th ACM Conference on Recommender Systems*, 2018.

[48] K. F. Yeung and Y. Yang. A proactive personalized mobile news recommendation system. In *Proceedings - 3rd International Conference on Developments in eSystems Engineering, DeSE 2010*, 2010.

[49] L. Yujian and L. Bo. A normalized Levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

Appendix

Table 1: Sample dataset content feature statistics.

Feature	Mean	Median	Min	Max
Number of words in title	9.78	10	2	25
Number of characters in title	60.16	61	11	195
Article image brightness	0.37	0.35	0.04	0.98
Article image sharpness	0.24	0.2	0.03	1.27
Article image contrast	0.18	0.18	0.01	0.64
Article image colorfulness	0.17	0.16	0	0.73
Article image entropy	7.05	7.33	0.75	7.95
Number of words in article body text	768.44	637	6	10640
Number of characters in article body text	4676.99	3895.5	38	65641
Article body text sentiment	0.54	0.54	0.05	0.89
Date of publication	2015-01-04	2014-12-31	2012-01-10	2017-08-22
Number of words in author biographies	21.63	17	4	306
Number of characters in author biographies	140.32	115	33	1989
Number of authors	1.05	1	1	8

Table 3: A complete overview of the categories the respective subcategories (sections) were mapped to.

Category	**Subcategory** (Section)
Politics	Politics, Courts & Law, Courts And Law, Fact Checker, The Fix, Monkey Cage, Polling, Congress, White House, Right Turn, GovBeat, In the Loop, DemocracyPost, The Volokh Conspiracy, Federal Insider, 2chambers, The Fed Page, World Politics, Wonkblog, Post Politics, Think Tanked, PowerPost, Jennifer Rubin
Opinions	Opinions, Opinion, The Post's View, Act Four, Global Opinions, Local Opinions, Letters to the Editor, The Opinions Essay, The Plum Line, Post Opinión, Post Opinion, Alexandra Petri, Telnaes Cartoons, Toles Cartoons, Erik Wemple, In Theory, The Watch, Post Partisan, PostPartisan, Post Local, Blogs & Columns
Investigations	Investigations, Investigative
Tech	Tech, Consumer Tech, Future of Transportation, Innovations, Internet Culture, Space, Tech Policy, Video Gaming, The Intersect, The Switch, On I.T., Technology
World	World, Africa, Americas, Asia, Europe, Middle East, Drawing The World Together, The Americas, WorldViews, Asia & Pacific
D.C., & Md. & Va.	Maryland Politics, D.C., Md. & Va, The District, Maryland, Virginia, Crime & Public Safety, Public Safety, Going Out Guide, Restaurants & Bars, Transportations, Cars, All Opinions Are Local, D.C., Virginia Politics, D.C. Politics, Bars & Clubs, Md. Politics, Restaurants, On Faith Local, Maryland Terrapins, District of DeBonis, A House Divided, Local, Digger

Sports	Sports, NFL, MLB, NBA, NHL, Boxing & MMA, College Sports, D.C. Sports Bog, Fantasy Sports, Golf, High School Sports, Olympics, Soccer, Tennis, WNBA, Fancy Stats, Nationals/MLB, Capitals/NHL, AllMetSports, Washington Capitals, D.C. United/Soccer, Wizards/NBA, Redskins/NFL, Colleges, Early Lead, Washington Nationals, Washington Wizards, Soccer Insider, London 2012 Olympics, The Insider
Arts & Entertainment	Arts & Entertainment, Arts and Entertainment, Books, Movies, Museums, Music, Pop Culture, Theater & Dance, TV, Comic Riffs, Celebrities, Book Club, Fall TV Preview, Entertainment, Video
Business	Business, Economic Policy, Economy, Energy, Health care, Leadership, Markets, Real Estate, Small Business, On Small Business, On Leadership, Where We Live, Capital Business, Fiscal Cliff, Keystone Highway, World Business
Personal	Finance: Personal Finance, Get There
Education	Education, Higher education, Grade Point, Answer Sheet
Food	Food, Voraciously
Health	Health, Medical mysteries, Wellness, Health & Science, Health Science, To Your Health
History	History, Made by History, Retropolis
Holiday	Guide Holiday Gift Guide, Holiday Guide 2012
Immigration	Immigration
Lifestyle	Lifestyle, Advice, Fashion, Home & Garden, Inspired Life, KidsPost, Parenting, Relationships, Reliable Source, Travel, Solo-ish, Tripping, Weddings, Style, On Parenting
Magazine	Magazine

National Security	National Security, Foreign Policy, Justice, Military, Josh Rogin, Checkpoint
Outlook	Outlook, Book Party, Five Myths, PostEverything
Science	Science, Animals, Animalia, Speaking of Science
Weather	Weather, Capital Weather Gang
Photography	Photography, In Sight, Your Photos
Puzzles & Games	Puzzles & Games, Comics, Horoscopes
Climate & Environment	Climate & Environment, Energy & Environment, Energy and Environment
Climate Solutions	Climate Solutions
Religion	Religion, Acts of Faith, On Faith
National	National, Post Nation, On Giving
Obituaries	Obituaries
Transportation	Transportation, Gridlock, Dr. Gridlock
By The Way	By The Way
Carolyn Hax	Carolyn Hax
Launcher	Launcher
The Lily	The Lily
Discussions	Discussions
Jobs	Jobs
Social Issues	Social Issues
She The People	She The People
Achenblog	Achenblog
ComPost	ComPost
Express	Express

El Tiempo Latino	El Tiempo Latino
Deportes	Deportes
Ask The Post	Ask The Post
Morning Mix	Morning Mix
PR	WashPost PR Blog, PR, Community Relations
America Answers	America Answers
Tablet	Tablet
Test	Test, Test , test
Storyline	Storyline
Rampage	Rampage
Events	Events
Ads	Brand Connect, Brand Studio
Crime	True Crime, Crime
Video	Post Politics Live, Washington Post Live, Post Live

"The Same, But Different": Comparing News Similarity Functions Across Recommender Domains Using Human Judgements

ANONYMOUS AUTHOR(S)

Similar item recommendation is a common starting point in various domains, including *news*. Such "more like this" approaches rely on similarity functions (i.e., describing how alike two items are) to generate new suggestions to a user, based on a given reference item. However, it is unclear to what extent similarity functions from one domain (e.g., movies) can be used in another domain (e.g., news). Moreover, what similarity functions describe as two similar items, might be quite different from how similar users perceive them to be.

In this study, we designed similarity functions for news item retrieval using human judgements of similarity. We performed a recommender study in which users assessed the similarity of nine news articles, which we benchmarked against various similarity functions. In turn, we compared our results with data from an earlier study on movie and recipe recommender systems. We find human judgements of similar news articles to be weakly correlated to our similarity functions, but show promising results for text-based similarity. In addition, we point out important differences between the news, movie, and recipe domains for similar item retrieval.

CCS Concepts: • **Human-centered computing** → *User studies*; • **Information systems** → **Recommender systems**; **Similarity measures**.

Additional Key Words and Phrases: Similar Item Recommendation, News Recommender Systems, Human Judgement

ACM Reference Format:
Anonymous Author(s). 2020. "The Same, But Different": Comparing News Similarity Functions Across Recommender Domains Using Human Judgements. In *RecSys '20: ACM Conference on Recommender Systems, September 22–26, 2020, Online, Worldwide.* ACM, New York, NY, USA, 9 pages. https://doi.org/xx.xxxx/xxxxxx.xxxxxx

1 INTRODUCTION

News recommender systems face a number of domain-specific challenges [15]. Compared to other domains, news articles are quite volatile, as they become obsolete quickly, may be updated, or are superseded by breaking news events [5, 7, 21]. Moreover, user preferences may also depend on contextual factors, such as time of day or location [8, 10].

Many news websites employ, in part due to cold-start problems [8, 10, 15], content-based recommender systems [19]. Such systems provide suggestions that are similar to a central reference item [29]. Such related-item recommendations are used in various domains (e.g., "More on this Story" recommendations at the BBC website), helping users to explore commodities (e.g., products, news, etc.) that are similar to, but also slightly different from an item that is currently being inspected [11, 12, 32]. These systems are also implemented at e-commerce platforms (e.g., Zalando) to keep users engaged with the service, particularly supporting those users who have a specific product goal in mind (e.g., a red dress) [31]. Many similarity functions are engineered through expert knowledge or offline validation studies [4, 27], others are constructed using the support of human evaluations of inter-item similarity [13, 31, 32]. The latter class of functions

Manuscript submitted to ACM

is able to reveal which features (e.g., a movie's title or plot [29, 32]) of reference items are relevant in assessing the similarity between them, by corroborating similarity judgements with different similarity functions per feature [29].

The importance of each feature seems to be partially domain-dependent [29], but research on this topic is limited. Although a few recommender studies have been published on how human judgements relate to different item features, these mostly concern 'taste-related' domains such as movies, music, and fashion [1, 29, 31, 32]. In contrast, a comprehensive approach for *news recommenders* is lacking [23], even though news platforms commonly use similar item recommendation (e.g., "More like this"). This may be due to news items requiring algorithms and evaluation metrics that differ from other recommender domains [15, 17, 24], leading to a different valuation of what features are relevant in similarity assessments, both for systems and humans. For instance, similarity based on date and time makes sense for news articles (e.g., 'Recent Stories' sections in online newspapers), but is less relevant for recipes.

This paper explores how human judgements of similarity between pairs of news articles are related to a set of similarity functions. Based on human judgements, we develop a model for news, and perform a cross-domain comparison (movies, recipes) using data from a recent study of Trattner and Jannach [29]. We posit the following **[Research Question]**: *How does human judgement of similarity relate to feature-specific similarity functions in the news recommender domain, as well as across other domains?*

2 RELATED WORK

We review related work in computer science on different approaches to inter-item similarity assessment. We discuss work that has explored the relation between similarity functions (e.g., Jaccard Index) and user similarity judgements.

2.1 Determining Item Similarity

Similar items recommendations are used by content-based recommender systems. Their goal is to identify *unseen* or *novel* items that are similar to those a user has interacted with or has elicited preferences for [12, 19], and presenting them to a user. Content-based approaches employ domain-specific features (e.g., plot text in movies) to assess similarity scores for between different items. These are formalised in a number of similarity functions that can be used for this purpose (without using human judgements), extracting information from different features [32]. Such item-based retrieval does not suffer as much from cold-start problems as approaches that are based on user activity [9].

The similarity functions that are relevant in news recommender research are summarised in Table 1, section 2.3. Due to space reasons, we cannot discuss all details of each function, please refer to [29], Table 10 for computational details.

One possibility is to derive vectors from items that a user has liked or which may be recommended in the future. *Term-frequency inverse document frequency* (TF-IDF) is a *vector space model* commonly used to create such vectors: $TF-IDF(t,d,D) = TF(t,d) * IDF(t,D)$, where $TF(t,d)$ denotes the number of times a term appears in a document, and $IDF(t,D)$ denotes the number of documents a term appears. Subsequently, the similarity between the vectors of liked and unseen items can be computed using Cosine similarity: $Cos(A,B) = \frac{A*B}{||A|||B||}$ [3]. As shown in Table 1, these are used for different text features (e.g., title, body text).

A simpler approach is proposed by [12], who derive a set of keywords from an item. For example, a book recommender could compute the similarity between a book vector ($b1 = fantasy, epic, bloody$) and another book vector ($b2 = fantasy, young, dragons$), using a *Jaccard coefficient*: $Jacc(b1,b2) = \frac{|b1 \cap b2|}{|b1 \cup b2|}$ [25]. Depending on the task, there are various similarity metrics available, such as *Dice coefficient* [12], the Levenshtein (also called the *edit distance*), LDA (Latent Dirichlet Allocation), etc. TF-IDF is one of the most commonly used methods in information-retrieval scenarios. Although it has been outperformed by other measures such as BM25 [23], it is still used regularly [3, 29].

Table 1. Features (i.e., cues) and similarity functions used in related news recommender literature.

Feature	Description & Relevant Articles
Title	Okapi BM25, LM-DIR, LM-JM, Cosine similarity, Language models [23]; TF-IDF [33];
Main text	Okapi BM25, LM-DIR, LM-JM, Cosine similarity, Language models [23];
Abstract	Okapi BM25, LM-DIR, LM-JM [23];
All text	TF-IDF & K-Nearest Neighbour [2, 3]; Cosine Similarity, Decision Trees (ID3) [3]; Overlap Coefficient [6]; Probabilistic Latent Semantic Indexing [16]; Latent Dirichlet Allocation [16, 22]; Fisher Kernel function (PLSA) [20];
Image labels	Pre-configured models [18];
Date of publication	Pre-/post-filtering, recency modeling [15];

In similar item computations, user-trace algorithms such as KNN (k-nearest-neighbor) and cosine similarity are commonly used to predict rating values (cf. [29]). In related work, Yao and Harper [32] examine how to use related-item similarity computations to determine what video a user should watch next. For example, YouTube's 'up next' function uses similarity estimates to re-rank a list of possible suggestions, based on the video a user is currently viewing.

2.2 Human judgements of similarity

An important question is to what extent similarity functions and related-item recommendations reflect a user's similarity assessment of two or more items. This could be problematic if a user either ignores or overvalues different item features compared to what is computed [29], or if a user's judgement suffers from biases not considered by a recommender [31]. With regard to the first category, a number of studies examine how user similarity assessments complement established similarity functions. Most prominently, Trattner and Jannach [29] contrast user similarity assessments to a set of functions for the movie and recipes recommender domains. They highlight that some cues (i.e., features), such as title, directions and images for recipes, and genre for movies, strongly correlate with user similarity judgements.

A couple of studies also examine this problem using a more psychological lens [1, 31]. They point out that humans are at times inconsistent in their judgements, which is also observed in Judgement and Decision-making research (see [14, 26, 30]). Recent work from Winecoff et al. [31] present similar item retrieval functions that are 'psychologically-aware', using the Tversky contrast model [30] to better predict human judgements of similarity in fashion recommendation. Although this contrasts with more 'psychology-naive' functions as Jaccard similarity (cf., [28]), our work focuses on the representativeness of more traditional information retrieval functions found in [29].

2.3 News recommender systems

To assess similarity in news, recommender algorithms focus primarily on textual representations of items. These approaches are usually geared towards utilising the main text or title of the news items, and ignore most other textual features such as the author [3, 15]. In contrast, while images are used often in other domains (e.g. recipes [29]), they are used much less frequently in news [15].

Table 1 provides an overview of the news features and accompanying similarity functions used in computer science. The textual features of news articles have been computed in various ways, based on different information (e.g., body text Decision Trees, title TF-IDF) [23, 33]. Moreover, although the presented images and an article's date of publication are used less frequently [15, 18], they are particularly important for cross-domain comparisons [29]. These functions are also used in the current study to compute similarity between different news item features, based on previous research.

3 METHOD

We investigate how human judgements of similarity in a news recommender system relate to established similarity functions. We describe the contents of our news article dataset, and contextualise them using examples from other domains. Then, we reiterate the similarity functions employed in our user study, and describe the study's design.

3.1 News Database Descriptive Statistics

Dataset. Our dataset comprised 310,577 articles from the Washington Post, published between January 2012 and August 2018. *All of our materials (dataset, processing, code) will be put online for open access re-use after review.* Articles were obtained from the TREC Washington Post Corpus (https://trec.nist.gov/data/wapost/). The following features were used for our similarity functions: a news article's title (**Title**), its author (**Author**), the author's bio (**AuthorBio**), publication date (**Date**), text (**BodyText**), its sub-category (**Subcat**, e.g., 'Fact checking'), and the displayed image (**Image**).

Feature Comparison Across Domains. As different features might contribute to inter-item similarity in different domains, we performed a cross-domain comparison between news articles (i.e., current user study) and movies and recipes [29]. The descriptive statistics of common features provided initial evidence for differences across domains (cf. Table 16 in [29]). For example, titles of news articles tended to consist of more words (M=9.78, Max=25) than titles of movies (M=2.79, Max=14), or recipes (M=3.84, Max=13). Moreover, the amount of 'body text' words in news articles was quite high (M=768), compared to movie plots (M=51) and recipe directions (M=111) [29]. For a fairer cross-domain comparison, we also assessed text-based similarity using the first 50 words in a news article.

3.2 News Recommender User Study

Procedure and Measures. To compare our similarity functions to human judgements, we used our database of news articles and relevant similarity functions from earlier research to design a recommender user study. Figure 1 depicts a mock-up of the main application, showing from top to bottom a news article's subcategory, title, image, author (a bio could also be inspected), date and time, body text (first 50 words), and all text if a user clicked 'read more'.

Users were presented ten pairs of such news articles (of which one was an attention check). They were asked to assess their similarity on a 5-point scale (from 'completely different' to 'they are more or less the same'; see Figure 1). Moreover, we also inquired on a user's familiarity with each article (5-point scales) and their similarity assessment's level of confidence (5-point scale). In addition, we collected info on a user's demographics and news usage frequency.

Participants. We recruited participants at Amazon MTurk. In line with previous research [29], we only recruited USA-based participants (NB: all news articles are USA-based), who had an average hit acceptance rate of 98% or higher and had done at least 500 HITs in the past. Eventually, a total of 401 participants completed our study, taking between 3-5 minutes to complete the task[1]. However, much to our surprise, *only* 241 (60%) of them passed our attention check (53% male), who each provided 9 similarity ratings and 1 attention check.

Our sample comprised adult of various age groups, but most fell between 25-34 (33.2%) and 35-44 (30.3%). The majority of participants reported to visit news websites at least once a week (66%), while others reported to rarely do so. In fact, 60 participants reported to read news every day of the week, while 50 reported to never read any news.

Relevant Similarity Functions. We contrasted the similarity assessments of our users with feature-specific similarity scores. Due to space reasons, we could not discuss all mathematical functions; refer to Table 10 in [29] for a full list. For each pair of presented articles, we computed similarity scores for their subcategories, title, presented images,

[1] Participants were compensated with 0.5 USD which was above the federal avg. income level and was evaluated as fair payment on https://turkerview.com/.

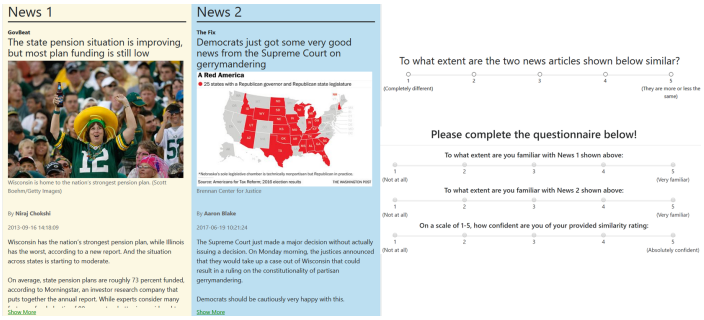

Fig. 1. Example of a pair-wise similarity assessment in our web application (on the left). Users were asked to assess the similarity of two presented news articles (on the right), as well as how familiar they were with the articles and the confidence level of their judgement. Note: this is a mock-up. In the actual design, the questions were positioned either above or below the news articles.

their authors (including bio) and publication dates, and their body text (first 50 words and full text). For article titles, we computed different distance-based similarity scores, such as Levenshtein (LV): $sim(r_i, r_j) = 1 - |dist_{LEV}(r_i, r_j)|$, but also Jaro-Winkler (JW), Longest Common Sequence (LCS), Bi-Gram (BI), and LDA Cosine. For image similarity, we computed distance-based similarities for different attributes, such as brightness (BR), sharpness (SH), contrast (CO), colorfulness (COL) and entropy (EN): e.g., $sim(r_i, r_j) = 1 - |EN(r_i) - EN(r_j)|$, as well as a cosine-based similarity for image embeddings. For other features (subcategory, atuhor, date, text), we either used a Jaccard Index (Jacc), cosine-based TFIDF ('50TFIDF' denotes an article's first 50 words), cosine-based LDA, or distance-based sentiment (Sent) similarity.

4 RESULTS

News Features Usage. We examined to what extent participants used different features or cues to assess similarity between news articles. Figure 2 (on the left) summarises the results for participants who passed the attention check, juxtaposed against cue usage of movies and recipes in Trattner and Jannach [29] on the right. On average, an article's title (M=4.2) and body text (M=4.4) were considered most often, while the text's sentiment (M=3.7) and an article's subcategory (M=3.2) saw above average use. In contrast, author features, publication date, an article's image were considered less important to assess inter-item similarity. Note that all differences between cues were significant (all: $p < 0.01$), based on a one-way ANOVA on cue usage and a Tukey's HSD post hoc analysis.

We observed a number of differences and similarities between common news, movie and recipe cues. Article and recipe titles were used often to assess similarity, but were less important in movies (M=2.8). Long texts (i.e., directions for recipes) were more comparatively used more often in news (i.e., comprising many words), while short text usage (i.e., plots of movies) was comparable to news body text use (M=4.5). In contrast, images were used more frequently in recipes to assess similarity (M=4.4) than in news (M=2.7) and movies (M=3).

Human Judgements vs Similarity Functions. To further address our **[RQ]**, we contrasted similarity functions of different cues to human judgement. Table 2 outlines the Spearman correlations between them, and juxtaposes them against correlation data from [29] on recipes and movies. We discerned between users who passed the attention check (ρ_{pass}), and all users (ρ_{all}). Although we found that a user's similarity rating correlated positively both with her familiarity for the given news articles ($\rho = 0.27^{***}$), and the stimulus trial (e.g., the 6th article presented; $\rho = 0.05^{***}$), only including 'familiar' users or excluding stimuli did not significantly affect the results presented in Table 2.

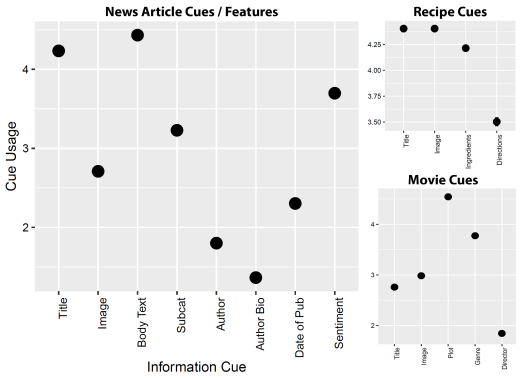

Fig. 2. Reported cue/feature usage (1 - did not use it; 5 - always used it) in the current news study on the left, compared to reported usage for movies and recipes in [29] on the right. Graphs adapted with permission.

Table 2 shows that the correlations for users who passed the attention check only increased a little compared to all users. Overall, most correlations were modest (all $\rho < 0.3$), suggesting that current news similarity functions did not fully reflect a user's judgement. Among all features, we found that full body text similarity (*BodyText:TFIDF*) had the strongest correlation: $\rho = 0.29$, $p < 0.001$, which was also the most commonly used cue in earlier news recommendation scenarios [15]. Although some users might have only inspected an article's first 50 words (cf. visible text in Figure 1; on average 15% of full body text), the *BodyText:50TFIDF* metric has a much lower correlation: $\rho = 0.14$, $p < 0.001$.

Table 2 shows that some functions do not represent a user's similarity judgement in the news domain, such as the text's sentiment (*BodyText:Sent*): $\rho = -0.02$. Surprisingly, although users indicated to use article titles, we only found weak correlations with all distance-based title similarity functions. Although most title metrics correlated significantly (not *Title:LDA*), they might not fully represent a user's perception. Possibly, both *Title:LDA* and *BodyText:LDA* might have suffered from poorly optimised parameters due to insufficient latent topic information.

Among all image similarity metrics, embeddings (*Image:EMB*) was revealed to have the strongest correlation, albeit still modest: $\rho = 0.17^{***}$. This metric, along with *BodyText:TFIDF*, author metrics (*Author:Jacc, AuthorBio:TFIDF*), and an article's subcategory (*Subcat:Jacc*), seem to best represent user similarity judgements.

Cross-domain Comparisons. Finally, we compared our set of news features to those used in [29]. Overall, computing body text similarity is useful in each domain if the right metric is used, as they were found to be among the highest ρ coefficients. In contrast, while producing the best results for recipes, titles and images were less representative of a user's judgement in the news and movie domains. Moreover, while genre ($\rho = 0.56^{***}$) and publication date ($\rho = 0.37^{***}$) had the strongest correlations for movies, their equivalents in news (*Subcat:Jacc*, *Date:ND*) were much less relevant, while a news article's author seemed to be more relevant than a recipe's author or movie's director.

To predict the user's similarity perception across domains, we developed simple similarity functions by each averaging the similarity metrics for 'title', 'image', and 'text'. We compare the results of our multilevel regression in Table 3, predicting similarity at a 0-1 scale. Although all models and features produced significant results ($p < 0.001$), the model for news similarity is less accurate than those for movies and recipes. This may in part be due to the large percentage of variance that is explained at the user level (i.e., 47.1%), indicating that users differed to a large extent in their rating behaviour (e.g., some users might have provided ratings between 1-3, while others did so between 3-5).

Table 2. Spearman correlations between similarity metrics in the news (current study), and recipe and movies domains (obtained from [29]). p_{pass} are correlations with users who passed the attention check. p_{all} denotes all users. $^{*}p < 0.05; ^{**}p < 0.01; ^{***}p < 0.001$.

	News Articles			**Recipes [29]**			**Movies [29]**	
Sim. Metric	ρ_{pass}	ρ_{all}	Sim. Metric	ρ_{pass}	ρ_{all}	Sim. Metric	ρ_{pass}	ρ_{all}
Subcat:Jacc	0.14^{***}	0.11	N/A	N/A	N/A	Genre:Jacc	0.56^{***}	0.53^{***}
Title:LV	0.06^{**}	0.04^{*}	Title:LV	0.48^{***}	0.38^{***}	Title:LV	0.19^{***}	0.18^{***}
Title:JW	0.05^{*}	0.03	Title:JW	0.46^{***}	0.35^{***}	Title:JW	0.16^{***}	0.16^{***}
Title:LCS	0.07^{***}	0.05^{**}	Title:LCS	0.50^{***}	0.40^{***}	Title:LCS	0.20^{***}	0.19^{***}
Title:BI	0.08^{***}	0.07^{***}	Title:BI	0.48^{***}	0.38^{***}	Title:BI	0.17^{***}	0.17^{***}
Title:LDA	0.02	0.00	Title:LDA	0.22^{***}	0.19^{***}	Title:LDA	0.01	0.01
Image:BR	0.10^{***}	0.07^{***}	Image:BR	0.18^{**}	0.14^{*}	Image:BR	0.22^{***}	0.20^{***}
Image:SH	0.06^{**}	0.03	Image:SH	0.16^{*}	0.11^{*}	Image:SH	0.10^{***}	0.08^{***}
Image:CO	0.05^{*}	0.05^{**}	Image:CO	0.29^{***}	0.20^{***}	Image:CO	0.03	0.03
Image:COL	0.05^{*}	0.03^{*}	Image:COL	0.09^{*}	0.07^{*}	Image:COL	0.15^{***}	0.14^{***}
Image:EN	0.07^{**}	0.05^{**}	Image:EN	0.34^{***}	0.28^{***}	Image:EN	0.15^{***}	0.09^{***}
Image:EMB	0.17^{***}	0.13^{***}	Image:EMB	0.44^{***}	0.34^{***}	Image:EMB	0.18^{***}	0.16^{***}
Author:Jacc	0.13^{***}	0.10^{***}	N/A	N/A	N/A	Dir:Jacc	0.10^{***}	0.07^{***}
Date:ND	0.09^{***}	0.08^{***}	N/A	N/A	N/A	Date:MD	0.37^{***}	0.35^{***}
BodyText:TFIDF	0.29^{***}	0.23^{***}	N/A	N/A	N/A	N/A	N/A	N/A
BodyText:50TFIDF	0.14^{***}	0.12^{***}	Dir:TFIDF	0.50^{***}	0.40^{***}	Plot:TFIDF	0.25^{***}	0.20^{***}
BodyText:LDA	0.03	0.01	Dir:LDA	0.54^{***}	0.43^{***}	Plot:LDA	0.37^{***}	0.34^{***}
BodyText:Sent	-0.02	-0.02	N/A	N/A	N/A	N/A	N/A	N/A
AuthorBio:TFIDF	0.15^{***}	0.12^{***}	N/A	N/A	N/A	N/A	N/A	N/A
AuthorBio:LDA	0.11^{***}	0.09^{***}	N/A	N/A	N/A	N/A	N/A	N/A

Table 3. Multilevel regression predicting a user's similarity judgement (set at 0-1), clustered at the user level. We averaged similarity functions in three categories (title, images, text), for the news, recipe, and movie domains. $^{*}p < 0.05; ^{**}p < 0.01; ^{***}p < 0.001$.

	Regression coefficients		
Sim. Metric	News Articles	Recipes [29]	Movies [29]
Title: $\overline{LV, JW, LCS, BI, LDA}$	0.22^{***}	0.39^{***}	0.35^{***}
Image: $\overline{BR, SH, CO, COL, EN, EMB}$	0.23^{***}	0.39^{***}	0.49^{***}
Text: $\overline{BodyText : TFIDF, 50TFIDF, LDA, Sent}; \overline{Dir/Plot : TFIDF, LDA}$	0.52^{***}	0.50^{***}	0.72^{***}
Constant	$-.86^{***}$	-1.04^{***}	-1.09^{***}
Within R^2	0.034^{***}	0.35^{***}	0.22^{***}
Overall R^2	0.013^{***}	0.28^{***}	0.20^{***}
Variance at user level	47.1%	21.1%	8.8%

5 CONCLUSION

The current study is a first attempt at developing a similarity function for the news domain using human judgements. Overall, most state-of-the-art cosine- and distance-based metrics only partially reflect a user's similarity judgement, as most correlations are modest at best. To best reflect user perceptions, recommender designers should rely on an article's body text, supported by image embeddings, article categories, and author information.

In line with [29], we have found further evidence that different domains call for different similarity functions. Whereas images are very important in recipe recommendations, their role is negligible in news similarity assessments. The promising results using text-based similarity metrics might also be applicable to other recommender domains.

Our model for news similarity assessment turned out to be rather inaccurate. However, this study's intention was not to develop the best functions or metrics possible, but to show how existing metrics would perform, as well as how they compare across domains. The news domain seems to require metrics that are less 'taste-related' than movies or recipes, but further research is needed to develop accurate ones, possibly by also using psychology as done by [31].

REFERENCES

[1] Jean-Julien Aucouturier, Francois Pachet, et al. 2002. Music similarity measures: What's the use?. In *ISMIR*. 13–17.

[2] Daniel Billsus and Michael J. Pazzani. 1999. Personal news agent that talks, learns and explains. In *Proceedings of the International Conference on Autonomous Agents*.

[3] Daniel Billsus and Michael J. Pazzani. 2000. User modeling for adaptive news access. *User Modelling and User-Adapted Interaction* (2000).

[4] Iván Cantador, Alejandro Bellogín, and David Vallet. 2010. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*. 237–240.

[5] Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. 2009. A case study of behavior-driven conjoint analysis on Yahoo! Front Page Today module. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1097–1104.

[6] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. 1999. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*.

[7] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*. 271–280.

[8] Toon De Pessemier, Cédric Courtois, Kris Vanhecke, Kristin Van Damme, Luc Martens, and Lieven De Marez. 2016. A user-centric evaluation of context-aware recommendations for a mobile news service. *Multimedia Tools and Applications* 75, 6 (2016), 3323–3351.

[9] Yashar Deldjoo, Mehdi Elahi, Massimo Quadrana, and Paolo Cremonesi. 2015. Toward building a content-based video recommendation system based on low-level features. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 45–56.

[10] Blaž Fortuna, Carolina Fortuna, and Dunja Mladenić. 2010. Real-time news recommender system. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 583–586.

[11] Dietmar Jannach and Gediminas Adomavicius. 2016. Recommendations with a Purpose. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 7–10.

[12] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender systems: an introduction*. Cambridge University Press.

[13] M Cameron Jones, J Stephen Downie, and Andreas F Ehmann. 2007. Human Similarity Judgments: Implications for the Design of Formal Evaluations.. In *ISMIR*. 539–542.

[14] Daniel Kahneman. 2011. *Thinking, fast and slow*. Macmillan.

[15] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems–Survey and roads ahead. *Information Processing & Management* 54, 6 (2018), 1203–1227.

[16] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: A scalable two-stage personalized news recommendation system. In *SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[17] Felicia Loecherbach, Judith Moeller, Damian Trilling, and Wouter van Atteveldt. 2020. The Unified Framework of Media Diversity: A Systematic Literature Review. *Digital Journalism* 0, 0 (May 2020), 1–38.

[18] Andreas Lommatzsch, Benjamin Kille, Frank Hopfgartner, and Leif Ramming. 2018. NewsREEL multimedia at MediaEval 2018: News recommendation with image and text content. In *CEUR Workshop Proceedings*.

[19] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

[20] Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-based collaborative filtering for news topic recommendation. *Proceedings of the National Conference on Artificial Intelligence* 1 (2015), 217–223.

[21] Tapio Luostarinen and Oskar Kohonen. 2013. Using topic models in content-based news recommender systems. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*. 239–251.

[22] Tapio Luostarinen and Oskar Kohonen. 2013. Using Topic Models in Content-Based News Recommender Systems. *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)* (2013).

[23] Yuanhua Lv, Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, and Yi Chang. 2011. Learning to model relatedness for news recommendation. In *Proceedings of the 20th international conference on World wide web*. 57–66.

[24] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting online performance of news recommender systems through richer evaluation metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 179–186.

[25] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of jaccard coefficient for keywords similarity. In *Lecture Notes in Engineering and Computer Science*.

[26] Robert M Nosofsky. 1991. Stimulus bias, asymmetric similarity, and classification. *Cognitive Psychology* 23, 1 (1991), 94–140.

[27] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[28] Michael Steinbach, Levent Ertöz, and Vipin Kumar. 2004. The challenges of clustering high dimensional data. In *New directions in statistical physics*. Springer, 273–309.

[29] Christoph Trattner and Dietmar Jannach. 2020. Learning to recommend similar items from human judgments. *User Modeling and User-Adapted Interaction* 30, 1 (2020), 1–49.

[30] Amos Tversky. 1977. Features of similarity. *Psychological review* 84, 4 (1977), 327.

[31] Amy A Winecoff, Florin Brasoveanu, Bryce Casavant, Pearce Washabaugh, and Matthew Graham. 2019. Users in the loop: a psychologically-informed approach to similar item retrieval. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 52–59.

[32] Yuan Yao and F Maxwell Harper. 2018. Judging similarity: a user-centric study of related item recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 288–296.

[33] Kam Fung Yeung and Yanyan Yang. 2010. A proactive personalized mobile news recommendation system. In *Proceedings - 3rd International Conference on Developments in eSystems Engineering, DeSE 2010*.

Table 2: Questions asked in the final stage of the user study.

Question	**Alternatives**
User characteristics & demographics	
What is your age?	<18, 18-24, 25-34, 35-44, 45-54, >55, N/A
What is your gender?	Male, female, other
Which of the following statements best describes your use of online newspapers (e.g. The Washington Post, The New York Times, etc)?	Daily, weekly, monthly, every three months, hardly use them
Over the course of a week, how many days do you access online newspapers?	0-7
Information cue usage	
I looked at the **subcategory** to estimate the similarity between the news	Likert scale 0(Totally Disagree)-5(Totally Agree)
I looked at the news **title** to estimate the similarity between the news	Likert scale 0(Totally Disagree)-5(Totally Agree)
I looked at the news **image** to estimate the similarity between the news	Likert scale 0(Totally Disagree)-5(Totally Agree)
I looked at the **author(s)** to estimate the similarity between the news	Likert scale 0(Totally Disagree)-5(Totally Agree)
I looked at the **date of publication** to estimate the similarity between the news	Likert scale 0(Totally Disagree)-5(Totally Agree)
I looked at the article **body text** to estimate the similarity between the news	Likert scale 0(Totally Disagree)-5(Totally Agree)
I looked at the **author biography** to estimate the similarity between the news	Likert scale 0(Totally Disagree)-5(Totally Agree)