Who is the Hero, the Villain, and the Victim? Detection of Roles in News Articles using Natural Language Techniques

Diego Gomez-Zara
Northwestern University
Intelligent Information Lab
Evanston, Illinois, USA
dgomezara@u.northwestern.edu

Miriam Boon
Northwestern University
Intelligent Information Lab
Evanston, Illinois, USA
MiriamBoon2012@u.northwestern.edu

Larry Birnbaum
Northwestern University
Intelligent Information Lab
Evanston, Illinois, USA
l-birnbaum@northwestern.edu

ABSTRACT
News articles often use narrative frames to present people, organizations, and facts. These narrative frames follow cultural archetypes, enabling readers to associate each of the presented elements with familiar stereotypes, well-known characters, and recognizable outcomes. In this way, authors can cast real people or organizations as heroes, villains, or victims. We present a system that identifies the main entities of a news article, and determines which is being cast as a hero, a villain, or a victim. As currently implemented, this system interacts directly with news consumers through a browser extension. Our hope is that by informing readers when an entity is cast in one of these roles, we can make implicit bias explicit, and thereby assist readers in applying their media literacy skills. This approach can also be used to identify roles in well-understood event sequences in a more prosaic manner, e.g., for information extraction.

Author Keywords
Computational journalism; entity recognition; information extraction; role detection; contextual information; sentiment analysis.

ACM Classification Keywords
• **Computing methodologies~Natural language processing** • *Human-centered computing~Natural language interfaces*

INTRODUCTION
A frequent problem in understanding news lies in comprehending the broader context of the particular news story or other informational content at hand. One important mechanism that the media may use to shape public opinion is by *framing* people, events, and issues in particular ways. By using common narrative structures and cultural references, framing enables a communication source to present and define an issue within a "field of meaning" [5]. Framing helps to contextualize events by connecting with readers' prior knowledge, cultural narratives, and moral values [8]. It can present the agents involved in a story as heroes, villains, or victims, so that readers can more easily anticipate and comprehend the attitudes, beliefs, decisions, and actions of these agents as characters. Narrative frames are widespread throughout media, be it films, literature, or news. Their use reflects pervasive cultural models for structuring the presentation of political discourse and identities [10, 12]. As a key element of popular culture storytelling, they use emotionality to unambiguously distinguish between good and evil through clear depictions of victimization, heroism, and villainy [2]. Frames tend to use positive terms to describe heroes, and negative terms for victims and villains [7, 13]. Culturally, heroes embody courage, outstanding achievements, or other noble qualities, whereas villains embody evil intentions, plotting, and other negative qualities. In sum, narrative frames are critical to how readers understand new situations in terms of previous ones, and thereby make sense of the causes, events, and consequences reported by news articles [4].

We present a system that detects how the main characters of a news article are framed as heroes, victims, or villains. As currently implemented, this system interacts with news consumers directly through a browser extension. This approach can also be used to identify roles in well-understood event sequences in a more prosaic manner, e.g., for information extraction.

A MODEL FOR ROLE DETECTION
A standard approach to recognizing narrative frames is to analyze the semantic relations among the different entities in the narrative, and with respect to the events it describes [1, 11]. However, determining these relations—understanding the events in a narrative, and the roles that the entities in that narrative play in those events—is a complex, difficult, and indeed, unsolved, computational challenge. Moreover, to the extent that we do understand how to address this challenge, the techniques involved can be quite expensive computationally. Using standard coreference resolution packages, for example, our system can take more than 20 seconds to analyze a single sentence. Users simply will not wait that long to see the classification results for an entire news article.

IUI'18, March 7–11, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-4945-1/18/03...$15.00
https://doi.org/10.1145/3172944.3172993

We propose a different approach: Rather than determining the specific events and event types described in the narrative, and the semantic relations among the entities involved in those events, in detail, we suggest analyzing the entities at a much higher level of abstraction—specifically, in terms of whether they possess the qualities of heroes, victims, or villains, as conveyed by the terms used to describe them. This approach suggests a relatively simple computational realization, in which the terms closest to each entity are assessed with respect to their similarity to the terms typically used in connection with heroes, villains, or victims. News articles, in particular, facilitate this approach given their use of consistent styles. We provide a simple, computationally cheap, and reasonably (although not perfectly) accurate role classifier.

More specifically, we seek to classify entities involved in narratives into the semantic / thematic categories "heroes," "victims," and "villains"—categories which are highly abstract and thus broadly applicable, and yet are also likely to be comprehensible to typical users. To accomplish this, we have developed a framework and platform for role detection that comprises the following core procedures:

1. Recognize and rank the main entities present in the news article based on prominence.
2. Identify the terms closest to these entities.
3. Use sentiment analysis to determine the polarity of these terms.
4. Calculate the similarity of these terms to those typically used to describe heroes, villains, and victims, using standard term sets or dictionaries.
5. Classify the main entities, as determined in step 1, according to the proposed semantic categories, using the analyses provided by steps 3 and 4.

Entity recognition

The system starts by identifying the main characters involved in a news story. Taking the news article's headline and text as inputs, it analyzes them for relevant entities using standard (and actually somewhat basic) entity recognition methods, for example those provided by the Natural Language Toolkit (NLTK) [3]. For each entity, the system saves the relevant terms and relations and tracks all the possible name forms to establish their connections later. For example, a person could be called "John Smith" and also be mentioned in the article as "Mr. Smith," "Smith," or just "John." The system tracks all these references, and reduces and merges them in a later step. Entities may be a *Person*, an *Organization*, *Geopolitical Entity* (GPE), or only a *Position* (such as "the witness" or "the police officer"). Since news articles have consistent narrative styles, we used standard grammatical forms to detect entities and their relations. The relations associated with *Persons* and *Organizations* are demonyms (e.g., "The *French* president…"), countries (e.g., "The president of *Spain*…"), organizations (e.g., "The CEO of *Google*…"), and positions (e.g., "*Judge* Mark…"). We assume that heroes, villains, and villains are *Persons*, *Organizations*, *GPE*, or *Positions* identified in this way.

Because of their significance, the system uses the headline first to detect the main characters of a story. Headlines present an optimal context for interpretation since the heroes, victims, or villains are often highlighted [6]. The system next analyzes the entities in the text, sentence by sentence, saving their locations in a list using the indices of their corresponding sentences. During this process, the system establishes the connection between the headline entities and those mentioned in the news article's body. This generates a list of several entities with their respective name forms and locations in the text.

One difficulty in applying entity detection in this context is that news articles often involve previously unreferenced names that *a priori* are novel for knowledge databases (e.g., the name of a victim) as well as using generic words to refer to these entities (e.g., "A *woman* was killed…"). To address this challenge, entity recognition incorporates a search for proper nouns as well as generic terms (i.e., "man," "woman," "boy," etc.). All these entities are classified as *Person*. According to the gender and role detected, the system then associates the most frequently mentioned entity in the text with the most relevant generic word found in the text. For example, if "Mrs. Smith" is the most mentioned "victim" in the text, and "*woman*" is often used to refer a victim, we merge these entities.

Entity cut-off

Once the system recognizes all the entities, it must detect and merge the multiple references to a given entity and select the most relevant entities in the article for role assignment. First, the system merges different references to the same entity and selects the longest entity that has proper nouns as the canonical form (e.g., the best results for "Mr. Smith," "John Smith," and "Smith," would be "John Smith"). During this process, the system counts the number of times that each entity is mentioned in the text. It then calculates the relevance score of each merged entity as:

$$r_i = \alpha h_i + \frac{n_i}{\|s\| f_i}$$

where r_i is the relevance score of the merged entity i, h_i is a dummy variable where 1 means that the merged entity was mentioned in the headline (0 otherwise), n_i is the number of times that the merged entity is mentioned in the body of the text, $\|s\|$ is the total number of sentences in the text, and f_i is the first location where the merged entity is mentioned in the text. In other words, we assume that entities that (a) appear in the headline, (b) are frequently mentioned in the article, and (c) are located near the beginning of the text are more relevant. The coefficient α is to weight the impact of a mention in the news article's headline.

Finally, the system takes all the entities found in the headline, and the three entities from the body with the highest relevance score, and assesses the role of each. We use both sets to ensure that all entities in the headline are included; and to compare the results as described below.

Role dictionaries

To determine the roles that these entities play in the narrative frame, the system checks the terms related to each entity and calculates their similarity with terms that typically describe heroes, villains, and victims. For each role, we created a term set or dictionary D_j, where j belongs to the set {"hero," "villain," "victim"}. The terms in each set were manually selected from news articles and blogs, as well as synonyms and antonyms. In total, each role dictionary comprises approximately 200 words.

Our intention with this step is to calculate how similar the terms related to each entity are to each role's dictionary. The system calculates the similarity of each term-pair by using WordNet, a lexical database of English [9] (also contained in NLTK). Specifically, we use the built-in methods for calculating Wu-Palmer Similarity [14]. This method returns a score denoting how similar two words are, based on the depth of the two senses in the taxonomy and that of their most specific ancestor node.

Sentiment analysis

One problem this approach faces is the similarity of opposite words that are closely related. For example, using Wu-Palmer, "*love*" is more similar to "*hate*" than to "*romance*." Since archetypal characters are frequently assessed in emotional terms—positive for heroes and negative for victims and villains—the system leverages this property by filtering the terms used for the similarity comparisons based on sentiment. The program calculates and saves the sentiment score of the terms closest to the relevant entities. If the term is positive or neutral, the system calculates its similarity with the hero dictionary. Equally, if the analyzed word is negative or neutral, the system checks its similarity to the victim and villain dictionaries. We calculate sentiment for each term using standard toolkits, in the current implementation *TextBlob*[1], yielding a polarity for each term, ranging from -1.0 (very negative) to 1.0 (very positive).

Role detection

We assume that the terms used to describe entities are usually closer to their respective characters in the text. Based on this assumption, the system calculates for each relevant entity its distance to other terms present in the sentence, and then prioritizes the similarity scores of the closest terms. More specifically, we define $S_i \in \{1, 2, \dots, K\}$ as the set of K sentences where the entity i is mentioned. Each element of S_k has a list of terms $\{w_{k,1}, w_{k,2}, \dots, w_{k,n}\}$, where n is the total length of the k^{th} sentence. The system calculates the similarity of a term w with respect to a role j by calculating the average of the similarities between that term w and each term d from the dictionary D_j, as follows:

[1] http://textblob.readthedocs.io/en/dev/

$$sim(w, j) = \sum_{d \in D_j} \frac{s(w, d)}{\|D_j\|}$$

To calculate the similarity that entity i has with role j in sentence k, we sum the similarity scores for each term in the sentence. To implement our proximity assumption, we apply a decay factor f according to the distance between the entity i and each term analyzed. Additionally, to take into account active vs. passive roles—heroes and villains are assumed to be more active, victims more passive—the system analyzes whether the entity i is the subject of the sentence (i.e., the entity that performs the action) or the object of the sentence (i.e., the entity that receives the action). In the first case, the words identified in the hero and villain dictionaries receive an additional score. If the entity is the object of the sentence, the words related to the victim dictionary receive an additional score. In summary, we calculate the score of the merged entity i for role j in sentence k as:

$$ro_{i,j,k} = \sum_{w \in S_k} \big(sim(w, j) + a(i, j, w)\big)(1 - f)^{d(i,w)}$$

where f is the decay factor between 0 and 1, $d(i, w)$ is the distance between the entity i and the term w of the sentence k, and $a(i, j, w)$ is the additional score that the role j receives according to the positions of the term w and the entity i in the sentence k. Finally, the system calculates the overall role scores for each entity by averaging all the scores computed using sentences in which they occur. The results are role scores for each relevant entity of the news article's headline and body.

The final step is to assign the hero, villain, and victim of the story, if any of these roles exist. To establish a baseline, we use the headline as a reference to compare the role scores assigned by the system. First, the system places the headline's entities in descending order according to their assigned role scores. Next, the system takes the highest value achieved by any entity on these three roles and classifies the entity according to that role. The system continues this selection process by picking the next highest role score from the remaining entities and classifies the next entity according to that role, and so on. Finally, the system performs the same ranking procedure for the relevant entities from the body. As a result, the system establishes a role classification of the headline's entities, and of the body's entities, which are then compared. If they differ, the role assignment based on the text is preferred.

INTERFACE AND EXAMPLES

Using the method described above, we have developed a Google Chrome extension that determines which entities are the hero, villain, or victim, for any article being displayed. To better contextualize these results, the system also presents the most relevant terms that explain each classification, based on similarity with the assigned role's dictionary and proximity to the respective entity.

Figure 1. The system detects President Trump as villain, President Macron as the hero, and France as the victim.

Figure 2. President Trump is detected as the hero, his friend Jim as the victim, and Paris as the villain.

We will illustrate the system's operation using two examples that show how the tool distinguishes the coverage by two different news organizations of the same news event. Figure 1 shows the system's results given an article from The New York Times entitled "Emmanuel Macron to welcome Trump, an unlikely partner, to France" (July 12, 2017). In his visit to France, the NYT focused on President Trump's relationship with the France's president Emmanuel Macron. Based on the article's body, the story's results indicate "Macron" as the hero, "Trump" as the villain, and "France" as the victim. On the other hand, Fox News covered this same visit focusing in Trump's friend Jim, who did not attend because—according to the president—Paris has been infiltrated by foreign extremists. This article was published the same day and entitled "Trump visits Paris without his friend, the mysterious Jim" (Figure 2). Initially, the system assigns Paris as the victim because it is included in the subject of the headline. Ultimately, however, it classifies Paris as the villain because is associated with the words "ruined," "terrorism," and "infiltrated." Additionally, the system classifies President Trump as the hero and his friend "Jim" as the victim of this story.

CONCLUSIONS

Our current implementation leverages standard entity recognition, sentiment analysis, and the use of term sets or dictionaries, which as a starting point have yielded promising results. During development, we iteratively improved our original results by incorporating a variety of disambiguation techniques and putting more weight on headlines. However, because it utilizes simple proximity, this approach cannot handle situations in which the article presents complex grammar structures such as use of passive voice. Moreover, the system clearly requires more testing to establish a reasonable threshold for assigning the roles: Sometimes, articles do not present clear heroes, victims, or villains, suggesting that a cut-off score is needed.

Moving forward, it will be critical to conduct a serious evaluation aimed at proving and quantifying the performance of the role detection process described here. We must develop performance indicators based on either user evaluation or a comparison with an annotated ground truth. As we conduct such an evaluation, and develop data that can be used to make reasoned choices, we expect to incorporate more up-to-date packages and technologies to improve the system's performance.

We have described an architecture for abstract or thematic role detection in news articles, instantiated in a browser extension. Rather than identifying entities in narratives as serving event- or topic-specific roles, such as buyer (or seller), inventor, thief, author, etc., we aim to identify them as serving highly abstract narrative or thematic roles, as either "hero," "villain," or "victim." From a computational perspective, we believe that these abstract thematic roles may be significantly easier to identify than topic-specific roles involving more fine-grained semantic analysis. Moreover, in many instances we believe that abstract role identification might be useful in determining these more specific roles. For example, if a story is about a kidnapping, and we identify an entity in that story as the villain, then he or she is probably the kidnapper.

We expect to release the tool and the code shortly for use by others to develop and perform experiments related to the consumption and diffusion of news articles. From a user perspective, we believe that these role identifications can ultimately help reveal bias in news articles and improve "media literate" information consumption. For example, this tool could be used to highlight and explain the differences in framing of a specific news event covered by different media sources. Similarly, it could be used to study polarization in media. Finally, since the system is able to identify the emotional words related to the main characters of an article, we may be able to use its results to modify the article to present a more neutral version of events.

ACKNOWLEDGMENTS

Our work at the Intelligent Information Lab (InfoLab) at Northwestern University has been supported in part by Google. Miriam Boon has also been supported by Northwestern University's program in Technology and Social Behavior. We thank Jason Cohn and Victor Bursztyn for many useful discussions.

REFERENCES

1. Gabor Angeli, Julie Tibshirani, Jean Wu and Christopher D Manning. 2014. Combining Distant and Partial Supervision for Relation Extraction. in *EMNLP*, 1556-1567.
2. Elisabeth Anker. 2005. Villains, victims and heroes: Melodrama, media, and September 11. *Journal of Communication*, 55 (1). 22-37.
3. Steven Bird, Ewan Klein and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".
4. Miriam L Boon. 2017. Augmenting Media Literacy with Automatic Characterization of News along Pragmatic Dimensions. in *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ACM, 49-52.
5. Claes H De Vreese. 2005. News framing: Theory and typology. *Information Design Journal & Document Design*, 13 (1).
6. Daniel Dor. 2003. On newspaper headlines as relevance optimizers. *Journal of Pragmatics*, 35 (5). 695-721. https://doi.org/10.1016/S0378-2166(02)00134-0
7. Jeff Goodwin, James M Jasper and Francesca Polletta. 2009. *Passionate politics: Emotions and social movements*. University of Chicago Press.
8. Melanie C Green. 2004. Transportation into narrative worlds: The role of prior knowledge and perceived realism. *Discourse processes*, 38 (2). 247-266.
9. George A. Miller. 1995. WordNet: a lexical database for English. *Commun. ACM*, 38 (11). 39-41. 10.1145/219717.219748
10. Zhongdang Pan and Gerald M Kosicki. 1993. Framing analysis: An approach to news discourse. *Political communication*, 10 (1). 55-75.
11. Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 492-501.
12. Dietram A Scheufele. 1999. Framing as a theory of media effects. *Journal of communication*, 49 (1). 103-122.
13. Stefan Stieglitz and Linh Dang-Xuan. 2013. Emotions and Information Diffusion in Social Media-Sentiment of Microblogs and Sharing Behavior. *Journal of Management Information Systems*, 29 (4). 217-248. 10.2753/MIS0742-1222290408
14. Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 133-138.