Automated Identification of Media Bias by Word Choice and Labeling in News Articles

Felix Hamborg[1], Anastasia Zhukova[1], Bela Gipp[2]
[1] University of Konstanz
{firstname.lastname}@uni-konstanz.de
[2] University of Wuppertal
gipp@uni-wuppertal.de

ABSTRACT

Media bias can strongly impact the individual and public perception of news events. One difficult-to-detect, yet powerful form of slanted news coverage is bias by word choice and labeling (WCL). Bias by WCL can occur when journalists refer to the same concept, yet use different terms, which results in different sentiments being sparked in the readers, such as the terms "economic migrants" vs. "refugees." We present an automated approach to identify bias by WCL that employs models and manual analysis approaches from the social sciences, a research domain in which media bias has been studied for decades. This paper makes three contributions. First, we present NewsWCL50, the first open evaluation dataset for the identification of bias by WCL consisting of 8,656 manual annotations in 50 news articles. Second, we propose a method capable of extracting instances of bias by WCL while outperforming state-of-the-art methods, such as coreference resolution, which currently cannot resolve very broadly defined or abstract coreferences used by journalists. We evaluate our method on the NewsWCL50 dataset, achieving an F1=45.7% compared to F1=29.8% achieved by the best performing state-of-the-art technique. Lastly, we present a prototype demonstrating the effectiveness of our approach in finding frames caused by bias by WCL.

CCS CONCEPTS

• **Information systems~Information extraction** • Information systems~Web searching and information discovery

KEYWORDS

News slant, news bias, automated content analysis, automated frame analysis, entity perception, emotions, CAS, CAQDAS, NLP.

1 INTRODUCTION

Slanted news coverage, or *media bias*, can have severe effects on both individuals and society [23]. Changes in the words used in a news text can significantly alter the perception of the reported event [47, 51]. When referring to a semantic concept, such as an actor, location, or action, journalists can *label* the concept, e.g., "illegal aliens," and *choose from various words* to refer to it, e.g., "immigrants" or "aliens" [24]. Instances of bias by *word choice and labeling* (WCL) *frame* the referred concept differently [13, 14], resulting in different opinions on the concept [22]. For example, a frame may alter readers' opinion positively or negatively, or focus readers on different aspects of the reported topic, e.g., highlight economic effects of immigration while downplaying cultural effects [13]. State-of-the-art techniques, such as coreference resolution, cannot resolve such coreferences currently (see Section 2).

The study of biased news coverage has a long tradition in the social sciences going back at least to the 1950s [56], resulting in comprehensive models to describe media bias and sophisticated methods for its analysis. Despite their effectiveness, these analyses are mostly conducted manually, requiring significant effort and expertise [22]. Thus, they do not scale with the vast amount of news that is published in times of online journalism and pack journalism (cf. [33, 44]), and social scientists can conduct such analyses for only few topics in the past. In computer science, the models used to analyze media bias tend to be simpler compared to the models established in the social sciences (cf. [28, 39]). Correspondingly, their results are often inconclusive or superficial, despite the approaches being technically superior [22].

To find and analyze bias by WCL, we propose an interdisciplinary approach that imitates the procedure of deductive content analysis, an analysis method established in the social sciences to systematically analyze media bias, while taking advantage of state-of-the-art natural language processing (NLP). Given a set of articles reporting on the same event, our approach determines which frames are ascribed to which actors and other, also abstract, semantic concepts. This paper extends our prior research on the identification of bias by WCL. Compared to the research reported in [22, 24], this paper contributes the evaluation dataset NewsWCL50 (contribution C1), more comprehensive and better performing candidate merging methods (formerly called *candidate alignment*, contribution C2) and methods for the estimation of frame properties (formerly *EoR estimation*, contribution C3), and an in-depth quantitative and qualitative evaluation of C2 and C3.

978-1-7281-1547-4/19/$31.00 ©2019 IEEE
DOI 10.1109/JCDL.2019.00036

In Section 3, we describe the creation of NewsWCL50 (C1). Section 4 describes the target concept analysis task, which resolves phrases referring to the same semantic concept across the input articles (C2). Section 5 describes how the approach estimates which frames an article ascribes to the target concepts (C3). In Section 6, we evaluate both approaches (C2 and C3) using NewsWCL50 and discuss our findings.

2 BACKGROUND AND RELATED WORK

To analyze bias by WCL, social scientists conduct *content analyses* (CAs) or *frame analyses*, i.e., systematic reading and annotating (also called coding) of texts relevant to analysis questions or hypotheses. Therefore, researchers typically go through three phases [22]: (1) collection of articles, (2) training: multiple training CAs to create and refine coding rules in the so-called *codebook*, and (3) a deductive CA. The first training CAs are often *inductive* (2), i.e., coders read articles without specified instruction on how to code the text, only knowing the analysis question. Afterward, based on discussions and the codings, researchers create a codebook, which describes the coding goals, examples, and a list of rules, i.e., what to code and how. The training phase is closed when no further changes to the codebook are necessary, i.e., the coding rules are comprehensive and clearly understandable, which is often measured using the inter-coder reliability (ICR, also called inter-annotator agreement). The ICR measures the agreement between the codings by multiple coders. The final, deductive CA is then conducted by one or more coders, e.g., to speed up the coding process by parallelization, or, in very sophisticated CAs, to continuously verify the ICR [50]. Researchers use the codings from the deductive CA to accept or reject their hypotheses.

In CAs focusing on bias by WCL, social scientists analyze how articles frame specific actors, topics, and other concepts, coined *target concepts*. For example, whether a politician is shown as being incompetent [43], or outlets use emotional or factual language when reporting on a specific topic [45]. A common way to quantify bias by WCL is to assign one or more *frame properties* to each *frame device* (cf. [24]). A frame device is a phrase that yields a specific frame on a target concept [7]. Frame properties allow for categorization of such frames, e.g., the phrase "illegal aliens" could yield a frame that highlights the unlawfulness of a target concept named "foreigners." In this basic example, in a CA a coder would mark the phrase "illegal aliens," assign the frame property "unlawfulness" to it, and set "foreigners" as its target concept.

In our literature review on identifying different forms of media bias, we found that no automated system focuses on the analysis of bias by WCL [22]. A closely related approach investigates the frequency of affective words close to user-defined words [19], e.g., names of politicians. Another approach aims to find bias words by employing IDF [28]. In contrast to prior work, we seek to imitate the process of the practice-proven CA, which is well-established in the social sciences. Automating CA presents two main challenges: identifying and resolving coreferential target concepts (contribution C2), and estimating frame properties (C3).

When analyzing bias by WCL, techniques capable of identifying and resolving target concepts, which we technically coin *WCL candidates*, need to find phrases that refer to the same semantic concepts. Most related techniques are named entity (NE) recognition (NER) and coreference resolution. These methods reliably identify NEs, synonyms, pronominal and nominal coreferences (precision up to $p \approx 80\%$) [5, 8, 24, 40], such as "Mr. Trump, "US President," "he," and "Donald Trump." However, the techniques cannot resolve WCL candidates, e.g., "terrorists" or "freedom fighters," because often journalists refer to the same concept in a broad sense, and such coreferences and synonyms may not be valid commonly but only in a specific context, e.g., articles reporting on the same event [24]. Other relevant techniques, such as sequence labeling [42], require large amounts of training data, which do not exist currently (see also Section 7).

Techniques for the second task, estimation of frame properties, need to analyze how a WCL candidate is framed by its modifiers [6], i.e., words the candidate depends on. Grefenstete et al. look for positive or negative words close to user-defined search terms [19]. Another related technique is sentiment analysis [30], which is state-of-the-art for opinion mining, e.g., in product reviews. However, considering only the polarity insufficiently represents the complexity of framing caused by bias by WCL [22].

In conclusion, to bridge between the bias models and effective, practice-proven, yet effortful analysis concepts from the social sciences, and automated, efficient text analysis methods, we propose an interdisciplinary approach that imitates and automates the process of deductive CA.

3 NEWSWCL50: DATASET CREATION

To create *NewsWCL50* (contribution C1), the first open dataset for the evaluation of methods to automatically identify bias by WCL, we conducted a manual CA. NewsWCL50 consists of 50 news articles that cover 10 political news events, each reported on by 5 online US news outlets representing the ideological spectrum. The dataset contains 8656 manual annotations, i.e., each news article approximately 170 annotations. Despite the recently increased interest of the CS community in media bias, no existing dataset is suitable for the evaluation of finding fine-grained instances of bias by WCL. For example, the dataset of the Hyperpartisan News Detection task at SemEval 2019 focuses only on whether an article is strongly slanted or not [11], whereas we seek to find the frames of each semantic concept within an article.

3.1 Collection of News Articles

We selected ten political events that happened during April 2018, and manually collected for each event five articles. To ease the identification and annotation of bias by WCL, we aimed to increase the diversity of both writing style and content. Therefore, we selected articles published by different news outlets and selected events associated with different topical categories. We selected five large, online US news outlets representing the political and ideological spectrum of the US media landscape [38, 49]: HuffPost (formerly The Huffington Post, far left, abbreviated *LL*), The New York Times (left, *L*), USA Today (middle, *M*), Fox News Channel (right, *R*), and Breitbart News Network (far right, *RR*). News outlets with different slants likely use different terms when reporting on the same topic [22], e.g., the negatively slanted term

"illegal aliens" is used by RR whereas "undocumented immigrants" is used by L when referring to DACA recipients (cf. [24]).

To increase the content diversity, we aimed to gather events for each of the following political categories (cf. [18]): economic policy (focusing on US economy), finance policy, foreign politics (events in which the US is directly involved), other national politics, and global / interventions (globally important events, which are part of the public, political discourse).

Table 1 shows the collected events of NewsWCL50. One frequent issue during data gathering was that even major events were not reported on by all five news outlets; especially the far left or far right outlets did not report on otherwise popular events (which may contribute to a different form of bias, named *event selection* [22]). We could not find any finance policy event in April that all five outlets reported on; hence, we discarded this category.

Table 1: Events in NewsWCL50

ID	Date 2018	Category	Name	# Annotations
0	04/18	for	Pompeo's meeting in PRK	684
1	04/19	nat	Comey memos	711
2	04/20	glo	PRK suspends nuclear tests	720
3	04/20	nat	DNC sues Russia, Trump campaign	1153
4	04/24	for	Macron and Trump meeting	1064
5	04/26	for	Planning of Trump's visit to the UK	621
6	04/29	nat	Migrant caravan crosses into the US	938
7	04/30	nat	Delays of US metal tariffs	784
8	04/30	eco	Mueller's questions for Trump	881
9	04/30	glo	Iran nuclear files	720

3.2 Training Phase: Creation of the Codebook

The training phase was conducted on news articles not contained in NewsWCL50. We collected the articles as described in Section 3.1 but for different time frames. In a first, inductive CA, we asked three coders (students in computer science or political sciences aged between 20 and 29) to read five news articles and use MAXQDA, a content analysis software, to code any phrase that they felt was influencing their perception or judgment of a person and other semantic concept mentioned in the article. Coders were asked to (1) mark such phrases, and state which (2) perception, judgment, or feeling the phrase caused in them, e.g., affection, and its (3) *target concept*, i.e., which concept the perception effect was ascribed to. We then used the initial codings to derive a set of frame properties, representing perceived effects on the reader of a phrase, and coding rules.

Our desired characteristics of frame properties are on the one hand to be *general* so that they can be applied meaningfully to a variety of political news events, but on the other hand to be *specific*, allowing fine granular categorization (cf. [24]). Thus, during training, we added, removed, refined or merged frame properties, e.g., we found that "unfairness" was always accompanied by (not necessarily physical) aggression, and hence was better, i.e., finer granularly, represented by "aggressor" or "victim." We created a codebook including frame properties, coding rules, and examples. During training, we refined the codebook until we reached an ICR=0.65 (after six training cycles). The codebook is available as part of NewsWCL50 (see Section 8).

Table 2: Frame properties in NewsWCL50

Name (name of antonym in parentheses, if any)	# mentions	# mentions of antonym
Affection (refusal)	173	70
Trustworthiness (no trustw.)	43	120
Reason (unreason/irrationality)	72	84
Fairness / morality	6	
Confidence	65	
Easiness (difficulty)	2	99
Positive economy (neg. eco.)	24	35
Honor (dishonor)	30	17
Importance (unimportance)	242	15
Lawfulness (unlawfulness)	26	63
Power / leadership (weakness / passiveness)	517	173
Good quality / functioning (poor quality)	35	56
Aggressor (victim)	262	150
Safety (unsafety)	46	78
Positive (negative)	26	26
Other bias	172	

3.3 Deductive Content Analysis

The deductive CA was conducted by one coder and two researchers who reviewed and revised the codings to ensure adherence to the codebook (cf. [41, 50]). The two coding units are *target concepts*, i.e., semantic concepts that can be the target of bias by WCL, and *frame properties*, i.e., categorized framing effects caused by bias by WCL (see Section 3.2). Target concepts can be *actors* (single individua), *actions*, *countries*, *events*, *groups* (of individua acting collectively, e.g., demonstrators), other (physical) *objects*, and also more abstract or broadly defined semantic concepts, such as "Immigration issues," coined *misc* (also see Table 5). We derived the frame properties shown in Table 2 during the training phase.

Following the codebook, the coder was asked to code any relevant phrase that represents either a target concept or frame property. For each frame property, additionally the corresponding target concept had to be assigned. For example, in "Russia seizes Ukrainian naval ships," "Russia" would be coded as a target concept (type country), and "seizes" as a frame property (type Aggressor) with "Russia" being its target. Each mention of a target concept in a text segment can be targeted by multiple frame property phrases. More details on the coding instructions can be found in NewsWCL50's codebook. The dataset consists of 5926 target concept codings, and 2730 frame property codings. NewsWCL50 is openly available in an online repository (see Section 8).

4 TARGET CONCEPT ANALYSIS

Given a set of articles reporting on the same event, our system first extracts WCL candidates (contribution C2), i.e., semantic concepts that can be targeted by instances of bias by WCL. Afterward, the system analyzes the WCL candidates and nearby phrases to estimate the induced frame properties (C3). The system's analysis

pipeline shown in Figure 1 performs three tasks: *preprocessing* (Section 4.1), *target concept analysis* (Sections 4.1 and 4.2, C2), and *frame identification* (Section 5, C3).

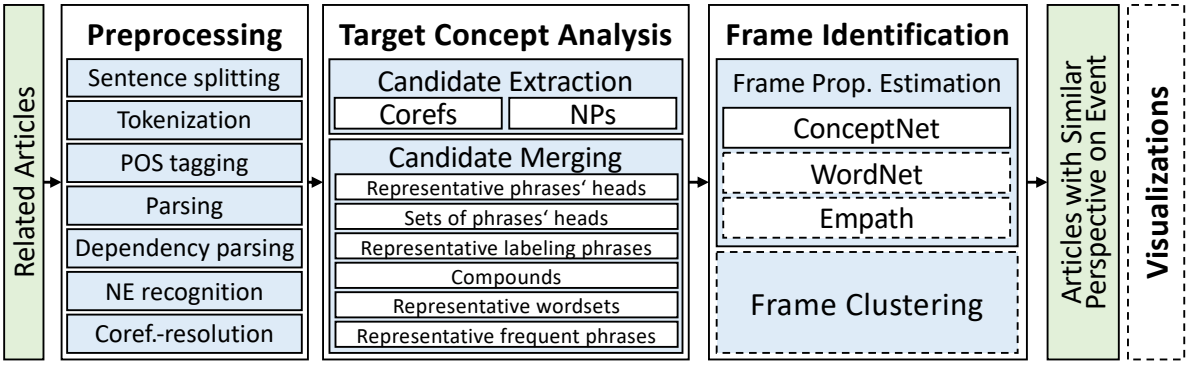

Figure 1: The three-tasks analysis pipeline preprocesses news articles, extracts and aligns phrases referring to the same semantic concepts, and groups articles reporting similarly on these concepts. Based on: [24]

In the target concept analysis, the goal of the first sub-task, *candidate extraction*, is to identify phrases that contain a semantic concept, i.e., phrases that could be the target of bias by WCL (Section 4.1). Currently, we only identify noun phrases (NPs). We coin such phrases *WCL candidate phrases*, and they compare to the mentions of target concepts annotated in the CA (except that we also coded VPs in the CA). The goal of the second sub-task, *candidate merging*, is to merge WCL candidates referring to the same semantic concept, i.e., groups of phrases that are coreferential (Section 4.2). Candidate merging includes state-of-the-art coreference resolution, but also aims to find coreferences across documents and in a broader sense (see Section 2), e.g., "undocumented immigrants" and "illegal aliens."

4.1 Preprocessing and Candidate Extraction

We perform state-of-the-art preprocessing, including part-of-speech (POS) tagging, dependency parsing, full parsing, named entity recognition (NER), and coreference resolution [8, 9], using Stanford CoreNLP with neural models where available, otherwise using the defaults for the English language [32].

As initial *WCL candidates*, we extract coreference chains and noun phrases (NPs). First, we extract each coreference chain including all its mentions found by coreference resolution as a single candidate. Conceptually, this can be seen as the first merging of candidates, since we merge all mentions of the coreference chain into one candidate. Second, we extract each NP found by the parser as a single candidate. We avoid long phrases by discarding any NP consisting of 20 or more words. If an NP contains one or more child NPs, we extract only the parent, i.e., longest, NP.

We set a *representative phrase* for each WCL candidate, which represents the candidate's meaning. For coreference chain candidates, we take the representative mention defined by CoreNLP's coreference resolution [54]. For NP-based candidates, we take the whole NP as the representative phrase. We use the representative phrases as one property to determine the similarity of candidates.

We also determine a candidate's *type*, which is one of the types shown in Table 3. Therefore, for each phrase in a candidate, we check whether its head is a person, group, or country, using the lexicographer dictionaries from WordNet [36] and NE types from NER [16], e.g., "crowd" or "hospital" are of type group. In linguistics, the head is defined as the word that determines a phrase's syntactic category [37], e.g., the noun "aliens" is the head of "illegal aliens," determining that the phrase is an NP. We count the frequencies of these three types over all phrases of a candidate and also count whether the heads are an NE or not. Lastly, we set the candidate's type as the most frequent lexicographer type. If the candidate contains at least one NE mention, we set the NE flag. For example, if most phrases of a candidate are NE mentions of a person, we set the candidate type person-ne. If the type is a person, we distinguish between singular and plural by counting the heads' POS types: NN and NNP for singular, NNS and NNPS for plural. If a candidate is neither a person, group, nor country, e.g., because the candidate is an abstract concept, such as "program," we set its type to misc. We use the candidate types to determine which candidates can be subject to merging, and for type-to-type specific merge thresholds.

Table 3: Candidate types identified during preprocessing

Candidate type	Definition	Example
person-ne	Single person (NE)	Trump
person-nes	Multiple persons (NE)	Democrats
person-nn	Single person (non-NE)	immigrant
person-nns	Multiple persons (non-NE)	Officials
group-ne	Organization (NE)	Congress
group	Group of people, place (non-NE)	crowd, hospital
country-ne	Country (NE)	Russia
country	Location (non-NE)	country
misc	Abstract concepts	program

4.2 Candidate Merging

The goal of the sub-task candidate merging is to find and merge candidates that refer to the same semantic concept. State-of-the-art methods, such as coreference resolution (see Section 2), cannot reliably resolve abstract and broadly defined coreferences as they occur in bias by WCL. Thus, we propose a merging method consisting of six steps (see Figure 1), where each step analyzes specific

characteristics of two candidates to determine whether the candidates should be merged. Merging steps 1 and 2 determine the similarity of two candidates as to their (core) meaning. Steps 3 and 4 focus on multi-word expressions. Steps 5 and 6 focus on frequently occurring words common in two candidates.

(1) Representative phrases' heads: we merge two candidates by determining the similarity of their core meaning (as a simplified example, we would merge "Donald Trump" and "President Trump"). *(2) Sets of phrases' heads*: we determine the similarity as to the meaning of all phrases of two candidates ({Trump, president} and {billionaire}). *(3) Representative labeling phrases:* similarity of adjectival labeling phrases. Labeling is an essential property in bias by WCL ("illegal immigrants" and "undocumented workers"). *(4) Compounds:* similarity of nouns bearing additional meaning to the heads ("DACA recipients" and "DACA applicants"). *(5) Representative wordsets:* similarity of frequently occurring words common in two candidates ("United States" and "U.S."). *(6) Representative frequent phrases:* similarity of longer multi-word expressions where the order is important for the meaning ("Deferred Action of Childhood Arrival" and "Childhood Arrivals").

For each merging step i, we define a 9×9 comparison matrix cmat_i spanned over the nine candidate types listed in Table 3. The normalized scalar in each cell $\mathrm{cmat}_{i,u,v}$ defines whether two candidates of types u and v are considered comparable (if $\mathrm{cmat}_{i,u,v} > 0$). As described later, for some merging steps, we also use $\mathrm{cmat}_{i,u,v}$ as a threshold, i.e., we merge two candidates with types u and v if the similarity of both candidates $\geq \mathrm{cmat}_{i,u,v}$. We found generally usable default values for the comparison matrices' cells and other parameters described in the following through experimenting and domain knowledge (see Section 7).

We organize candidates in a list sorted by their number of phrases, i.e., mentions in the texts; thus, larger candidates are at the top of the list. In each merging step, we compare the first, thus largest, candidate with the second candidate, then third, etc. If two candidates at comparison meet a specific similarity criterion, we merge the current (smaller) candidate into the first candidate, thereby removing the smaller candidate from the list. Once the pairwise comparison reaches the end of the list for the first candidate, we repeat the procedure for each remaining candidate in the list, e.g., we compare the second (then third, etc.) candidate pairwise with the remainder of the list. Once all candidates have been compared with another, we proceed with the next merging step.

In the first merging step, we merge two candidates if the *head* of each of their *representative phrase* (see Section 4.1) is identical by string comparison. By default, we apply the first merging step only to candidates of identical NE-based types, but one can configure the step's comparison table cmat_1 to be less restrictive, e.g., allow also other type comparison or inter-type comparisons.

Second, we merge two candidates if their *sets of phrases' heads* are semantically similar. For each candidate, we create a set H consisting of the heads from all phrases belonging to the candidate. We then vectorize each head within H into the word embeddings space of the enhanced word2vec model trained on the GoogleNews corpus (300M words, 300 dimensions) [35]. We then compute the mean vector $\overrightarrow{m_H}$ for the whole set of head vectors.

Then, to determine whether two candidates c_0 and c_1 are semantically similar, we compute their similarity $s(c_0, c_1) = \mathrm{cossim}(\overrightarrow{m_H}, \overrightarrow{n_H})$, where $\overrightarrow{m_H}$ is the mean head vector of c_0, $\overrightarrow{n_H}$ the mean head vector of c_1, and $\mathrm{cossim}(\dots)$ the cosine similarity function. We merge the candidates, if c_0 and c_1 are of the same type, e.g., each represents a person, and if their cosine similarity $\mathrm{s}(c_0, c_1) \geq t_{2,\mathrm{low}} = 0.5$. We also merge candidates that are of different types if we consider them comparable (defined in cmat_2), e.g., NEs such as "Trump" with proper nouns (NNP) such as "President," and if $\mathrm{s}(c_0, c_1) \geq t_{2,\mathrm{high}} = 0.7$. We use a higher, i.e., more restrictive, threshold since the candidates are not of the same type.

Third, we merge two candidates if their *representative labeling phrases* are semantically similar. First, we extract all *adjective NPs* from a candidate containing a noun and one or more labels, i.e., adjectives attributing to the noun. If the NP contains multiple labels, we extract for each label one NP, e.g., "young illegal immigrant" is extracted as "young immigrant" and "illegal immigrant." Then, we vectorize all NPs of a candidate and cluster them using affinity propagation [17]. To vectorize each NP, we concatenate its words, e.g., "illegal_worker" and look it up in the embeddings space produced by the enhanced word2vec model (see second merging step), where frequently occurring phrases were treated as separate words during training [25]. If the concatenated NP is not part of the model, we calculate a mean vector of the vectors of the NP's words. Each resulting cluster consists of NPs that are similar in meaning. For each cluster within one candidate, we select the single adjective NP with the global most frequent label, i.e., the label that is most frequent among all candidates. This way selected NPs are the *representative labeling phrases* of a candidate.

Then, to determine the similarity between two candidates c_0 and c_1 in the third merging step, we compute a similarity score matrix $S(V, W)$ spanned by the representative labeling phrases $v_i \in V$ of c_0 and $w_j \in W$ of c_1. We look up a type-to-type specific threshold $t_3 = \mathrm{cmat}_3[\mathrm{type}(c_0)][\mathrm{type}(c_1)]$, and $\mathrm{type}(c)$ returns the type of candidate c (see Table 3). For each cell $s_{i,j}$ in $S(V, W)$, we define a three-class similarity score:

$$s_{i,j} = \begin{cases} 2, \text{if } \mathrm{cossim}(\overrightarrow{v_i}, \overrightarrow{w_j}) \geq t_3 + t_{3,\mathrm{r}} \\ 1, \text{if } \mathrm{cossim}(\overrightarrow{v_i}, \overrightarrow{w_j}) \geq t_3 \\ 0, \text{else} \end{cases}$$

where $\mathrm{cossim}(\overrightarrow{v_i}, \overrightarrow{w_j})$ is the cosine similarity of both vectors, and $\mathrm{t}_{3,\mathrm{r}} = 0.2$ to reward more similar vectors into the highest similarity class. We found the three-class score to yield better results than using the cosine similarity directly. We merge c_0 and c_1 if $\mathrm{V} \sim \mathrm{W}$, i.e., $\mathrm{sim}(V, W) = \frac{\sum_{s \in S} s_{i,j}}{|V||W|} \geq t_{3,\mathrm{m}} = 0.3$. When merging candidates, we transitively merge different candidates U, V, W if $\mathrm{U} \sim \mathrm{W}$ and $\mathrm{V} \sim \mathrm{W}$, i.e., we say $\mathrm{U} \sim \mathrm{W}, \mathrm{V} \sim \mathrm{W} \xrightarrow{yields} U \sim V$, and merge both candidates U and W into V.

Fourth, we merge two candidates if they contain *compounds* that are semantically similar. In linguistics, a compound is a word or multi-word expression that consists of more than one stem, and that cannot be separated without changing its meaning [27]. We focus only on multi-word compounds, such as "DACA recipient."

First, we analyze the semantic similarity of the stems common in multiple candidates. Therefore, we first find all words that are

common in at least one compound of each candidate at comparison. In each candidate, we then select as its *compound phrases* all phrases that contain at least one of these words, and vectorize the compound phrases into the word embeddings space. Then, to determine the similarity of two candidates, we compute a similarity score matrix $S(V,W)$ spanned by all compound phrases $v_i \in V$ of candidate c_0 and $w_j \in W$ of c_1 using the same approach we used for the third merging step (including merging candidates that are transitively similar). If $\mathrm{sim}(V,W) \geq t_{4,m}$ we merge both candidates. Else, we proceed with the second merge method.

In the second method, we check for the lexical identity of specific stems in multiple candidates. Specifically, we merge two candidates c_0 and c_1 if there is at least one phrase in c_0 that contains a head that is a dependent in at least one phrase in c_1, and if both candidates are comparable according to cmat_4. For instance, two candidates are of type person-ne (see Table 3), and one phrase in c_0 has a headword "Donald," and one phrase in c_1 is "Donald Trump," where "Donald" is the dependent word.

Fifth, we merge two candidates if their *representative wordsets* are semantically similar. To create the representative wordset of a candidate, we perform the following steps. We create frequent itemsets of the words contained in the candidate's phrases excluding stopwords (we currently use an absolute support $\mathrm{supp} = 4$) and select all maximal frequent itemsets [1]. Note that this merging step thus ignores the order of the words within the phrases. To select the most representative wordsets from the maximal frequent itemsets, we introduce a representativeness score $\mathrm{r}(w) = \log(1 + \mathrm{l}(w)) * \log(\mathrm{f}(w))$, where w is the current itemset, $\mathrm{l}(w)$ the number of words in the itemset, and $\mathrm{f}(w)$ the frequency of the itemset in the current candidate. The representativeness score balances two factors: first, the *descriptiveness* of an itemset, i.e., the more words an itemset contains, the more comprehensively it describes its meaning. Second, the *importance*, i.e., the more often the itemset occurs in phrases of the candidate, the more relevant the itemset is. We then select as the *representative wordsets* the N itemsets with the highest representativeness score, where $N = \min 6, \mathrm{f_p}(c)$, where $\mathrm{f_p}(c)$ is the number of phrases in a candidate. If a word appears in more than $\mathrm{rs}_5 = 0.9$ of all phrases in a candidate but is not present in the maximal frequent itemsets, we select only $N - 1$ representative wordsets and add an itemset consisting only of that word to the representative wordsets. Lastly, we compute the mean vector $\vec{v}$ of each representative wordset v by vectorizing each word in the representative wordset using the word embeddings model introduced in the second merging step.

Then, to determine the similarity of two WCL candidates c_0 and c_1 in the fifth merging step, we compute a similarity score matrix $S(V,W)$ spanned by all representative wordsets $v_i \in V$ of candidate c_0 and $w_j \in W$ of c_1 analogously constructed as the matrix described in the third merging step. We merge c_0 and c_1, if $\mathrm{sim}(V,W) \geq t_5 = 0.3$.

Sixth, we merge two candidates if they have similar *representative frequent phrases*. To determine the most representative wordlists of a candidate, we conceptually follow the procedure from the fifth merging step but apply the steps to phrases instead of wordsets. Specifically, the representativeness score of a phrase o is $\mathrm{r}(o) = \log(1 + \mathrm{l}(o)) * \log(\mathrm{f}(o))$, where $\mathrm{l}(o)$ is the number of words in o, and $\mathrm{f}(o)$ the absolute frequency of o in the candidate. We then select as the *representative frequent phrases* the N phrases with the highest representative score, where $N = \min 6, \mathrm{f_p}(c)$.

Then, to determine the similarity of two candidates c_0 and c_1 in the sixth merging step, we compute a similarity score matrix $S(V,W)$ spanned by all representative wordlists $v_i \in V$ of candidate c_0 and $w_j \in W$ of c_1. We look up a type-to-type specific threshold $t_6 = \mathrm{cmat}_6[\mathrm{type}(c_0)][\mathrm{type}(c_1)]$. We compute the similarity score of each cell $s_{i,j}$ in $S(V,W)$:

$$s_{i,j} = \begin{cases} 2, \text{if } \mathrm{levend}(v_i, w_j) \leq t_6 - \mathrm{t}_{6,\mathrm{r}} \\ 1, \text{if } \mathrm{levend}(v_i, w_j) \leq t_6 \\ 0, \text{else} \end{cases}$$

where $\mathrm{levend}(v_i, w_j)$ is the normalized Levenshtein distance [26, 31] of both phrases, and $\mathrm{t}_{6,\mathrm{r}} = 0.2$. Then, over all rows j we find the maximum sum of similarity scores $\mathrm{sim_{hor}}$, and likewise $\mathrm{sim_{vert}}$ over all columns i:

$$\mathrm{sim_{hor}} = \max_{0 \leq i < |W|} \left(\sum_{j=0}^{|V|} s_{i,j} \right) / |W| \text{ and}$$

$$\mathrm{sim_{vert}} = \max_{0 \leq j < |V|} \left(\sum_{i=0}^{|W|} s_{i,j} \right) / |V|$$

We compute a similarity score for the matrix:

$$\mathrm{simval(V,W)} = \begin{cases} \mathrm{sim_{hor}}, \text{if } \mathrm{sim_{hor}} \geq \mathrm{sim_{vert}} \wedge |W| > 1 \\ \mathrm{sim_{vert}}, \text{else if } |V| > 1 \\ 0, \text{else} \end{cases}$$

Finally, we merge candidates c_0 and c_1 if $\mathrm{simval} \geq t_{6,m} = 0.5$.

5 FRAME IDENTIFICATION

Frame identification, the third task in the analysis pipeline (Figure 1), aims to identify frames present in the input set of news articles (contribution C3). Specifically, we look for frame devices, or technically frame property words, that express one or more frame properties on the WCL candidates identified during the second task, target concept analysis. Afterward, we aggregate the frame properties for each WCL candidate.

In a one-time process, we manually define a set of seed words for each of the frame properties $S_k \in S$ identified during the CA (see Table 2) [24]. For each frame property S_k, we gather seed words by selecting its top five synonyms from a dictionary [34], e.g., for the frame property "affection" we select the seed words: attachment, devotion, fondness, love, passion.

When the user inputs a set of news articles to the system, we perform the following procedure for each article. First, to identify frame property words, i.e., words that attribute to a frame, we first iterate all words in a news article and determine for each word its semantic similarity to each of the frame properties. Specifically, we compute the cosine similarity of the current word w and each seed word $s \in S_k$ of the current frame property S_k in the word embeddings space of the semantic network ConceptNet [53]. We define the semantic similarity $\mathrm{sim}(w, S_k) = \max_{s \in S_k} \mathrm{cossim}(\vec{w}, \vec{s})$. We assign to a word w any frame property S_k, where $\mathrm{sim}(w, S_k) > t_p = 0.4$. At the end of this procedure, each word has a set of weighted frame properties. The weight of a frame property on a word is defined by $\mathrm{sim}(w, S_k)$.

Second, for each WCL candidate c_i, we aggregate the frame properties $S_k \in S$ from all its modifiers w_j of c_i found by dependency parsing [6]. We use a set of manually devised rules to handle the different types of relations between head c_i and modifier w_j, e.g., to assign the frame properties of an attribute (modifier) to its noun (WCL candidate), or a predeterminer (modifier) to its head (WCL candidate).

6 EVALUATION AND DISCUSSION

We evaluate the effectiveness of the target concept analysis task in a quantitative evaluation (Section 6.1). To measure the effectiveness of the WCL candidate extraction, we compare the automatically extracted WCL candidates with the target concepts manually annotated in the CA. Furthermore, we demonstrate the effectiveness and usability as to the overall goal, i.e., finding and aggregating frame properties that an article expresses on each WCL candidate (Section 6.2). For both tasks, we also discuss the strengths and weaknesses of our approach, from which we derive future research directives, which we discuss in Section 7.

6.1 Target Concept Analysis

Table 4 shows that the performance of the target concept analysis evaluated on NewsWCL50 is significantly improved compared to the state-of-the-art. The overall precision p = 63.6%, recall r = 41.3%, and F1 score F1= 45.7%. The best performing baseline B3 achieves F1 = 29.8%. B2, a baseline that uses the closest related technique coreference resolution, achieves F1 = 22.6%.

To evaluate the performance of the target concept analysis, we compare automatically extracted and merged WCL candidates with manually coded target concepts, e.g., "USA/Donald Trump." We find for each target concept in NewsWCL50 the best matching WCL candidate [31], i.e., the candidate whose phrases yield the largest overlap to the mentions of the target concept. To account for the subjectivity of the coding task in the CA, particularly when coding abstract target concepts (see Section 3), we allow in our evaluation multiple candidates to be assigned to the same target concept. Unmatched candidates and unmatched target concepts account to false positives and false negatives, respectively.

We compare our approach with three baselines (Table 4). B1 randomly assigns each NP and each mention extracted by coreference resolution to a single target concept (F1 = 11.5%). B2 uses state-of-the-art coreference resolution [8, 9] to extract candidates (F1 = 22.6%). Specifically, B2 extracts each coreference chain as a single candidate and sets all mentions within the chain as the candidate's phrases. B3 extracts each NP and each mention of coreference chains as single candidates and clusters them in the word2vec space [25] using affinity propagation [17]. Each resulting cluster of phrases yields one candidate (F1 = 29.8%).

Table 4 shows that using merging steps 1 to 5 or 1 to 6 achieves the best performance (F1 = 45.7%) and that each of the merging steps improves the F1 performance (from 33.2% to 45.7%). We argue for using steps 1 to 6 since the sixth step is the only merging step capable of merging longer, order-conveying multi-word expressions. These, however, do not occur frequently enough in NewsWCL50 to significantly improve the F1 (also see Section 7).

Table 4: Performance of the target concept analysis and baseline approaches on all events of NewsWCL50. Best and worst performing approaches are highlighted.

Merging approach	P	R	F1
Baseline B1	12.7	13.4	**11.5**
B2	83.6	13.8	22.6
B3	55.4	23.5	**29.8**
Step 1	75.7	25.6	**33.2**
Steps 1 to 2	67.0	36.2	42.6
Steps 1 to 3	66.4	37.2	43.2
Steps 1 to 4	64.5	39.0	43.7
Steps 1 to 5	63.6	41.3	**45.7**
Steps 1 to 6	63.6	41.3	**45.7**

Table 5 shows that our approach also performs significantly better than each baseline for all candidate types. We find that our approach performs best on concepts that consist mainly of NPs and that are narrowly defined, e.g., we achieve $F1_{Actor} = 66.3\%$ on the type Actor, whose candidates are single persons.

Our approach performs worse on concepts that consist mainly of (1) VPs, or are (2) broadly defined or (3) abstract. However, also for those concept types, our approach performs significantly better than each baseline. Since our approach is currently not designed for VPs (see Sections 4 and 7), we achieve a lower $F1_{Action} = 30.4\%$ as expected on the Action type, whose candidates consist mainly of VPs. The concept type Actor-I is very broadly defined as to our codebook and has the lowest performance $F1_{Actor-I} = 26.6\%$: in the CA, different individua were subsumed under one Actor-I concept to save time (see Section 3), which we plan to improve (see Section 7).

Table 5: Performance on different concept types

Concept type	*Size*	$F1_{B1}$	$F1_{B2}$	$F1_{B3}$	P	R	F1
Action	*132*	10.8	9.5	31.3	43.5	26.5	**30.4**
Actor	*1473*	11.6	36.9	34.9	73.0	63.1	**66.3**
Actor-I	*554*	9.8	17.2	19.9	78.6	18.4	**26.6**
Country	*1626*	11.8	22.7	30.0	59.5	34.4	39.6
Event	*538*	10.4	20.3	36.0	51.6	47.6	46.8
Group	*545*	12.8	14.8	25.0	78.6	41.7	49.0
Misc	*612*	12.6	9.3	20.8	48.9	25.7	**29.1**
Object	*614*	10.9	18.2	33.3	54.6	41.0	45.6

The extraction of WCL candidates of the type Misc is as expected most challenging, since by definition its concepts are mostly abstract or complex. For example, the concept "Reaction to IRN deal" (event #9) contains both actual as well as possible, future (re)actions to the event (the "Iran deal"), and assessments and other statements by persons regarding the event. While the extraction performance is second lowest compared to the other concept types, our approach performs significantly better than the best performing baseline $F1_{B3,Misc} = 20.8\% \ll F1_{Misc} = 29.1\%$.

Table 6 shows the performance of our approach on the individual events of NewsWCL50. The approach performs best on events #0, #1, and #3 ($F1_0 = 58.3\%$) and worse on events #4, #9, and #7 ($F1_4 = 32.8\%$). We find that the target concepts in the CA

of events with high performance are mainly NPs, e.g., of concept type Actor. Events with lower performance contain more broadly defined concepts or Action concepts. This is in line with our findings regarding the performance of the individual concept types.

Table 6: Performance on individual events

	0	1	2	3	4	5	6	7	8	9
Size	*569*	*632*	*597*	*890*	*620*	*470*	*608*	*567*	*647*	*494*
P	69.9	68.1	48.8	82.5	52.0	60.7	68.2	58.3	66.7	48.1
R	57.5	46.4	46.1	43.4	29.2	46.8	37.2	29.8	43.7	32.0
F1	**58.3**	**53.2**	45.4	**51.7**	**32.8**	50.1	42.9	**35.8**	48.1	**34.9**

We also find that our approach is able to extract and merge unknown concepts, i.e., concepts that are not contained in the word embeddings space [25]. For example, when the GoogleNews corpus was published in 2013 [35], many concepts, such as "US President Trump" or "Denuclearization," did not exist yet or had a different, typically more general, meaning than in 2018. Yet, the approach was able to correctly merge phrases with similar meanings, e.g., in event #2, the target concept "Peace" contains among others "a long-term detente," "denuclearization," and "peace." In event #6, the approach was able to resolve, for example, "many immigrants," "the caravan," "the group marching toward the border," "families," "refugees," "asylum seekers," and "unauthorized immigrants." In event #1, the approach resolved, among others, "allegations," "the infamous Steele dossier," "the salacious dossier," and "unsubstantiated allegations."

In sum, the results of the evaluation show a significantly improved performance of our approach in finding and resolving phrases referring to the same concept in bias by WCL compared to state-of-the-art techniques, such as coreference resolution.

6.2 Frame Identification

We demonstrate and discuss the effectiveness and usability of our approach as to analyzing and finding frames in a set of news articles reporting on the same event in two use cases. In the first use case, we investigate the frame properties of WCL candidates in event #3, where the DNC, a part of the Democratic Party in the US sued Russia and associates of Trump's presidential campaign (see Table 1). Table 7 shows exemplary frame properties of the three main actors involved in the event: Donald Trump, the Democratic Party, and the Russian Federation; each being a different concept type (shown in parentheses in Table 7). The first column shows each WCL candidate's representative phrase (see Section 4.1). The linearly normalized scores $s(c, a, f)$ in the three exemplary frame property columns represent how strongly each article a (row) portrays a frame property f regarding a WCL candidate c: $s = 1$ or -1 indicates the maximum presence of the property or its antonym, respectively. A value of 0 indicates the absence of the property, or equal presence of the property and its antonym.

Left-wing outlets (LL and L) more strongly ascribe the property "aggressor" to Trump, e.g., $\mathrm{s}(\mathrm{Trump}, \mathrm{LL}, \mathrm{aggressor}) = 1$, than right-wing outlets, e.g., $\mathrm{s}(\mathrm{Tr.}, \mathrm{R}, \mathrm{aggr.}) = 0.34$, which is conformal with the findings of manual analyses of news coverage of left- vs. right-wing outlets regarding Republicans [12, 20, 21]. The Democratic Party is portrayed in all outlets as rather aggressive (s = [0.91,1]), which can be expected due to the nature of the event, since the DNC sued various political actors. Other frame properties, such as "reason," yield less clear patterns. We find that an increased level of abstractness is the main cause for lower frame identification performance (cf. [22, 23, 28, 46]). For example, in the CA (Section 3), we noticed that "reason" was often not induced by single words but rather more abstractly through actions that were assessed as reasonable by the human coders.

Table 7: Exemplary frame properties in the 1st use case

WCL candidate	Outlet	honor	aggressor	reason
Trump (Actor)	LL	-0.51	1.00	-0.32
	L	-0.75	0.76	0.00
	M	0.04	0.00	0.89
	R	0.00	0.34	0.00
	RR	0.00	0.44	0.00
Democratic Party (Group)	LL	0.40	1.00	0.00
	L	0.57	1.00	0.00
	M	-1.00	0.91	-0.87
	R	0.93	1.00	-0.37
	RR	-0.98	1.00	0.00
Russia (Country)	LL	1.00	0.53	0.00
	L	0.00	0.00	0.00
	M	0.00	-0.89	0.87
	R	1.00	0.03	0.00
	RR	-0.98	1.00	0.00

In the second use case, we analyze bias by WCL in event #8, where special counsel Mueller provided a list of questions to Trump. Table 8 shows selected frame properties of the two main actors involved in the event: Trump and Mueller, both WCL candidates of type Actor. Using our approach, we find that left-wing outlets ascribe more confidence and trustworthiness to Mueller, e.g., $\mathrm{s}(\mathrm{Mueller}, \mathrm{LL}, \mathrm{confidence}) = 0.7$, than right-wing outlets, e.g., $\mathrm{s}(\mathrm{Mu.}, \mathrm{RR}, \mathrm{confidence}) = 0$, while for Trump this is the converse, e.g., $\mathrm{s}(\mathrm{Trump}, \mathrm{LL}, \mathrm{conf.}) = -0.19$ vs. $\mathrm{s}(\mathrm{Tr.}, \mathrm{RR}, \mathrm{conf.}) = 1$. More strongly, left-wing news outlets even ascribe non-trustworthiness to Trump, e.g., $\mathrm{s}(\mathrm{Tr.}, \mathrm{LL}, \mathrm{trustworthiness}) = -0.93$.

Table 8: Exemplary frame properties in the 2nd use case

WCL candidate	Outlet	confidence	power	trustworth.
Mueller (Actor)	LL	0.70	0.00	1.00
	L	1.00	0.00	0.97
	M	0.61	0.00	0.63
	R	0.13	1.00	0.49
	RR	0.00	0.17	0.00
Trump (Actor)	LL	-0.19	0.41	-0.93
	L	-0.80	0.00	-1.00
	M	0.41	0.00	-0.19
	R	0.48	0.00	0.00
	RR	1.00	0.47	-0.16

Due to the difficulty of automatically estimating frames (cf. [22, 23, 28]), the identification of frame properties ascribed to WCL

candidates does not always yield clear or expected patterns, particularly for abstract or implicitly ascribed frame properties. For example, we could not find clear patterns for the frame properties "reason" in the first use case (Table 7) and "power" in the second use case (Table 8), which is mainly due to the abstractness used to portray a person as being powerful or reasonable.

In sum, in the qualitative evaluation, we find that the approach capably analyzes most frame properties that news articles portray on the contained WCL candidates, e.g., politicians and countries. The majority of automatically identified and aggregated frame properties, such as "honor," "aggressor," and "trustworthiness" are in line with both the publicly assumed slant of the analyzed news outlets as well as with findings from manual CAs conducted by social scientists (cf. [22, 23, 28]). Abstract or implicitly ascribed frame properties require further improvements (see Section 7).

7 FUTURE WORK

We plan to create a larger WCL dataset (contribution C1), from which the two other contributions may benefit. A larger dataset can be used for finding the optimal parameter configuration [3] in the target concept analysis (C2, Section 4) and frame identification (C3, Section 5), where we determined the parameter values using domain knowledge and through experiments. A larger dataset may also be used to devise and train machine learning methods to identify bias by WCL (C2 and C3), such as sequence labeling [42]. Lastly, we plan to create a bias by WCL dictionary (cf. [19, 29, 55]) by extracting common phrases for each frame property (C3).

Future CAs to create WCL datasets will require less effort than the creation of NewsWCL50, since our codebook can be reused [22]. Before creating a larger WCL dataset, we plan to implement and validate minor improvements in the codebook, e.g., infrequent individua are currently coded jointly into a single "[Actor]-I" target concept. While such coding requires less coding effort, it also negatively skews the measured evaluation performance (see Section 6.1). An idea is to either not code infrequent target concepts, or code them as single concepts. Furthermore, we plan to add emotions to the list of frame properties (cf. [24]).

To improve the target concept analysis (C2), we plan to devise an extraction approach for VPs and investigate how to merge semantically coreferential Action candidates. We think that conceptually most of the current merging steps require only minor adaptions to analyze the semantic similarity of VP-based candidates.

While the estimation of abstract or implicitly described frame properties (C3) is beyond the capabilities of current NLP (cf. [22]), we plan the investigate the use of bias dictionaries to generally improve the estimation performance [24]. Promising dictionaries are, for example, SÉANCE [10], Empath [15], General Inquirer [52], and bias-inducing phrases (cf. [2, 48]). Afterward, we will perform a quantitative evaluation of our approach by comparing the estimated frame properties with those from NewsWCL50.

To make bias by WCL understandable and accessible to regular news readers, we plan to devise a frame-based clustering and visualizations (see Figure 1). From the input articles, the clustering needs to group articles that similarly frame the actors and other concepts involved in the reported event. Visualizations that show most contrasting WCL phrases within coverage on the same event [23], effectively and efficiently reveal bias by WCL (cf. [23, 46]).

8 CONCLUSION

The three main contributions of this paper are: the first openly available dataset for the evaluation of automated methods to identify instances of bias by word choice and labeling (WCL). Second, the first approach that capably finds and resolves coreferences as they occur in bias by WCL (F1=45.7%), going well beyond the capabilities of coreference resolution (F1=22.6%) and other state-of-the-art techniques (best baseline: F1=29.8%): our approach resolves cross-document coreferences for also broadly defined and abstract semantic concepts, such as "Reactions to …" and "Denuclearization." Third, we demonstrate the usability of our approach as to analyzing bias by WCL using a prototype that estimates the effects of words modifying the WCL coreferences. While the results are mixed for implicitly described frame properties, such as being "reasonable," we find many results being conformal with bias studies conducted by social scientists. We think that the system presents a first step towards our goal of enabling news readers to become aware of bias by WCL in their daily news consumption. NewsWCL50 and its codebook are available under a Creative Commons Attribution Share Alike 4.0 International license at: **https://github.com/fhamborg/NewsWCL50**

ACKNOWLEDGMENTS

This work was supported by the Carl Zeiss Foundation and the Zukunftskolleg program of the University of Konstanz. We thank Christina Zuber and Karsten Donnay for their support on the creation of NewsWCL50.

REFERENCES

[1] Aggarwal, C.C. and Han, J. 2014. *Frequent pattern mining*.
[2] Baumer, E.P.S. et al. 2015. Testing and Comparing Computational Approaches for Identifying the Language of Framing in Political News. *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL* (Denver, Colorado, USA, 2015), 1472–1482.
[3] Bishop, C. *Pattern Recognition and Machine Learning*. Springer-Verlag New York.
[4] Blei. David M. et al. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*. (2003). DOI:https://doi.org/10.1162/jmlr.2003.3.4-5.993.
[5] Chang, A.X. et al. 2016. A comparison of Named-Entity Disambiguation and Word Sense Disambiguation. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* (Portoroz, Slovenia, 2016), 860–867.
[6] Chen, D. and Manning, C. 2014. A Fast and Accurate Dependency Parser using Neural Networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014).
[7] Chong, D. and Druckman, J.N. 2007. Framing Theory. *Annual Review of Political Science*. 10, 1 (2007), 103–126. DOI:https://doi.org/10.1146/annurev.polisci.10.072805.103054.
[8] Clark, K. and Manning, C.D. 2016. Deep Reinforcement Learning for Mention-Ranking Coreference Models. *Empirical Methods on Natural Language Processing* (2016).
[9] Clark, K. and Manning, C.D. 2016. Improving coreference resolution by learning entity-level distributed representations. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Berlin, Germany, 2016), 643–653.
[10] Crossley, S.A. et al. 2017. Sentiment Analysis and Social Cognition

Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis. *Behavior Research Methods*. 49, 3 (2017), 803–821. DOI:https://doi.org/10.3758/s13428-016-0743-z.

[11] Data for PAN at SemEval 2019 Task 4: Hyperpartisan News Detection: 2018. *https://doi.org/10.5281/zenodo.1489920*. Accessed: 2019-04-10.

[12] DellaVigna, S. and Kaplan, E. 2007. The Fox News effect: Media bias and voting. *The Quarterly Journal of Economics*. 122, 3 (2007), 1187–1234. DOI:https://doi.org/10.3386/w12169.

[13] Entman, R.M. 1993. Framing: Toward clarification of a fractured paradigm. *Journal of communication*. 43, 4 (1993), 51–58.

[14] Entman, R.M. 2007. Framing bias: Media in the distribution of power. *Journal of communication*. 57, 1 (2007), 163–173.

[15] Fast, E. et al. 2016. Empath: Understanding Topic Signals in Large-Scale Text. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI'16)* (San Jose, California, USA, 2016), 4647–4657.

[16] Finkel, J.R. et al. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of the 43rd annual meeting on association for computational linguistics* (2005), 363–370.

[17] Frey, B.J. and Dueck, D. 2007. Clustering by passing messages between data points. *Science*. 315, 5814 (2007), 972–976. DOI:https://doi.org/10.1126/science.1136800.

[18] Garner, R. et al. 2012. *Introduction to politics*. Oxford University Press.

[19] Grefenstette, G. et al. 2004. Coupling Niche Browsers and Affect Analysis for an Opinion Mining Application. *In Proceedings of the 12th International Conference Recherche d'Information Assistee par Ordinateur*. (2004), 186–194.

[20] Grefenstette, G. et al. 2004. Coupling Niche Browsers and Affect Analysis for an Opinion Mining Application. *In Proceedings of the 12th International Conference Recherche d'Information Assistee par Ordinateur*. (2004).

[21] Groseclose, T. and Milyo, J. 2005. A measure of media bias. *The Quarterly Journal of Economics*. 120, 4 (2005), 1191–1237. DOI:https://doi.org/10.1162/003355305775097542.

[22] Hamborg, F. et al. 2018. Automated identification of media bias in news articles: an interdisciplinary literature review. *International Journal on Digital Libraries*. (2018), 1–25. DOI:https://doi.org/10.1007/s00799-018-0261-y.

[23] Hamborg, F. et al. 2018. Bias-aware news analysis using matrix-based news aggregation. *International Journal on Digital Libraries*. (2018). DOI:https://doi.org/10.1007/s00799-018-0239-9.

[24] Hamborg, F. et al. 2019. Illegal Aliens or Undocumented Immigrants? Towards the Automated Identification of Bias by Word Choice and Labeling. *Proceedings of the iConference 2019* (Washington, DC, USA, 2019), 1–10.

[25] Le, Q. and Mikolov, T. 2014. Distributed Representations of Sentences and Documents. *International Conference on Machine Learning - ICML 2014*. 32, (2014). DOI:https://doi.org/10.1145/2740908.2742760.

[26] Levenshtein, V.I. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*. (1966). DOI:https://doi.org/citeulike-article-id:311174.

[27] Lieber, R. and Štekauer, P. 2012. *The Oxford Handbook of Compounding*.

[28] Lim, S. et al. 2018. Towards Bias Inducing Word Detection by Linguistic Cue Analysis in News Articles. (2018).

[29] Lima, S. et al. Understanding Characteristics of Biased Sentences in News Articles. *Proceedings of the 6th International Workshop on News Recommendation and Analytics (INRA 2018)*.

[30] Liu, B. 2012. Sentiment Analysis and Opinion Mining. May (2012), 1–108. DOI:https://doi.org/10.2200/S00416ED1V01Y201204HLT016.

[31] Manning, C.D. et al. 2009. An Introduction to Information Retrieval. *Online*. (2009). DOI:https://doi.org/10.1109/LPT.2009.2020494.

[32] Manning, C.D. et al. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. (2014), 55–60. DOI:https://doi.org/10.3115/v1/P14-5010.

[33] McCarthy, J. et al. 2008. Assessing Stability in the Patterns of Selection Bias in Newspaper Coverage of Protest During the Transition from Communism in Belarus. *Mobilization: An International Quarterly*. 13, 2 (2008), 127–146.

[34] Merriam-Webster Inc. 2016. *The Merriam-Webster Dictionary New Edition*.

[35] Mikolov, T. et al. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*. (2013). DOI:https://doi.org/10.1162/153244303322533223.

[36] Miller, G.A. 1995. WordNet: a lexical database for English. *Communications of the ACM*. 38, 11 (1995), 39–41. DOI:https://doi.org/10.1145/219717.219748.

[37] Miller, J. 2011. *A Critical Introduction to Syntax*. Continuum International Publishing Group.

[38] Mitchell, A. et al. 2014. *Political Polarization and Media Habits - From Fox News to Facebook, How Liberals and Conservatives Keep Up with Politics*.

[39] Munson, S.A. et al. 2009. Sidelines: An Algorithm for Increasing Diversity in News and Opinion Aggregators. *ICWSM* (2009).

[40] Nadeau, D. and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*. 30, 1 (2007), 3–26.

[41] Neuendorf, K.A. 2016. *The content analysis guidebook*. Sage Publications.

[42] Nguyen, N. and Guo, Y. 2007. Comparisons of Sequence Labeling Algorithms and Extensions. *Proceedings of the 24th International Conference on Machine Learning* (Corvalis, Oregon, USA, 2007), 681–688.

[43] Niven, D. 2002. *Tilt?: The search for media bias*. Greenwood Publishing Group.

[44] Oliver, P.E. and Maney, G.M. 2000. Political Processes and Local Newspaper Coverage of Protest Events: From Selection Bias to Triadic Interactions. *American Journal of Sociology*. 106, 2 (2000), 463–505.

[45] Papacharissi, Z. and de Fatima Oliveira, M. 2008. News Frames Terrorism: A Comparative Analysis of Frames Employed in Terrorism Coverage in U.S. and U.K. Newspapers. *The International Journal of Press/Politics*. 13, 1 (2008), 52–74. DOI:https://doi.org/10.1177/1940161207312676.

[46] Park, S. et al. 2009. NewsCube: Delivering Multiple Aspects of News to Mitigate Media Bias. *Proceedings of CHI '09, the SIGCHI Conference on Human Factors in Computing Systems*. (2009), 443–453. DOI:https://doi.org/10.1145/1518701.1518772.

[47] Price, V. et al. 2005. Framing public discussion of gay civil unions. *Public Opinion Quarterly*.

[48] Recasens, M. et al. 2013. Linguistic Models for Analyzing and Detecting Biased Language. *Proceedings of the 51st Annual Meeting on Association for Computational Linguistics*. (2013), 1650–1659.

[49] Research Guides: "Fake News," Lies and Propaganda: How to Sort Fact from Fiction: Where do news sources fall on the political bias spectrum? *http://guides.lib.umich.edu/c.php?g=637508&p=4462444*. Accessed: 2018-12-03.

[50] Schreier, M. 2012. *Qualitative content analysis in practice*. Sage publications.

[51] Schuldt, J.P. et al. 2011. "Global warming" or "climate change"? Whether the planet is warming depends on question wording. *Public Opinion Quarterly*. (2011). DOI:https://doi.org/10.1093/poq/nfq073.

[52] Smith, M.S. et al. 1967. The General Inquirer: A Computer Approach to Content Analysis. *American Sociological Review*.

[53] Speer, R. and Havasi, C. 2012. Representing General Relational Knowledge in ConceptNet 5. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (Istanbul, Turkey, 2012).

[54] Stanford CoreNLP Javadoc: 2018. *https://nlp.stanford.edu/nlp/javadoc/javanlp/*. Accessed: 2018-12-19.

[55] Subasic, P. and Huettner, A. 2001. Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems*. 9, 4 (2001), 483–496. DOI:https://doi.org/10.1109/91.940962.

[56] White, D.M. 1950. The "Gate Keeper": A Case Study in the Selection of News. *Journalism Bulletin*. 27, 4 (1950), 383–390. DOI:https://doi.org/10.1177/107769905002700403.