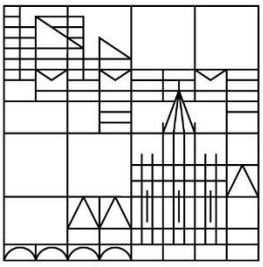
Master Thesis

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

submitted by
Anastasia Zhukova

at the

Universität Konstanz



Department of Computer and Information Science

1. Reviewer: Prof. Dr. Bela Gipp
2. Reviewer: Prof. Dr. Karsten Donnay

Konstanz, February 2019

Abstract

The term media bias denotes the differences of the news coverage about the same event. Slanted news coverage occurs when journalists frame the information favorably, i.e., they report with different word choice about the same concept, thus leading to the readers' distorted information perception. A word choice and labeling (WCL) analysis system was implemented to reveal biased language in news articles. In the area of Artificial Intelligence (AI), the WCL analysis system imitates well-established methodologies of content and framing analyses employed by the social sciences. The central thesis contribution is a development and implementation of the multi-step merging approach (MSMA) that unlike state-of-the-art natural language preprocessing (NLP) techniques, e.g., coreference resolution, identifies coreferential phrases of a broader sense, e.g., "undocumented immigrants" and "illegal aliens." An evaluation of the approach on the extended NewsWCL50 dataset was made achieving the performance of $F1 = 0.84$, which is twice higher than a best performing baseline. Finally, to enable visual exploration of the identified entities, a four-visualization usability prototype was proposed and implemented, which enables exploring entity composition of the analyzed news articles and phrasing diversity of the identified entities.

Acknowledgments

I would like to express my sincere gratitude to various people for their contribution to this thesis. Without their substantial support, this thesis would not be possible.

First and foremost, I am extremely grateful to my supervisor Felix Hamborg, who has encouraged me to work on this topic and provided invaluable guidance throughout all time of working on this thesis. I deeply appreciate Felix's constructive critics and feedbacks, the warm encouragement that motivated me to advance in this research. Our discussions have greatly helped me to reconsider the accounted problems from different perspectives. I have greatly benefited from this longtime support during the work on my project and thesis.

I also wish to thank Prof. Dr. Bela Gipp for supporting me in this research and offering me a once in a lifetime opportunity to partially work on my thesis at the National Institute of Informatics (NII) in Tokyo.

I gratefully acknowledge the contribution of my hosting professor at NII Prof. Dr. Yusuke Miyao and my colleagues at the Natural Language Processing and Computational Linguistics laboratory. I have clearly benefited from our weekly seminars learning both from the colleague's projects and the feedback provided on my own project. Discussions with Juan Ignacio Navarro Horñiacek, Joan Ginés i Ametllé, and Namgi Han were especially insightful for the experiment setup and evaluation design.

Many thanks to Oliver Sampson, who supervised me during the work as a research assistant and helped to advance my programming and analytical skills. I very much appreciate Norman Meuschke whose valuable advice helped to improve my academic writing greatly. I would like to acknowledge the help of Donald Werve and Julia Kelly-Neumann for helping me proofread my thesis. Thanks also to Peter Burger, who has provided technical assistance to the project. Thanks should also go to the administration members of the Department of Computer and Information Science – Maria Hesse, Dr. Martin Brunner, Krispina Rellstab Kauter, and Barbara Lüthke – who were very helpful in solving any questions whenever I approached them.

I would also like to extend my deepest gratitude to my friends and fellow students Julia Gavriushina and Natalia Poletukhina, who has supported me during my master studies and especially during the work on this thesis.

Finally and most importantly, I wish to thank my family, my boyfriend, and my friends for their love; you helped me to overcome difficulties and substantially supported me during the completion of my thesis. I am especially grateful to my parents Mikhail Zhukov and Alexandra Zhukova for their endless confidence in me that encourages me to move on and never give up.

Contents

List of figures

List of tables

1. Introduction

Nowadays news articles play the role of the main source of information [26]. Some news sources neglect the objectiveness of the reported information and exhibit media bias. *Media bias* denotes a phenomenon of different content presentation in the news articles [28]. Media bias negatively affects news consumption and influences readers' choices and behavior [32]. Frequently, news publishers exhibit media bias by framing covered topic differently, i.e., promoting certain interpretations of events by highlighting certain aspects [26,36]. These information interpretations lead to manipulation with the presented information and, consequently, to readers' switch of the information perception [26]. Framing can depict the information positively or negatively or highlight specific perspectives in information coverage [26]; for instance, Figure 1 shows an example how two differently compiled and framed front pages can drastically change the perception of the Russian president.

Figure 1: An example of framing that promotes a specific perception of the Russian president[1]

Among all types of media bias, framing by word choice and labeling (WCL) is the only type of bias that occurs on the writing stage [26]. Wording chosen to refer to semantic concepts or to contextualize them can distort readers' perception of the article content. WCL depends on a person who conveys information (e.g., politicians, the media, scientific experts, and other opinion leaders [10]), a goal of the intended message (e.g., political elite tend to manipulate popular preferences [10], or media outlet can cover the interests of a particular group of people [26]), an author's writing style [17], or story perspective (e.g., cover an immigration crisis story from an immigrant's perspective [10]).

[1] https://tgram.ru/channels/otsuka_bld

When framing by WCL, journalists can *label* a concept differently, e.g., “invasion forces” vs. “coalition forces” [55], or *choose from various words* to refer to a concept, e.g., “heart-wrenching tales of hardship” vs. “information on the lifestyles” [17]. Not only entities can be framed, but also chosen word choice of predicates influences the text perception as well, e.g., “they won 4-3” vs. “they executed a decisive win (4-3)” [9]. Table 1 demonstrates examples of different types of WCL that can depend on story perspective selection or selection of specific emotion coverage of entities. When unidentified, the difference of WCL strongly influences not-bias-aware users by suggesting a specific emotion evaluation line, thus affecting the decision-making process, e.g., influencing voting preference in the elections [26] and causing false information propagation [15,37].

WCL type	Publisher	Title	Excerpt
Story perspective difference [55]	New York Times	Iraq forces suspension of U.S. surveillance flights	Iraqi fighter jets threatened two American U-2 surveillance planes, forcing them to abort their mission and to return.
	USA Today	U.N. withdraws U-2 planes	U.N. arms inspectors said they had withdrawn two U-2 reconnaissance planes over Iraq for safety reasons.
Amplification of emotion reaction [24]	CNN	UK soldiers cleared in Iraqi death	Seven British soldiers were acquitted on Thursday of charges of beating an innocent Iraqi teenager to death with rifle butts.
	Al Jazeera	British murderers in Iraq acquitted	The judge on Thursday dismissed murder charges against seven soldiers, who are accused of murdering Iraqi teenager.

Table 1: Excerpts with comparable semantic meaning but different WCL hence article perception

Social sciences have been studying framing for decades and have successfully employed content and framing analysis methodologies to reveal the difference of WCL [26]. In qualitative *content analysis*, researchers systematically describe the meaning of texts by annotating the text with predefined categories derived from a specific research question [48]. Among various categories, the researchers can annotate frequently appearing actors, their properties, actions, etc. The researchers focus on the meaning and interpretation of the coded excerpts, thus capturing latent connotation of the employed word choice used to refer to the predefined categories. *Framing analysis* focuses on estimation of how readers perceive the information [10]. To analyze the framing effect, the researchers read and interpret the text and then annotate the most influential parts of the text that exhibit bias [18]. Framing analysis extends content analysis by answering two combined analysis questions: “what information is comprised in the text?” and “how this information is perceived?” [19,26].

Despite being advanced well-studied techniques, manual content and framing analyses are time-consuming procedures that require long periods to achieve reliable results. Moreover, the techniques do not scale to a large number of articles released daily [29]. Conversely, the existing automated approaches either do not resolve different in wording but semantically related phrases

referring to the same concepts, or yield simple results of the word choice difference, e.g., lists of the dissimilar words between two publishers, that are often superficial or require interpretation based on the domain-specific knowledge (cf. [43,44,50]). In contrast to the existing automated approaches, mainly designed and employed by the social sciences, Artificial Intelligence (AI) and computer science's natural language processing (NLP) methods demonstrate the capability of addressing framing by WCL [26].

This leads to the following research question (RQ):

How can we automatically identify instances of bias by WCL that refer to the semantic concepts in English news articles reporting on the same event by using NLP?

To answer the research question, we derived the following research tasks (RT):

1. Design and develop a modular WCL analysis system;
2. Develop a usability prototype with interactive visualizations to explore the results of the WCL analysis;
3. *Research, propose, and implement an approach based on NLP methods to identify semantic concepts that can be a target of bias by WCL;*
4. Evaluate proposed semantic concept identification approach.

Given the research tasks, the thesis is structured as follows:

- Chapter 2 provides an overview of the content and framing analysis methodologies used by the social sciences, approaches of automated WCL analysis, and tasks that address semantic concept identification;
- Chapter 3 explains a methodology of automated WCL analysis system and describes a WCL analysis system architecture and implementation;
- Chapter 4 proposes a multi-step merging approach that identifies semantic concepts in the news articles;
- Chapter 5 evaluates the proposed multi-step merging approach and describes use cases that demonstrate the functionality of the usability prototypes;
- Chapter 6 discusses the evaluation results and outlines future work;
- Chapter 7 concludes the thesis with a summary.

2. Related work

In the related work overview, we start with methodologies of manual content and framing analyses (Section 2.1), then proceed with a comparison of automated WCL analysis methods, describe NLP tasks related to the semantic concept identification (Section 2.2), and, finally, conclude the chapter with a summary (Section 2.3).

2.1. Manual identification of framing by WCL

Social sciences researchers employ content and framing analyses to identify the biased language in the news articles. *Content analysis* focuses on the identification and characterization of semantic concepts based on annotation of referential phrases by adhering to a hypothesis, a task, or a coding book, whereas *framing analysis* studies influence of the WCL instances on the readers' perception [26]. For example, if a content analysis task is to identify frequently occurring actors in the text, then the analysis will require annotating referencing anaphora, e.g., phrases "Donald Trump," "forceful Mr. Trump," and "the blame-averse president" will be annotated as a "Trump" semantic concept. Additionally, estimation of what kind of emotion reaction these annotated excerpts or the context words will cause can be a possible task of framing analysis.

Content analysis consists of two parts: *inductive* content analysis and *deductive* content analysis [26]. In an inductive phrase, the researchers start with a hypothesis or a research question and create a preliminary *coding book*, i.e., a set of categories that the researchers expect to identify in the text and the descriptions of these *codes*. To improve the *concept-driven* coding book, the researchers collect news articles and manually annotate the text fragments that match the codes in the coding book. The inductive analysis is performed multiple times and, at each iteration, the researchers revise the coding book to formulate comprehensive coding scheme based on the analyzed text, thus resulting in a *data-driven* coding book [48]. In turn, when conducting the deductive content analysis, the coders annotate the text elements with the previously defined categories adhering to rules and descriptions of the coding book. Schreier provides an example of semantic concepts annotation; she coded actors of a popular newspaper cartoon such as friends and enemies of the main character [48].

Framing analysis explores the reasons why readers perceive the news differently [26]. Framing analysis also consists of two phases: inductive and deductive. In the inductive analysis, the social science researchers identify influential text elements called *frame devices*, e.g., words and phrases, by reading and interpreting the text [10,18,19], and *framing properties*, i.e., the way how readers evaluate the framing devices [26]. The researchers systemize the findings in a framing coding book. The deductive analysis phase resembles content analysis and, given a framing coding book, coders need to identify and annotate framing devices and their influence adhering to the defined rules.

One of the framing analysis types is the identification of the *equivalency* framing [10], i.e., when "different but logically equivalent phrases cause individuals to alter their preferences" [32]. For example, while applying different labeling, positive labeling leads to favorable associations in memory, whereas negative labeling can trigger harsh associations [15]. Framing devices allow identifying equivalency frames caused by contrasting WCL referring to the semantic concepts or

framing the context of a neutral reference. In contrast, another framing analysis type focuses on the *emphasis* frames that refer to the prominent contrasting word choice. The emphasis frames highlight potentially relevant considerations thus leading readers to focus on these considerations, e.g., an economic or a national security focus in the covered topic [15].

2.2. Automated identification of framing by WCL

Existing automated approaches that reveal framing by WCL identify the phrasing difference from *topic* or *actor* perspectives. Most of the approaches estimate the difference of word choice based on the word frequency, but the others extract more advanced features and apply model training.

Tian et al. analyzed the similarity and difference of word choice covering SARS crisis topic published in CNN and BBC by extracting the interconnected words used in the concept networks formed by forty most frequent words [54]. The human interpretation was required to understand the system outcome and comment on similarity or dissimilarity of the extracted emphasis frames, but the analysis revealed a similar general trend of the word choice in covering the issue.

Papacharissi et al. also analyzed the difference of the word choice in the topic coverage from the perspective of the most frequent dissimilar words [44]. They applied CRA and compared the word choice among four U.S. and U.K. publishers, i.e., they extracted the most influential words that frequently occur in combination with other words. The most influential words formed chains of word choice that depicted the most prominent aspects of the terrorism issue. With some interpretation, the analysis revealed different emphasis frames, i.e., the Washington post focused more on the National security, whereas the Financial Times tried to broaden the audience's understanding of terrorism by frequently covering the international players. Similarly, Garyantess et al. applied CRA to compare word choice between CNN and Al Jazeera and identified how two outlets covered the same factual circumstances [25]. Unlike Papacharissi et al. who have identified word choice related to the emphasis frames, Garyantess et al. post-analyzed the extracted lists of dissimilar word choice to identify equality framed, thus yielding two different word choice of the conflicting ideologies [25].

Fortuna et al. analyzed the coverage of the terrorism topic and trained the support vector machine (SVM) classifier to learn the word choice difference between four outlets of a 21,000 articles corpus [24]. Unlike the other approaches, the researchers used a calculated metric to compare the word choice between the outlets. They used break-even points in the classifiers as the similarity measure between the outlets. The researchers revealed that the International Herald Tribune was similar in WCL to Detroit News and that Al Jazeera applied the most dissimilar word choice to all other outlets. Additionally, the researchers applied Kernel Canonical Correlation analysis (kCCA), an approach frequently used to analyze bilingual corpora to extract related words in the two languages, and identified the contrasting word choice between Al Jazeera and CNN. The analysis outcome showed a comparison from the actor perspective and revealed comparable pairs of words such as "militants" vs. "settlers," "missiles" vs. "barriers," and "launch" vs. "farms."

One of the ways to analyze word choice difference of the same semantic concepts, i.e., word choice difference from an actor perspective, is to annotate referential phrases manually. For example, as an additional analysis of the automated identification of word choice in the topic

coverage, Papacharissi et al. performed the qualitative analysis and revealed some equivalency frames [44]. For example, the New York Times applied more negative labeling to refer to the terrorist-related actions (e.g., "vicious," "indiscriminate," "jihadist," etc.), but the other publishers employed more careful word choice and referred to the terrorist suspects as "alleged" conspirators.

Senden et al. analyzed the word choice associated with actors "he" and "she" by employing latent semantic analysis (LSA) to one hundred most frequent words in the corpus and estimating valence of the analyzed words by calculating a regression with a word list of Affective Norms of English Words (ANEW) [6,50]. The analysis demonstrated that among 400,000 articles the context of "he"-pronoun was more positively framed than of "she"-pronoun. Moreover, they showed that "she"-pronoun was associated with gender defining context, e.g., "woman," "mother," and "girl," whereas "he"-pronouns occurred in the context of action.

Card et al. focused on extracting so-called "personas" that explained WCL from the actor perspective [8]. Unlike our definition of an actor, i.e., an actor is a single unique person, personas do not refer to a specific actor or an entity but represent frequently appearing characterization of entities within a large topic. The researchers applied a statistical Dirichlet persona model and identified, for instance, multiple personas within an "immigrant"-entity, e.g., one "alien"-persona was associated with criminals and another "alien"-persona had a connotation of "working people." They ignored direct mentions of the entities and if the multiple officials occur in the text, the model yields one "agent"-entity associated with police, officials, and authority.

Unlike the previous approaches, Recasens et al. studied WCL as identification of biased language between pairs of paraphrased sentences that are excerpts of the Wikipedia articles [46]. Their experiments with training logistic regressions on features containing markers with bias inducing words yields somewhat worse performance than human bias identification by annotators from Amazon Mechanical Turk (AMT). The study yielded a set of features that can be reused in the identification of the word choice difference both from actor and topic perspective, and additionally, help estimate frame properties, i.e., evaluate how readers perceive contrasting wording.

Analyzing the WCL problem from an actor perspective, the described approaches concentrate on the analysis of the large text corpora. The typical analysis goal is to identify a broad context of the actor coverage, whereas our RQ specifies a requirement to develop a system capable of identification of the semantic concepts contained in the articles related to one event, i.e., a rather small text corpus. Additionally, none of the automatic approaches targets identification of semantic concepts' mentions whereas semantic concept identification is a requirement for the WCL analysis system.

When analyzing framing by WCL, we need to identify and resolve phrases referring to the same semantic concept. The following NLP tasks address entity mentions categorization: coreference resolution, named entity recognition (NER), and cross-document coreference resolution.

Coreference resolution resolves pronominal, e.g., "Donald Trump" and "he," and nominal, e.g., "Trump" and "the president" (cf. [11,33]), anaphora. Coreference resolution identifies chains of the entity mentions only within one text and does not yield one group of entity mentions but multiple chains of mentions. NER extracts text elements and classifies entity mentions into

predefined categories, e.g., persons, organization, location, etc. (cf. [23]). Each category contains multiple phrases, but NER does not specify if two or more phrases are coreferences. Cross-document coreference resolution disambiguates identical or similar phrases referring to different entities (cf.[16,52]); the approach resolve mentions only of a common knowledge represented in the knowledge bases. Despite high performance of the approaches (e.g., ~80% of coreference resolution [11,33]), the approaches lack the functionality of resolving anaphora of a broader meaning, e.g., “undocumented immigrants” and “illegals,” across multiple documents covering one event.

2.3. Summary

Framing analysis is a well-established methodology that is successfully applied by the social sciences to identify framing by of WCL manually. Despite being advanced, manual approaches are very time consuming, cannot scale to the larger topics, and are not capable of analyzing the news articles in real time. Multiple approaches were proposed to identify the WCL difference in the news coverage automatically, but, the approaches are limited either to the identification of contrastive word choice among publishers or to the extraction of a general context associated with a particular entity. Moreover, the approaches do not target assembling semantic concept mentions within a small set of topics.

Unlike the existing WCL approaches, state-of-the-art NLP techniques such as NER and coreference resolution address a problem of entities’ references categorization, but their functionality is limited to resolving anaphora of the common meaning, which is frequently represented in the knowledge bases. Determination and resolution of the phrases referring to the semantic concepts in a broader sense across multiple documents yield a research gap in the identification of coreferential anaphora of various WCL.

The following Chapter 3 introduces a methodology of WCL analysis pipeline that identifies semantic concepts and combines WCL analyses from actor and topic perspective and describes a proposed WCL analysis system. Afterward, Chapter 4 proposes and implements a multi-step merging approach that resolves coreferences of a broader sense.

3. Methodology and prototype of the WCL analysis system

While existing automated approaches analyze framing by WCL by comparing contrastive word choice between outlets (cf.[54]) or analyze context words of the frequent entities (cf.[8]), we seek to unify these approaches and enrich the WCL analysis with a resolution of coreferential phrases of the broader sense. Hence, we propose that the goal of WCL analysis is to identify semantic concepts that are target bias of WCL, analyze framing difference within the semantic concepts, and find groups of articles that frame the event similarly by reporting about the semantic concepts similarly [29].

Imitating the well-established social science's methodologies for inductive analysis, the automated WCL analysis pipeline reveals the difference of word choice both on actor and topic perspectives. Figure 2 depicts the following tasks included in the WCL analysis pipeline: (1) data preprocessing, (2) semantic concept identification, (3) framing analysis of semantic concepts, and (4) framing similarity across news articles.

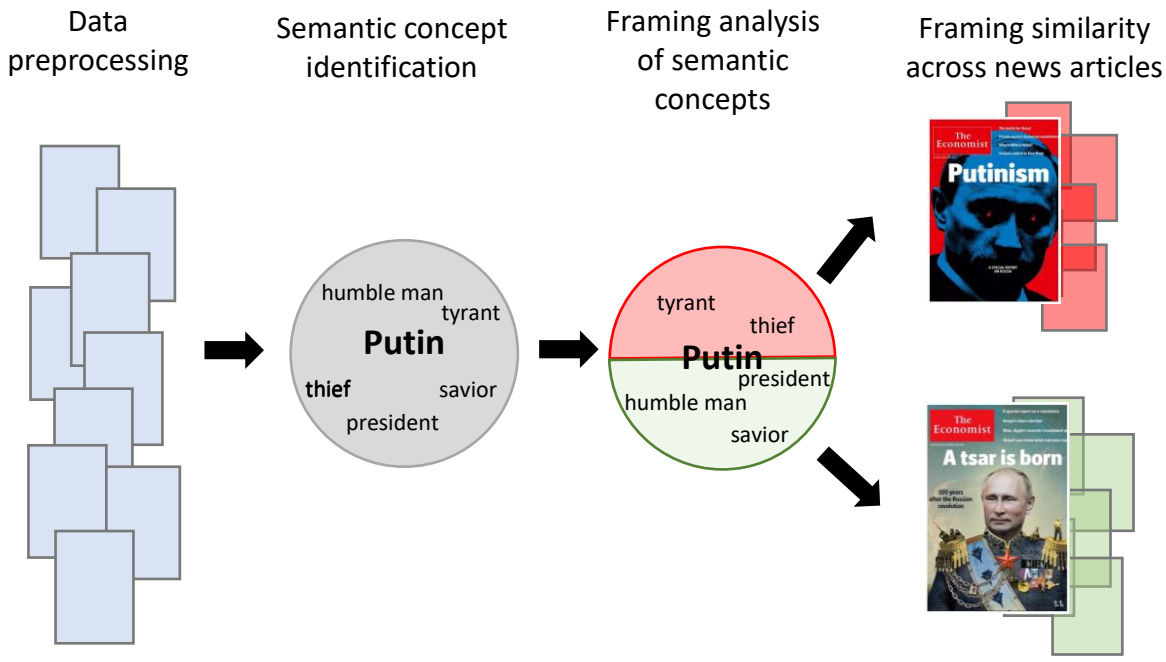


Figure 2: Overview of the WCL tasks[2]

The first task of the WCL analysis pipeline is to preprocess selected articles reporting about one event, e.g., employ standard NLP techniques such as tokenization, POS-tagging, parsing, etc. The second task, semantic concept identification, extracts and aligns anaphora or candidate phrases referring to one semantic concept, e.g., NEs or abstract concepts, to identify the main actors and other concepts covered in the text. The third task, framing analysis of semantic concepts, estimates the effect of semantic concepts' anaphora and their context on the readers' perception. The final

[2] Front pages: https://tgram.ru/channels/otsuka_bld

task of the pipeline, identification of framing similarity across new articles, categorizes articles that use similarly framed WCL to report about an event.

The WCL analysis pipeline plays a role of a roadmap with unified milestones and allows different implementations of the pipeline tasks. In the following Section 3.1, we explain system's architecture and present the first implementation of the WCL analysis pipeline; Sections 3.2 – 3.6 give an overview on the implementation of the system's functional modules; Section 3.7 summarizes the WCL analysis system design.

3.1. WCL analysis system

The WCL analysis system is a module-based implementation of the WCL analysis pipeline (RT1). The system is designed to maintain functional independence of each module. The modular architecture establishes standards in the functionality of each block and describes the requirements of input and output for each module. The analysis system is implemented in Python 3.6 and can be executed fully on a user's computer.

Figure 3 depicts eight sequential functional modules of the WCL analysis system. The system consists of the following modules: (1) preprocessing, (2) candidate extraction, (3) candidate alignment, (4) frame properties estimation, (5) framing analysis, (6) frame clustering, (7) preprocessing for visualization, and (8) visualization. We establish predefined formats of modules' inputs and outputs, thus ensuring standard communication between modules.

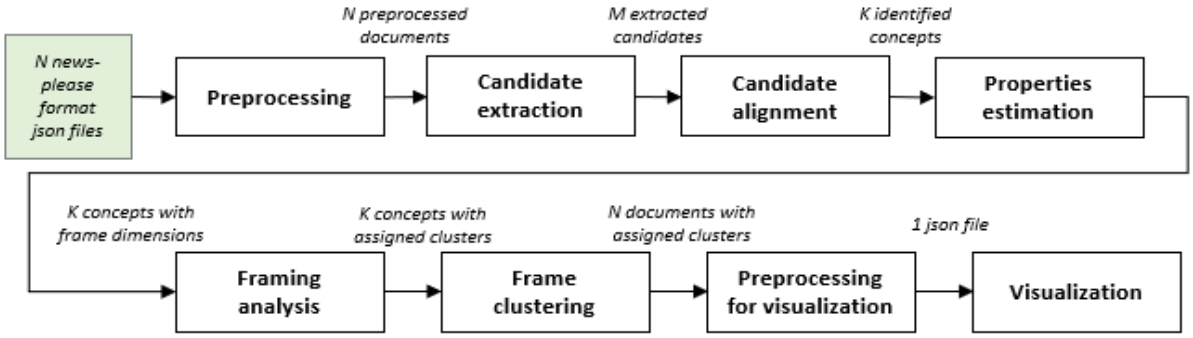


Figure 3: Modular architecture of the WCL analysis system (modules are the white blocks; green blocks are the inputs and outputs of each block)

The system takes as input a set of related news articles of a news-please format [27]. The preprocessing module parses documents and annotates the text resulting in a data format resembling the input but enriched with the preprocessing output fields. Candidate extraction retrieves words and phrases that could refer to the common semantic concepts and outputs a list of extractions. Candidate alignment assembles semantically similar phrases that refer to distinct semantic concepts. Frame properties estimator analyzes how readers perceive and evaluate semantic concepts and assigns frame dimensions to all phrases within identified concepts. Framing

analysis takes as input the enriched concepts and estimates within-concept framing differences. Given the identified intra-concept frames, frame clustering categorizes news articles that report about the same event similarly. Lastly, preprocessing for visualization converts the extracted data structures, i.e., the documents with assigned classes denoting framing similarity, the identified semantic concepts, and the constituting candidates, into a JSON file that is used as a visualization data source to explore the WCL analysis model's results.

The WCL analysis system's modules can be executed fully or partially: the system can be restored from the module on which the system execution stopped before. Each module has reading and writing functionality, and if a user only wants to explore the results of the analysis visually, he or she does not need to execute all modules but only restore saved results of the previous module. Moreover, due to the standard input and output, the system allows comparing different implementations of the same module, e.g., a candidate alignment module, or even a chain of modules, e.g., frame properties estimation and framing analysis.

Although the WCL analysis system contains all previously described modules, we implemented functionality only of the modules related to the semantic concept identification task. Figure 4 depicts our implementation of WCL analysis system that addressed the RQ2. The implementation concentrates on preprocessing, candidate extraction, and, mainly, candidate alignment modules. To enable visual exploration the results of the concept identification task, we created a usability prototype that incorporates simple methods of framing analysis module and a four-view visualization. The following sections give an overview of the methods implemented in the WCL analysis system.

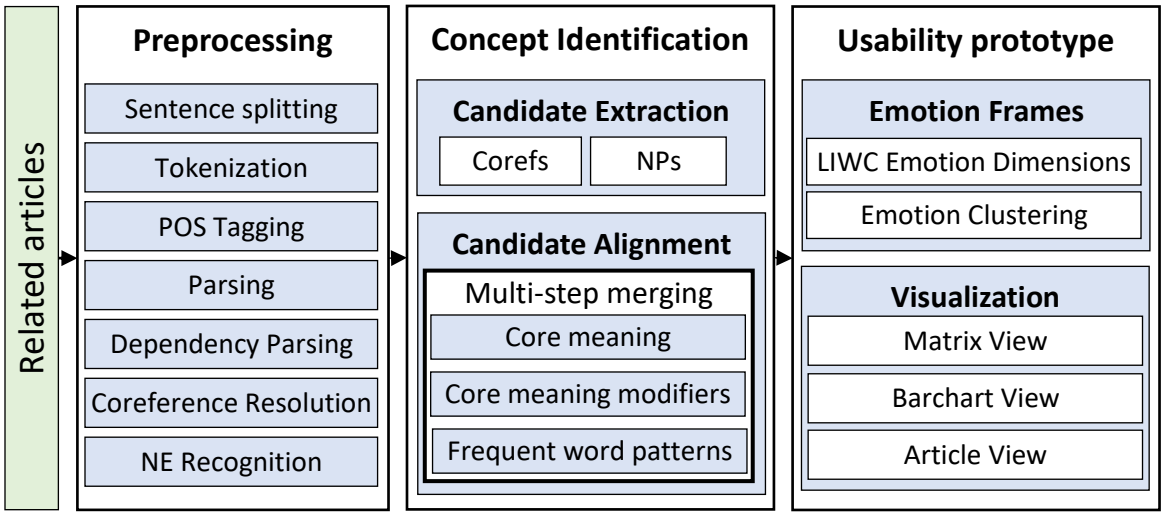


Figure 4: WCL analysis system with the focus on concept identification block

3.2. Preprocessing

The WCL analysis starts with preprocessing a set of related news articles covering one issue or event. The module accepts news articles in JSON format with fields specified by a news crawler news-please [27]. Among all fields specified in news-please, the system analyzes *text content*

fields, i.e., "title," "description," and "text," and uses a field "source domain" as a marker of the political ideology of an article. Based on the news-please format, we create data structure that we call a Document class and use the input fields as attributes of this class.

We use Stanford CoreNLP natural language processing toolkit to preprocess the combined content fields [38]. We split the text into sentences, tokenize sentences, annotate words with POS-tags, parse text into syntactic constituents, apply coreference resolution and extract named entities (NEs). The preprocessing results are saved as additional attributes of the Document class and are cached after the end of the module execution.

3.3. Candidate extraction

Candidate extractor retrieves *candidates* or *candidate phrases* from the news articles, i.e., words and phrases that could refer to semantic concepts. To extract candidates, we employ coreferential chains extracted by CoreNLP [11,12] and additionally extract noun phrases (NPs) that are not included in the coreferential phrases. We extract NPs from the parsing trees and take the longest parent NPs if multiple NPs are chained; we discard NPs longer than 20 words.

CoreNLP produces standard output for coreferences: for each phrase in a coreference chain indicates if a phrase is a representative mention, a head of the phrase, and a type of a coreference chain. A *phrase's head* is a word that determines a syntactic category of a phrase [41]. For each phrase in a coreference chain, we create a specific data structure called a Candidate class and use the abovementioned properties of coreference resolution as Candidate's attributes. Additionally, for each candidate phrase, we extract supplementary properties from the preprocessed text, e.g., parts of dependency trees containing a candidate and all related tokens. We also maintain indexes of the documents sentences which a candidate phrase was extracted.

To convert each NP into a Candidate class, we extract properties similar to those of CoreNLP coreference resolution. First, a representative phrase of an NP is a phrase itself. Second, to identify a phrase's head, we take a word of the highest order in the phrase's dependency subtree. Third, we set an "NP" value as a type of coreference. All other attributes are extracted similarly to a coreferential phrase.

The output of the module is a list of grouped candidates, and a group of size $N > 1$ indicates a coreferential chain and $N = 1$ implies an NP.

3.4. Candidate alignment

Candidate alignment categorizes phrases referring to one concept, aiming at resolving phrases of well-known meaning, e.g., NEs, of broad meaning, i.e., phrases frequently depending on the author's writing style, e.g., "illegal aliens" and "undocumented immigrants," and abstract concepts, e.g., a reaction on something. The general purpose of the candidate alignment task is to resolve mentions of any concept frequently mentioned in the text, but in this work, we limit concept identification to the entity identification. That is, we extract only NP-based candidate phrases and align candidate phrases referring to persons, organizations, countries, groups of people, events, objects, and other entities, excluding more complex concepts such as actions or reactions on some event or accident.

To address the candidate alignment task, we implement a multi-step merging approach, which we explain in detail in Chapter 4. The multi-step merging approach takes groups of candidates, i.e., initially grouped phrases by coreference resolution, then iterates multiple times over the groups of candidates and merges those groups that share similarity on a specific criterion on each step. The approach yields a list of identified entities that is passed to the next module.

3.5. Emotion framing

Emotion framing is a part of the usability prototype (RT2) and implements simple methods for identification of intra- and cross-entity framing similarity. Emotion framing is a joined name for the implementation of frames properties estimation and framing analysis modules (see *Figure 3*).

To each candidate within each entity, we assign psycho-emotional dimensions of LIWC dictionary [45,53]. We consider LIWC dimensions suitable to explain emotion connotation of each phase. To assign emotion dimensions, we iterate over all words in a phrase and calculate the number of times each emotion dimension occurs. A numeric vector of emotion occurrences determines framing properties.

To estimate intra- and cross-entity emotion similarity, we perform two-level clustering of candidate phrases based on their frame properties; Figure 5 depicts the approach. We employ k-means++ clustering twice [3]. First, to identify the intra-entity cluster, we cluster candidates' frame properties within each entity with the number of clusters set to $k_1 = 10$. The algorithm outputs $\min(k_{identified}, k)$, where $k_{identified}$ is the estimated number of clusters. Then, we employ k-means++ for the second time to cluster all emotionally similar groups of candidates with $k_2 = 20$, thus determining which entities contain emotionally similar phrases. Lastly, we output a list of entities where each entity contains an attribute that specifies emotion frame categories and lists candidates belonging to the emotion frames.

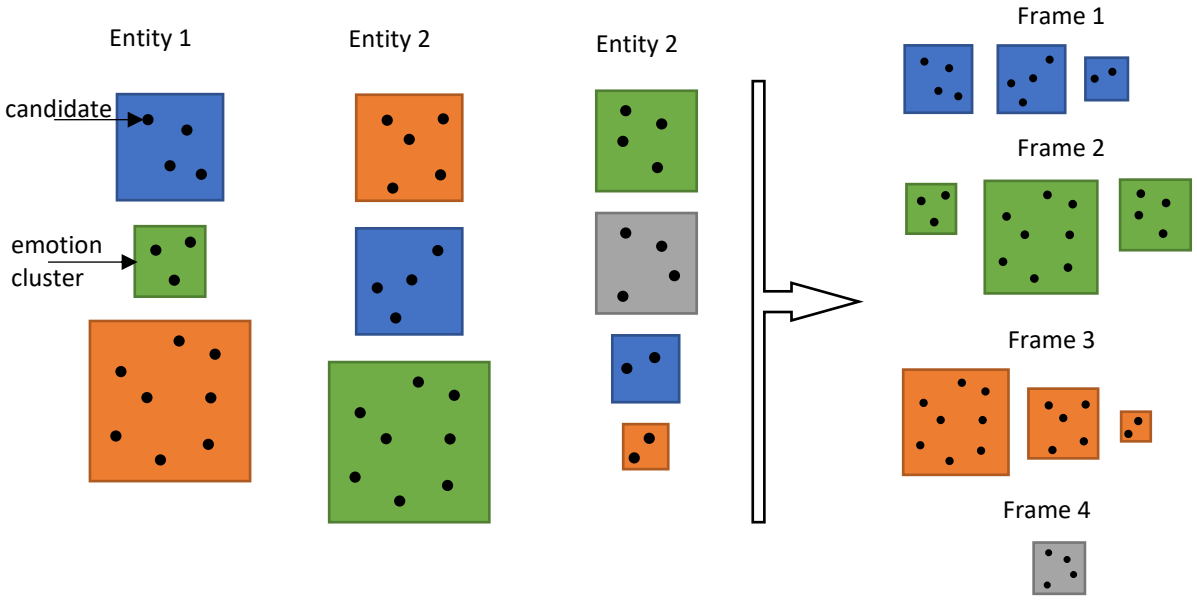


Figure 5: Emotion frames identification

3.6. Visualization

Visualization of the extracted entities is the last part of the usability prototype (RT2). The visualization tool enables a user to explore the results while interacting with different views; the tool is implemented with JavaScript D3.js library.

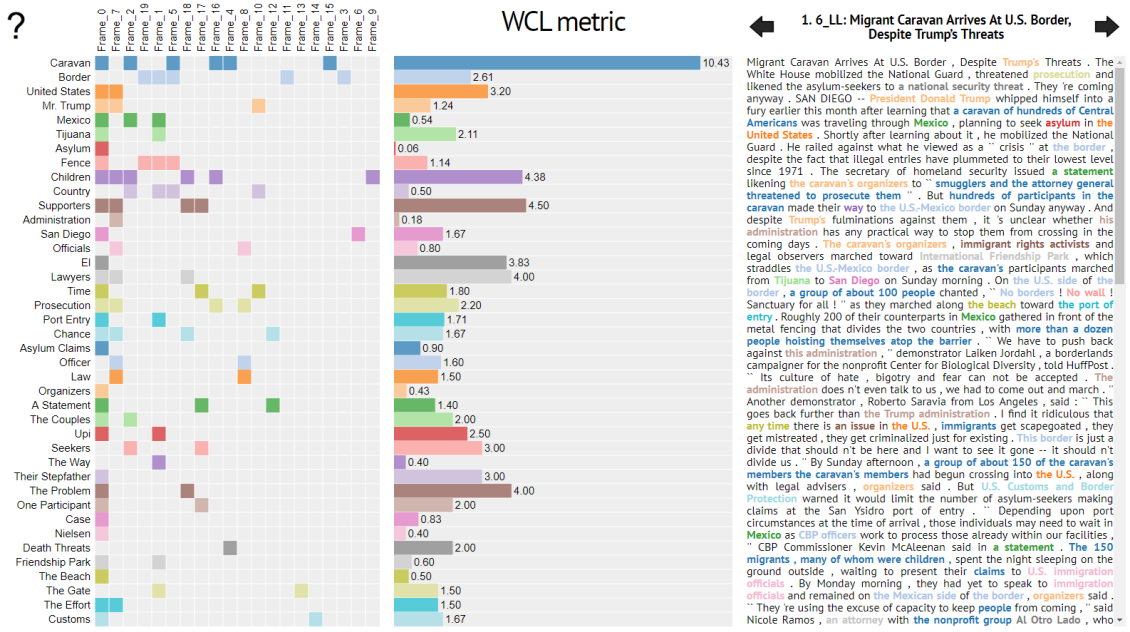


Figure 6: Matrix, bar plot, and article views of the usability prototype

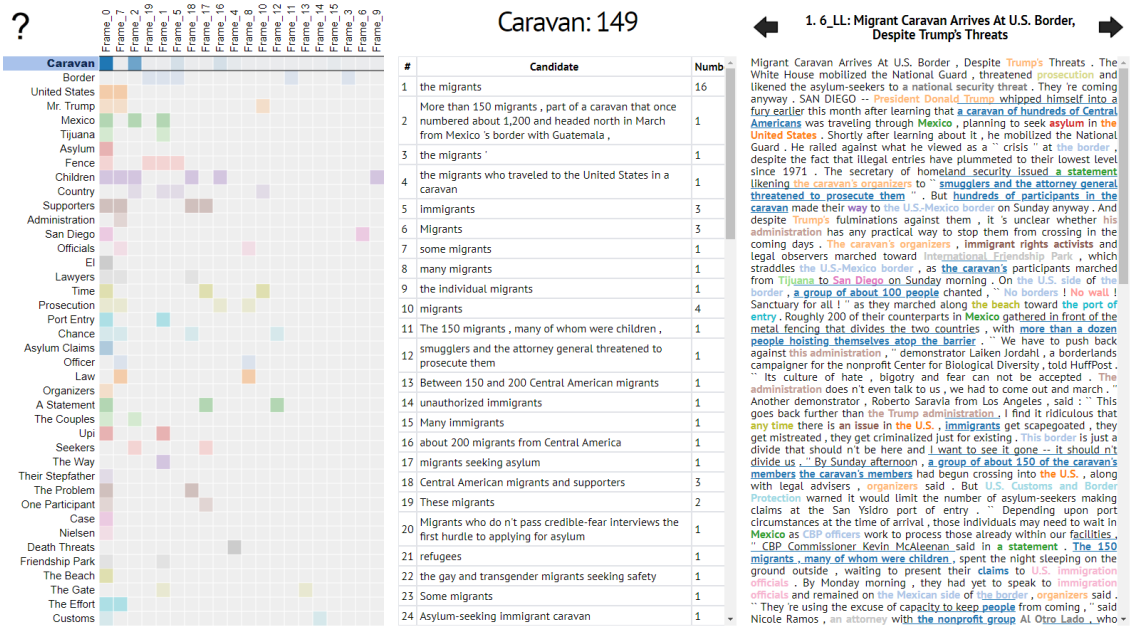


Figure 7: Matrix, candidate, and article views of the usability prototype

Figure 6 and Figure 7 depict four views of the visualization tool: matrix view, bar plot view, candidate view, and article view. Matrix view plots distribution of emotion frames over the entities, bar plot view shows phrasing diversity used to refer to each entity, candidate view (see Figure 7)

lists all candidate phrases when an entity is selected, and an article view shows the original text of the news articles with highlighted with distinct colors phrases, thus referring to the extracted entities.

The *matrix view* plots the distribution of frames (columns) over entities (rows). The entities are sorted in the descending order of their size, i.e., the number of identified mentions; similarly, the frames are sorted in the descending order of the number of phrases that have similar emotion connotation. In the default matrix view, a colored matrix cell indicates the presence of an identified frame within an entity and hovering over a cell shows the number of candidate phases of a particular entity that is framed in a specific way. In the selection matrix view, when an entity or a frame is selected, a different level of opacity encodes the previously hidden number of phrases contained in a cell thereby allowing to compare the cells' values.

The *bar plot view* depicts phrasing diversity of each entity. In Section 5.1.3, we will introduce a WCL-metric that estimates the variance of the word choice of the entity mentions. The higher WCL-metric means higher phrasing diversity. Color code of each bar is identical to those in the matrix view. Labels of the rows in the matrix view correspond to the bars in the bar plot view. When an entity or a frame is selected, the bar plot view is replaced by the candidate view.

The *candidate view* lists all phrases that belong to a selected entity or a frame. The heading of the candidate view shows the name of the selected item and the overall number of phrases comprised. The view lists not only the unique phrases but also the number of times these phrases have occurred in the text.

Finally, the *article view* displays the original article text with highlighted phrases identified as entity mentions. Color code of the highlighted phrases matches the color code of the entities, thus enabling to interlink the views. When an entity is selected, the entity mentions become underlined throughout all articles and allow a user to get an overview of the mention's context (see Figure 7).

3.7. Summary

In this chapter, we presented a methodology of the automated WCL analysis pipeline that imitates the well-established methodology of the inductive word choice analysis used by social sciences. We specified the tasks required to analyze the difference of word choice starting from the actor perspective and moving to the analysis of contrastive reporting about the same event from the topic perspective.

Then, we described the architecture of the WCL analysis system that aims at solving the WCL analysis pipeline tasks. The proposed system consists of eight functional units that extract coreferential phrases of semantic entities and analyze the perception of the reader, i.e., the way how the selected word choice frames the context.

Lastly, we explained the implementation of the WCL system's modules that address the candidate alignment task. The implementation of WCL system focuses on the preprocessing, candidate extraction, and candidate alignment modules, and additionally covers simple methods in the other modules to provide a usability prototype designed to explore the results of the identified semantic concepts.

4. Candidate alignment using multi-step merging approach

Whereas existing approaches for coreference resolution, e.g., CoreNLP, resolve with high-performance coreferential anaphora within one document, they target mainly coreferences based on named entities (NE). Moreover, cross-document coreferences also focus on the proper nouns by resolving their NE and non-NE mentions based on factual information extracted from knowledge bases. The approaches do not address a problem of identification of diverse word choice of the non-NE concepts, e.g., group of people or more abstract entities.

Candidate alignment of different WCL instances mainly focuses on cross-document frequently mentioned non-NE actors and concepts in the news articles that are out of the scope of coreference resolution, e.g., resolution of "DACA illegals" and "young undocumented children." Additionally, candidate alignment includes cross-document coreference resolution for proper nouns. To address the resolution of phrases of a broader sense (RT3), we propose a multi-step merging approach.

The section structured as follows: Section 4.1 discusses essential ideas for the solution of the candidate alignment task, Section 4.2 introduces multi-step merging approach and outlines the merging steps, Section 4.3 and Section 4.4 cover merging preprocessing steps, Section 4.5 – Section 4.10 describe the details of each merging step, and, finally, Section 4.11 concludes the section with a summary.

4.1. Design discussion

General description of the candidate alignment task (see Section 3.4) is defined as follows: given multiple candidates, consolidate them into several groups unified by similar meaning, i.e., determine phrases related to the same frequently used concepts in the analyzed texts. The task description of the candidate alignment resembles the definition of the clustering task. In this section, we discuss the milestones of the multi-step merging approach design development and highlight properties of the influential clustering algorithms that formed the basis of the approach.

At the initial step of the approach development, we faced two questions: (1) how do we compare candidates to each other, i.e., in which order do we process candidates, and (2) which similarity criterion suits determination of semantically related groups of candidates the best. To answer these questions, we conducted several experiments with the state-of-the-art clustering algorithms and analyzed their strengths and weaknesses applied to the candidate alignment task. We incorporated the findings in the approach development.

At the beginning of experimenting with suitable processing order, we had a hypothesis that if two candidates are semantically similar to each other and represent a frequent concept in the text, together with the other related concept mentions, the candidates should be densely situated within a certain semantic distance. We applied DBSCAN, a density-based clustering algorithm [20], that starts a cluster is a sufficient number of points is found within a predefined distance-radius and extends the cluster as long as the cluster points also have a sufficient number of points in their radiuses (see Figure 8). Varying the radius value, we discovered that the algorithm yielded either

few small-size clusters, i.e., having very narrow meaning, with many noise points, or few big clusters merging almost all candidates, i.e., considering all points transitively similar to each other.

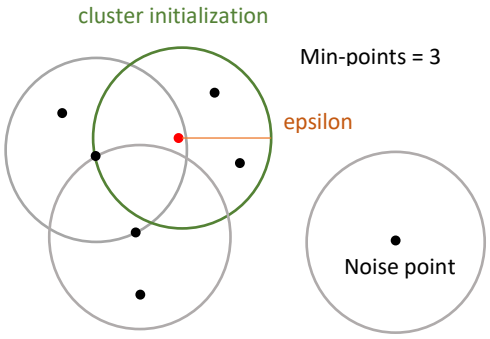


Figure 8: DBSCAN clustering algorithm [20]: if for a randomly chosen point (red) there is a sufficient number of points, i.e., $N \geq \text{minpoints}$, within a predefined radius (green), start a cluster and expand it as long as points in the cluster also have a sufficient number of points in their radiuses

We assumed that the algorithm yielded too small or too big clusters due to the different cluster density of the candidates related to the same concepts combined with DBSCAN's inability to estimate the variant density [2]. The variant cluster density could happen when some candidates related to one concept had straightforward shared meaning, hence being closer in the semantic space, whereas other candidates referring to the other concept were related more subtly, thus leading the bigger distances between the points. To test this hypothesis, we clustered candidates with the OPTICS clustering algorithm that, unlike DBSCAN, determines clusters when points of a true group are not evenly distributed [2].

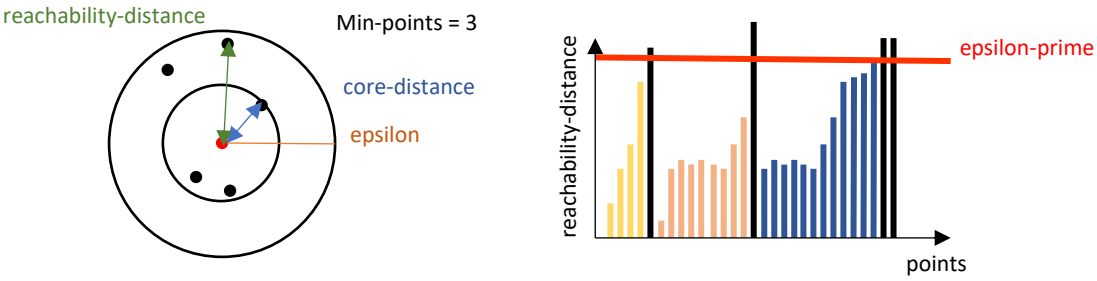


Figure 9: OPTICS clustering algorithm employs two-stage clustering evaluation[2]: (1) OPTICS uses minimum number of points and a threshold to calculate core- and reachability-distances between points and order the points according to the distances (left), (2) given a second threshold and the calculated distances, OPTICS clusters the points

Varying parameters of the OPTICS model, we obtained the results similar to DBSCAN: the model yielded either a small number of small clusters or one unified cluster. Unlike the expected output of OPTICS, i.e., when the algorithm leads clear separation of groups of points in the vector

space (see Figure 9, right), the OPTICS algorithm visualization[3] of our results showed that most of the candidate phrases are located too close in the vector space to identify clear dissimilarity boundaries. We concluded that the density-based approaches are not suitable for the candidate alignment as a principle of processing candidates to determine semantically similar phrases referring to the same concepts.

The experiments with Word2Vec word vector model [39] showed that consecutive pairwise comparison of the semantically similar candidates could resolve concept mentions instead of the density-based cluster identification. To test this idea, we employed a hierarchical agglomerative clustering (HAC) algorithm [14]. The algorithm groups data points by constructing a hierarchy starting from the individual points and then, in multiple iterations, merges clusters in the decreasing order of the similarity value, i.e., identifying the most similar points first, finally yielding one unified cluster (see Figure 10). We incorporated the pairwise candidate comparison into the core idea of candidate processing in the merging approach. That is, we merge candidates when they match specific similarity criterion and proceed until we merge all or most of the candidates related to the semantic concepts into separate clusters.

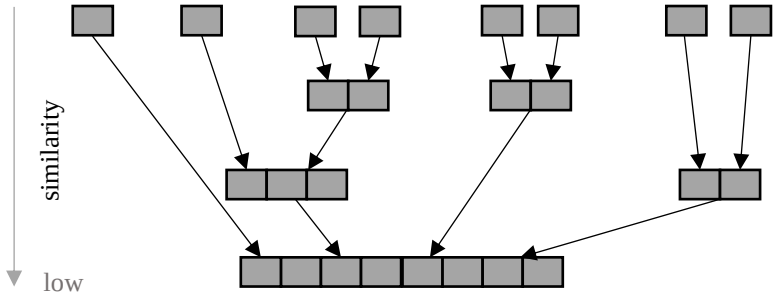


Figure 10: Hierarchical clustering dendrogram [14]: the levels represent the decreasing similarity between points; the grouped points mean that at this threshold the points were similar and thus merged

After we decided about the main processing principle for the multi-step merging approach, we experimented with the similarity criteria, indicating when candidates should be merged. We started the experiments by applying one of the linkage methods employed in HAC called a mean-linkage criterion. When clustering with the mean-linkage criterion, the algorithm merges two sub-clusters in decreasing similarity value calculated between the centers of the sub-clusters, i.e., the algorithm merges the most similar candidates first. In the semantic vector space, it means that two sub-clusters are merged if they have similar average meaning.

The experiments with HAC and the mean-linkage criterion showed that: (1) while varying the cut-off similarity value, the algorithm yielded either big number of clusters of very narrow semantic meaning, or small number of large clusters with a lot of falsely merged candidates; (2) while processing candidates in order from the most similar to the most dissimilar, we do not employ the valuable information of the initially grouped by coreference resolution candidates.

[3] OPTICS implementation in KNIME Analytics Platform: https://www.knime.com/whats-new-in-knime-35#optics

Unlike the agglomerative hierarchical clustering, which merges all sub-clusters when they reach certain cross-cluster similarity, although they might not be mentions of the semantic concepts, DBSCAN merges only points that exceed a threshold of minimum similarity required to consider two candidates or group of candidates similar. Therefore, to address the first problem of merging all candidates with each other, we combined the pairwise candidate comparison principle from HAC and the merging principle of exceeding a minimum similarity threshold from DBSCAN. That is, we proceed with pairwise candidate comparison and merge candidates only if they are similar enough.

To address the second problem of the candidate comparison order, we prioritized the merging procedure by considering candidate chains with the largest number of candidates first. That is, we assumed that the bigger coreferential chain, the more probable that it represents a frequently appearing concept in the text, thereby the other mentions to this concept need to be identified and merged to the bigger chains first.

The first implementation of our approach compared the candidates in the decreasing order of the candidate group size (one coreferential chain or one single candidate represent a candidate group), merged smaller candidate groups into the bigger ones if the mean word vectors of two candidate groups were similar enough regarding a predefined threshold, and repeated the procedure multiple times to follow the dendrogram principle. Despite yielding significantly better results than the previous experiments, the first implementation revealed problems that: (1) the mean word vector of all candidates did not represent a group of phrases well, yielding merging of some unrelated phrases in the first merging level, and (2) starting the second merging level, the approach yielded very mixed clusters.

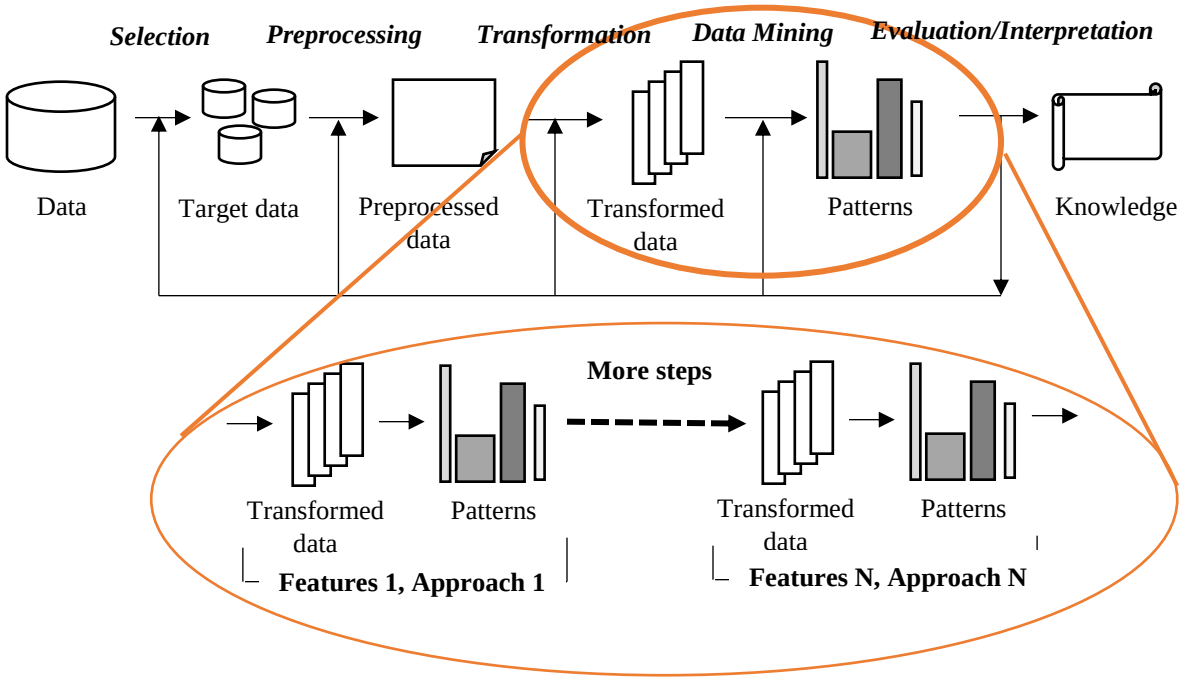


Figure 11: Standard KDD pipeline [22] and suggested enhancement of chaining multiple transformations and data mining steps resulting in the multi-step merging approach

By now, we followed a standard KDD pipeline [22] that includes five consecutive steps (see Figure 11, top): data selection, preprocessing, transformation, data mining, and evaluation. The merging approach covers two steps: data transformation and data mining. During the transformation step, we calculated mean vectors of the candidate groups, employed the mean values as features, and applied the merging approach as the data mining algorithm. The experiments with the merging approach showed that only one feature is not enough to capture semantic similarity between the candidate groups and that only one data mining algorithm is not enough to determine meaningful groups of candidates related to the same concepts.

Figure 11 (bottom) depicts a proposed methodology of the multiple consecutively applied merging steps, each of which includes extraction of the specific features among the candidate sub-clusters, i.e., performs data transformation, and identifies the similarity between the candidate sub-clusters given extracted features, i.e., apply specific data mining algorithms. The idea behind the chaining is to consider candidate sub-clusters from different perspectives and to take into account the already discovered patterns, i.e., extract new features from the results obtained on the previous data mining step. In the new design, each merging step represents one level in the merging dendrogram.

Lastly, we experimented with different features and the way how to determine similar sub-clusters using these features. We discovered that the more sophisticated features could not be compared directly, i.e., an additional calculation step required to find similarities. Akin OPTICS, we incorporated two-stage similarity determination that takes into account two different similarity thresholds, thus enabling to capture more information about the relations between candidate sub-clusters.

The following sections summarize the discussed ideas of the algorithm development, present the multi-step merging methodology, and explain features and merging criteria of each merging step.

4.2. Overview of the multi-step merging approach

Multi-step merging approach (MSMA) is a proposed clustering method to address a candidate alignment task. The method merges candidate coreferential phrases into an entity by identifying similar phrases based on different characteristics extracted from candidate phrases. MSMA includes a local preprocessing step and six merging steps, which include feature construction and similarity identification parts.

Retrieved and initially categorized within the candidate extraction module (see Section 3.3), a list of candidate groups is taken as input into the candidate alignment module. A candidate group can be of two types: coreferential chains of candidate phrases or single candidate phrases (NPs). To merge groups of candidates, we require representative features extracted from each candidate group to compare the groups to each other on these representative features.

From the initial, pre-merging, step, we consider each candidate group as a separate object of an Entity class. Extracted properties from the candidate phrases are assigned to the class as attributes. The following preprocessing steps required to create an entity:

- removal of the uninformative candidate phrases,

- quality improvement of the coreferential chains,
- conversion of a candidate group into an entity and extraction of the entity attributes.

Features, extracted from its candidate phrases and assigned to entity attributes, represent entities in the feature space instead of the candidate phrases themselves. After the conversion from a candidate group to an entity, we will also call the candidates constituting an entity as *entity members*. Section 4.3 and Section 4.4 describe preprocessing steps and identification of one *entity attribute* called an *entity type*, Sections 4.5 - 4.10 cover other entity attributes required for each particular merging step detailed in these sections. All attributes covered in the following sections are initially calculated for each entity during the preprocessing step and, afterward, updated before each merging step. In the following sections, we only highlight the attribute usability for each specific merging step.

To compare entities to each other, each merging step follows the identical procedure depicted in Figure 12. A list of entities is sorted by the entity size, i.e., the number of members in an entity, in the descending order. Then, the first – largest – entity is alternately compared with the remaining smaller entities; if the current (smaller) entity is similar on a specific criterion to the first entity, we remove it from the list and merge the members of the smaller entity into the first entity. We compare the first entity to all remaining entities in the list, and afterward take the second (third, etc.) biggest entity and repeat the comparison procedure until all entities compared to each other. After all merging steps executed, each entity represents a dendrogram similar to a depicted dendrogram in Figure 10, but where every dendrogram level is a result of a specific merging step.

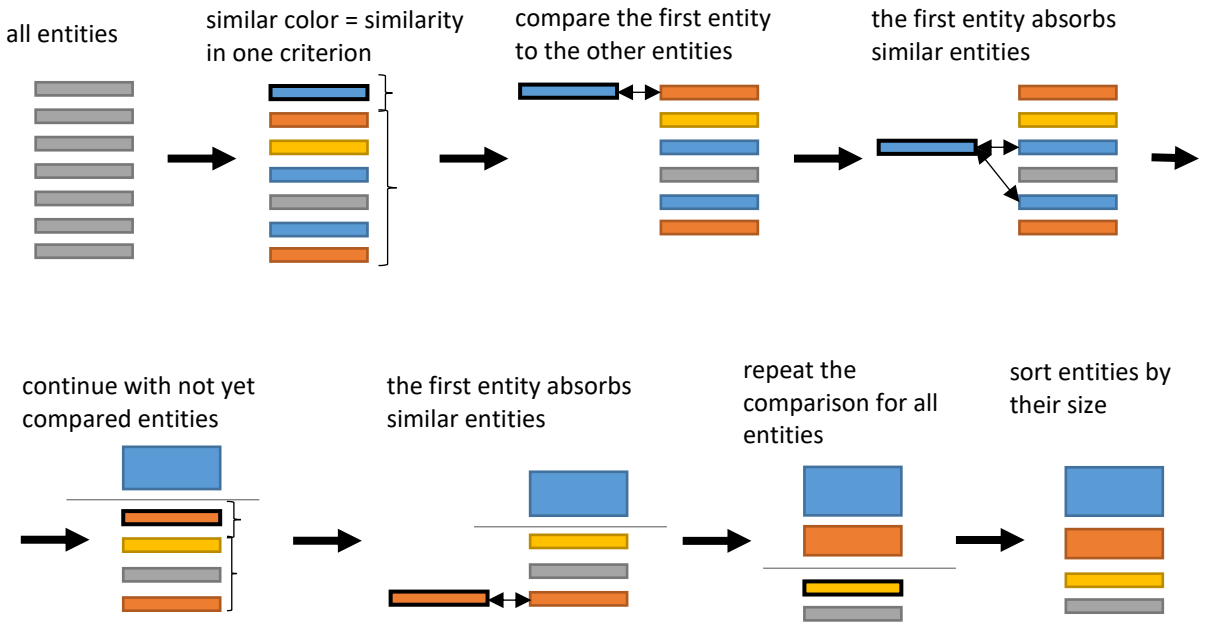


Figure 12: Comparison procedure of the merging steps: start with the bigger sub-clusters and merge similar sub-clusters

The multi-step merging examines the similarity between entities in the increasing complexity of the representative attributes. The first, simpler, merging steps allow assembling sufficient

number candidates for each entity to calculate the more sophisticated attributes required for the more next merging steps. The first and second steps focus on similarity on the entity core meaning comprised in the phrases' heads of entity members, e.g., "nuclear *program*"; the third and fourth steps target similarity between core meaning modifiers, i.e., adjectival NPs; the forth and the sixth steps identify frequent word patterns within each entity and compare entities using the extracted patterns.

4.3. Entity preprocessing

The goal of entity preprocessing is to remove uninformative candidates, improve quality of the candidate groups and, afterward, transform preprocessed candidate groups into entities, i.e., objects that not only include all candidate phrases but also contain attributes extracted from the candidates.

Given the CoreNLP accuracy of 80% [12], the preprocessing step removes the erroneously extracted information, e.g., single articles extracted as NPs, and improves the quality of extracted information, e.g., splits a coreference chain into multiple chains if a chain contains most likely erroneously coreferenced phrases. Additionally, the step eliminates the candidate phrases that are non-informative for the WCL analysis, e.g., if an NP consisting only of the stop-words.

The preprocessing step eliminates candidates if the phrases do not fulfill a straightforward definition of an NP, i.e., a phrase's head is not a noun. Therefore, we remove candidates that were fully tagged wrongly as NPs (e.g., single-word adjectives or articles) or that had a possessive ending identified as a phrase's head (i.e., "'s"). Additionally, if a phrase was tagged correctly as an NP, but we consider a phrase as non-informative, such a phrase was a subject to removal. A candidate is uninformative for WCL analysis if:

- a phrase's head is a pronoun, e.g., personal or demonstrative;
- a phrase's head is classified as a time-related NE category, e.g., a month;
- an NP consists only of the stop-words.

The improvement of extracted information contains two parts: splitting a coreference chain into multiple chains if we consider a chain containing falsely grouped phrases and, in specific cases, changing a phase's head into a different noun within the same NP.

The motivation for a candidate split is to make the results of the merging steps, especially the early steps, as much error-prone as possible, therefore, ensuring the homogeneity of coreferential candidates, i.e., leaving candidates related only to one concept. The early merging steps analyze the core meaning comprised in the phrase's heads, and if a coreferential chain contains false positive phrases, they will have a certain impact on the merging results. To minimize the risk of merging false positives entities, we created a set of rules based on the most commonly observed cases of falsely categorized phrases into a coreference chain. We assume that there is a chance of the candidate split leading to a possible loss of few interesting WCL cases, but we choose the minimization of the number of false positives over the loss of few relevant phrases.

To estimate if a group of phrases correctly belong to the same coreferential chain, we designed a set of handcrafted rules applied to the phrases' heads within a coreferential chain. If any of the following rules fires, we remove a candidate or multiple candidates from the considered

coreferential chain into the separate chains and then create different entities out of them. A candidate coreferential chain is split if:

- two NE categories, organization and person, are present among the headwords, e.g., "President *Trump*" and "Trump's *administration*"*;*
- two NE categories, organization and title, are present among the headwords, e.g., "*President*" and "*Congress*"*;*
- several heads of phrases are classified as NE-person category, e.g., "Michael *Flynn*" and "James *Comey*"*;*
- some heads of phrases are classified as NE-person category, and some heads have neither an NE category nor WordNet synsets with a person category, e.g., "President *Trump*" and "*accident*"*;*
- among the chain of phrases' heads, there is one non-NE phrase's head that occurs in less than 20% of the overall number of phrases, e.g., "*DACA*" (10 times) and "*shadows*".

Some candidate phrases' heads are less informative for WCL analysis than their noun modifiers, e.g., "*hundreds* of immigrants" or "a *group* of terrorists." For the cases where a phrase's head is a number or is a word "group," we set as a phrase's head a dependent word, e.g., "hundreds of *immigrants*", and use this modified phrase structure in the merging steps. Such manipulation helps to preserve the core meaning of a phrase in its head.

The last preprocessing step is to create an entity out of the preprocessed candidate groups, i.e., convert the groups into separate objects with candidates as entity members and extract attributes for each entity. Entity attributes, e.g., an entity type or representative wordsets, will be used to estimate similarity at the merging steps. The attributes are calculated during the conversion procedure and will be updated for those entities that will absorb smaller entities. More attributes such as emotion dimension are added for the emotion frame identification module (see Section 3.5).

4.4. Entity type determination

An *entity type* is an entity attribute to indicate a type of the real-world objects or concepts, e.g., persons, organizations, countries, etc., that an entity represents. An entity type plays the role of a controlling element to specify type-to-type comparisons in the merging steps.

In the multi-step merging, we use nine entity types presented in Table 2. Eight of nine entity types (e.g., "person-nn" or "country-ne") originate from the predefined NE categories used in the NER [42] and the ninth type – "misc" – refers to abstract concepts that did not fall into any NE category. To identify an entity type, we use NE categories from CoreNLP and a lexicographical dictionary from WordNet [40] that specify if an entity type is an NE or a non-NE object respectively. As an additional source for more detailed entity type identification, we incorporated POS tags from CoreNLP tokenization.

	Type definition			***Type source***		
Entity type	*Definition*		*Example*	*Source: category*		*POS tags*
person-ne	NE	Single person	Trump	CoreNLP	Person	
person-nes		Multiple persons	Democrats		Person + Organization	NNS or NNPS
group-ne		Organization	Congress		Organization	
country-ne		Country	Russia		Country, Location, State or Province, City	
person-nn	non-NE	Single person	immigrant	WordNet	noun.person	NN or NNP
person-nns		Multiple persons	Officials		noun.person	NNS or NNPS
group		Group of people, place	crowd, court		noun.group	
country		Location	country		noun.location	
misc		Abstract concepts	program			

Table 2: Entity types used in the multi-step merging

Before the entity type identification, we construct an NE-dictionary from the candidates identified as NEs and perform several preprocessing steps. As stated in the Section 4.3, the performance of CoreNLP leads to some misclassified phrases. We created a set of hand-crafted rules to minimize the ambiguity of the NER results thus improving the results of the MSMA.

We create three bins with categories used for the entity type identification: person, group, and country. For each bin we collect words and phrases that have the following NE categories:

- person-bin: Person;
- group-bin: Organization, Ideology, Misc;
- country-bin: Country, Location, State or Province, City.

Category bins serve to unify some NE categories and improve the misclassification issues. Phrases selected for country-bin have only a couple of misclassification cases within the country-related group of categories. On the contrary, the phrases such as “Democrats” or “Republicans” tend to be classified with different NE categories, e.g., within one article these phrases can be classified as either as Organization or Ideology.

As the last preprocessing step for the NE dictionary construction, we check if heads of NE phrases are uniquely assigned to each bin, e.g., all phrases with a phrase’s head “Trump” must be in the person-bin. If the same phrase’s head was placed in two bins, then we remove all NE phrases from the bin with the fewer number of occurrences.

We identify an entity type by calculating the frequency of each entity type assigned to one phrase and choosing the most frequent type as an entity type. We add a “ne” suffix to a type-basis – person, group, or country – if a phrase belongs to the corresponding bin in the NE dictionary. To

identify if an entity is a single person or multiple persons, we used POS tags assigned to the head of a phrase. Figure 13 depicts the pseudocode with detailed entity type identification procedure.

```
Input: head_list:= [h1, h2, …, hn] (a list of entity members' heads of phrases),
       NE_dict (NE phrases obtained in the NE dictionary preprocessing step)

# init score structure

score_array:= zero-array of the input list length
Score_struct:= {"person": {"ne": score_array,
                           "non-ne": score_array},
                "group": {"ne": score_array,
                           "non-ne": score_array},
                "country": {"ne": score_array,
                           "non-ne": score_array}}
Types = ["person", "group", "country"]

# entity type identification

for head in headlist:
   for type in Types:
       Score_struct[type]["ne"]:= 1 if head in NE_dict[type] else 0
       Score_struct[type]["non-ne"]:= sum(number Wordnet synsets of head where
                                          synset.lemma== type)

# type-base identification

Type_base:= argmax(score_sum[i]) if max(score_sum) > 0 else "misc", where
    score_sum[i]:= score_struct[Types[i]]["ne"] + score_struct[Types[i]]["non-ne"]
       and i=0..2

# type-suffix identification

if Type_base == "person":

   if sum(Score_struct["person"]) > 0 and sum(Score_struct["group"]["ne"]) > 0:
      Type_suffix:= "-nes" if sum(number of heads where POS tag is NNS or NNPS) >=
                              sum(number of heads where POS tag is NN or NNP) else "nn"
      return "person-nes" or "person-nn"

   if sum(Score_struct["person"]["ne"]) > 0:
      return "person-ne"

   else:
      Type_suffix:= "-nns" if sum(number of heads where POS tag is NNS or NNPS) >=
                              sum(number of heads where POS tag is NN or NNP) else "nn"
      return "person-nn" or "person-nns"

if Type_base == "misc":
   return "misc"

else:
   Type_suffix:= "-ne" if sum(Score_struct[Type_base]["ne"]) > 0 else ""
   return "group" or "group-ne"; "country" or "country-ne"
```

Figure 13: Pseudocode of entity type identification

For each merging step, we specify a *comparison table* $ctable_i$ that is a 9 x 9 matrix, where the size of each dimension refers to the number of entity types. A comparison table has two purposes: first, to define if two entities are comparable by their entity types (two entities are *comparable* if $ctable_{i,x,y} \geq 0$, where x and y are entity types), and second, to specify a threshold of the minimum similarity required to consider two entities similar. Typically, a comparison table allows merging two entities of the similar or adjacent types, e.g., "country" and "country-ne". Comparisons between other types are possible but these comparisons can require a higher, i.e., more restrictive, similarity threshold. We identified general default parameters for the comparisons for each merging step but custom parametrization for a specific set of articles is also possible.

4.5. Merging using representative phrases' heads

Representative phrase's head is an entity attribute that originates from an inherited coreference resolution attribute called a *representative phrase*. Every phrase in a coreference chain has a flag set to true if a phrase is a representative member of a chain or to false, otherwise; a representative phrase comprises the meaning of the entire chain. We extract a head of a representative phrase and proceed with merging comparing the heads of the representative phrases.

As the first merging step, two entities are merged if their heads of phrases are identical by string comparison;

Figure 14 depicts the merging procedure. By default, we apply the first merging step only to NE-based types, as set in the $ctable_1$.

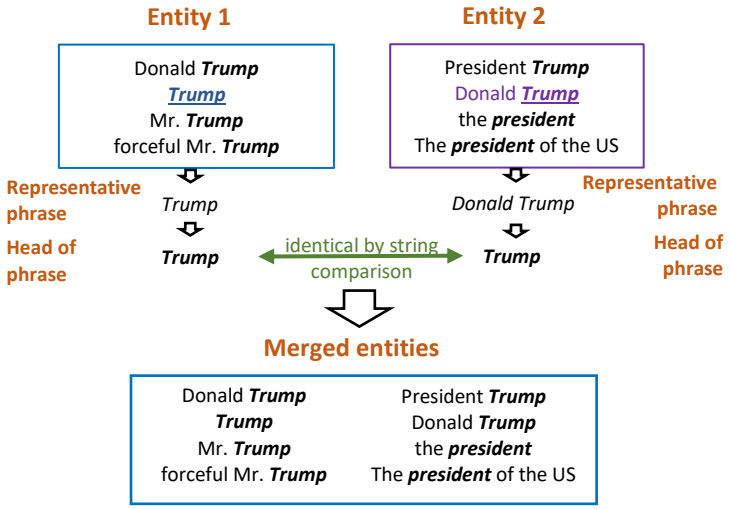


Figure 14: First step: merging using representative phrases' heads

4.6. Merging using sets of phrases' heads

Merging using *sets of phrases' heads*, the second merging step, finds similarity between two entities as a combined core meaning of all heads of entity members. Unlike the first merging step

(Section 4.5), we compare two entities in the word vector space. Figure 15 depicts the second merging step.

Before merging, we extract unique heads of phrases within one entity and vectorize the sets of phrases' heads into the word embedding space. We use a state-of-the-art Word2Vec 300-dimensional model trained on the Google News corpus[39]; the model is the improved version of the originally proposed model, and it enhances the model performance by representing in the vector space not only words but also frequently occurring phrases.

Then, we determine the semantic similarity between sets of phrases' heads $V \in e_0$ and $W \in e_1$ by calculating a similarity value $\text{sim}(V, W)$ for the entities that are comparable:

$$sim(V, W) = cossim(\vec{V}, \vec{W}) \quad (1)$$

where $\text{cossim}(\dots)$ is the cosine similarity function, $\vec{V}$ and $\vec{W}$ are mean head vectors of the entities. Likewise in DBSCAN (see Section 4.1), we seek to merge e_1 into e_0 if e_1 is within an epsilon-radius, but because we use similarity metric instead of the distance, we merge two entities if $\text{sim}(e_0, e_1) \geq t_2 = 0.5$, where t_2 is a type-to-type threshold $t_2 = \text{ctable}_2[e_0.type][e_1.type]$.

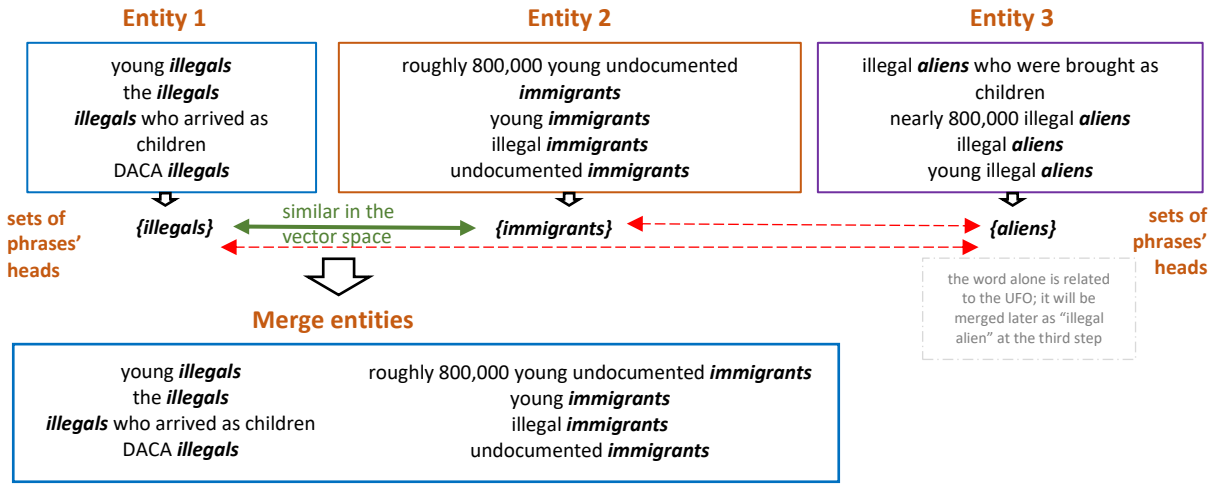


Figure 15: Second step: merging using sets of phrases' heads

One of the most significant disadvantages of DBSCAN is the $O(n^2)$ runtime. Unlike DBSCAN, where all points are examined for being potential candidates for clustering, the goal of merging is to assemble entities of similar entity type or adjacent entity types, e.g., "person-nn" (immigrant) and "person-nns" (illegals). Therefore, to minimize the runtime, we compute cosine similarity only if two entities are comparable, hence minimizing the run time from $O(n^2)$ approaching to $O(n \log(n))$ with some configurations of comparison tables.

4.7. Merging using representative labeling phrases

Merging using *representative labeling phrases* searches for patterns among adjectival NPs that are parts of the entity members. Unlike merging using the sets of phrases' heads where we merge entities with a similar core meaning, the idea behind merging using representative labeling phrases is to identify similarity based on the core meaning modifiers, e.g., adjectives. If the core meaning of two entities is not specific enough, the vector space model will reflect it as a bigger distance between two points and fail to merge the entities at the second merging step (Section 4.6). On the contrary, semantically similar labeling brings a shared meaning and, therefore, minimizes the dissimilarity in the vector space and makes merging possible. We noticed that entities mostly affected by labeling similarity have "person-nns" or "misc" entity types. Figure 16 depicts the third merging step.

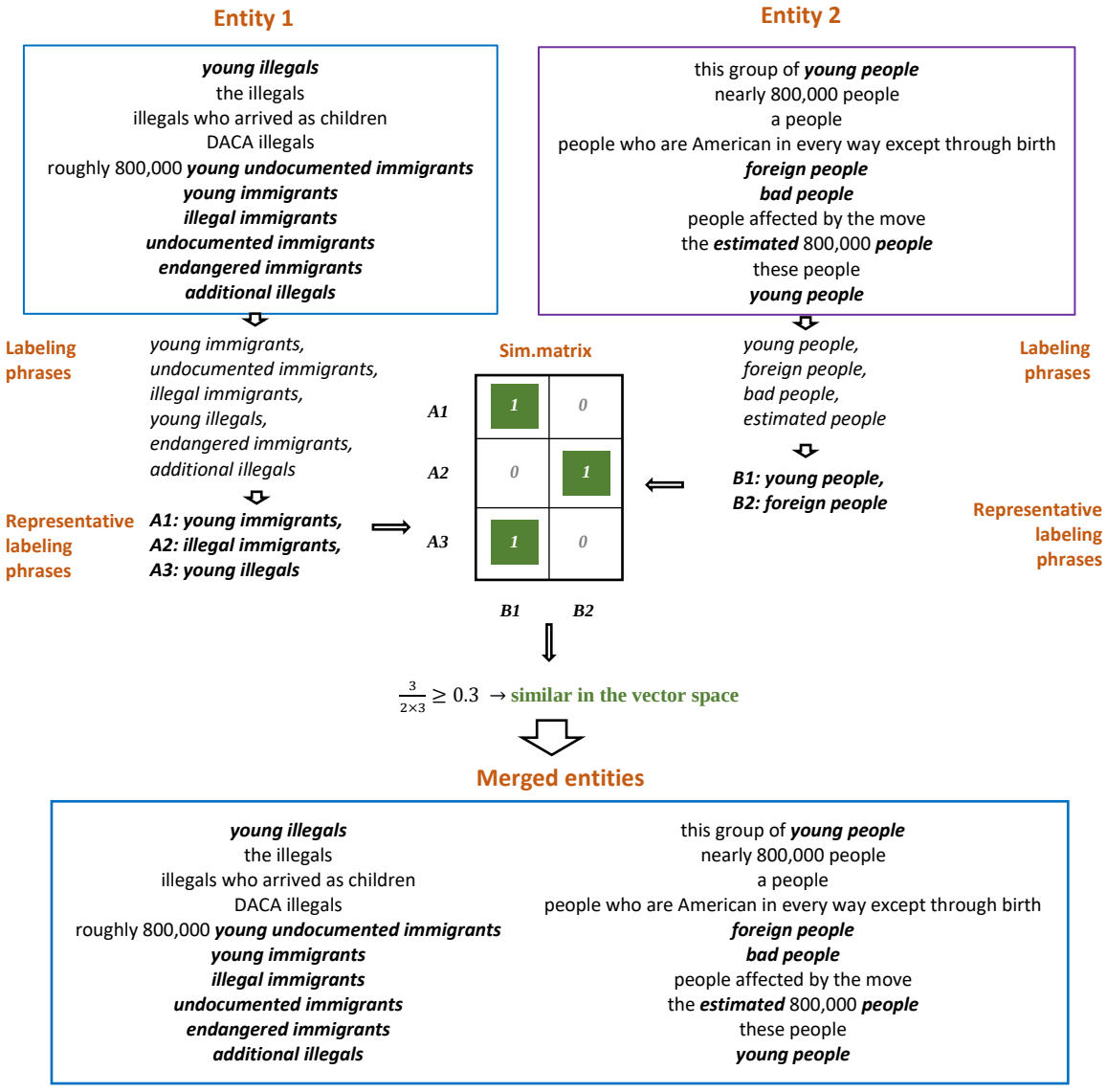


Figure 16: Third step: merging using representative labeling

To compute representative labeling phrases of an entity, we extract unique NPs with adjectival modifiers from the entity members by searching for "amod" relations in a dependency tree of each member. To vectorize each phrase, we first look up if a concatenated labeling phrase, e.g., "illegal immigrant", is a part of the word vector model, and if found, we retrieve phrase's vector, else we calculate a mean value of word vectors retrieved for the constituting words in a labeling phrase. Then, we employ affinity propagation [7] to cluster labeling phrases thus obtaining groups of labeling phrases with similar meaning. Unlike the original way of choosing the cluster centers, we select a representative phrase that has the most global frequent adjective as a representative of the cluster.

Likewise in OPTICS (see Section 4.1), we determine the similarity as a two-step procedure: first, we compute a similarity matrix that consolidates the results of the pairwise comparisons among representative labeling phrases of two entities, and, second, we aggregate values in the calculated matrix to make a final decision about the degree of similarity between two entities. Moreover, similarly to OPTICS and its two thresholds – epsilon and epsilon-prime – we use two different thresholds t_3 and $t_{3,\mathrm{m}}$ to decide whether to merge two entities or not.

As in the second merging step (see Section 4.6), we calculate similarity values for the comparable entities e_0 and e_1, i.e., $\mathrm{ctable}_3[e_0.type][e_1.type] > 0$. Then, we compute a *similarity matrix* $S(V,W)$ spanned by the representative labeling phrases $v_i \in V$ of e_0, $w_j \in W$ of e_1, and $|V| \times |W| \geq 2$. We require the similarity matrix to be at least a vector to minimize the number of possible false positives, i.e., some entities can be merged as outlier-entities rather than truly semantically related entities. Afterwards, for each cell $s_{i,j}$ in $S(V,W)$, we define a three-class similarity score:

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

where $\mathrm{cossim}(\overrightarrow{v_i}, \overrightarrow{w_j})$ is the cosine similarity of both vectors, t_3 is a type-to-type threshold $t_3 = \mathrm{ctable}_3[e_0.type][e_1.type]$, and $\mathrm{t}_{3,\mathrm{r}} = 0.2$ is a reward for more similar vectors to underline the significant similarity by the highest similarity class. We found the three-class score yields better results than using the cosine similarity directly.

Conceptually, we merge e_0 and e_1 if the number of semantically similar representative labeling phrases' pairs is sufficient to consider two entities similar. Therefore, we compute a similarity value $\mathrm{sim}(V,W)$:

$$\mathrm{sim}(V,W) = \frac{\sum_{s \in S} s_{i,j}}{|V||W|} \geq t_{3,\mathrm{m}} = 0.3 \quad (3)$$

Analogously to DBSCAN and OPTICS, before we proceed with comparing V with the remaining entities, we recursively check whether already merged entity W has similar entities and merge if any of such entities determined. The idea is to find entities that are transitively similar to V, and if there is an entity U and $\mathrm{sim}(V,U) < t_{3,\mathrm{m}}$, but if U ∼ W and V ∼ W, then we say U ∼ W, V ∼ W $\xrightarrow{yields}$ $U \sim V$, i.e., U is transitively similar to V, and merge both candidates U and W into V.

4.8. Merging using compound phrases

Merging using *compound phrases* is the fourth merging method that, likewise the merge using representative labeling (Section 4.7), identifies similarity based on the core meaning modifiers. In the fourth merging step, we are looking for parts of the multiword expressions, specifically noun-to-noun compound phrases, e.g., "a staff meeting," "US soldier," and "President Trump." We extract compound phrases by retrieving all "compound" relations from the entity members' dependency trees. Merging using compound phrases consists of two types of methods: *compound-headword match* and *common compounds method*. The methods cover different cases of compound phrases but executed in the order mentioned above within the current merging step. Figure 17 and Figure 18 depict the methods respectively.

Compound-headword match merges entities with the lexical identity of individual words between entities. Specifically, we merge two entities if an entity e_0 contains at least one phrase's head that is a dependent word in at least one compound phrase of an entity e_1 and if e_0 and e_1 are comparable according to ctable_4. The methods work well to merge proper nouns with their titles.

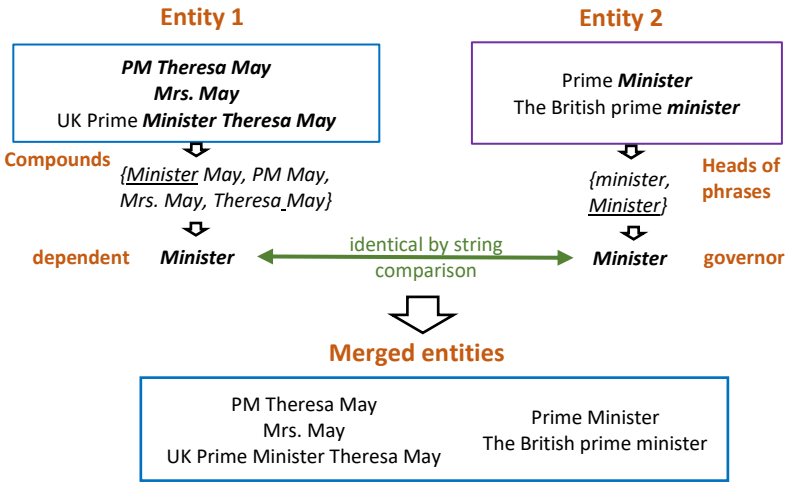


Figure 17: Fourth step: merging using compound-headword match

Common compounds method merges two entities if they have at least one identical compound and the compound phrases containing the shared compound are semantically similar in the vector space. Unlike adjectival modifiers, noun-to-noun compound phrases contain their main meaning stress on the left side of a phrase in most of the cases [35], e.g., in a phrase "staff meeting" a "staff" comprised more meaning than "meeting" thus specifying the type of meeting. Such a meaning shift requires more sophisticated methods to account this shift rather than a mean vector of the constituting words. Therefore, for now, we focused only on the compound phrases where a dependent word is an NE, e.g., "US soldier."

To obtain compound phrases of an entity, we retrieve those compound NPs that have NE as a dependent word. If two entities have at least one compound in common, then we vectorize each compound NP in the same way as the labeling phrases: first, we try to retrieve the entire phrase

from the word vector model, but if it is an out-of-vocabulary (OOV) phrase, then, calculate a mean vector of the phrase.

We compute a similarity matrix $S(V, W)$ spanned by the common compounds $v_i \in V$ of e_0, $w_j \in W$ of e_1, $|V| \times |W| \geq 2$, where e_0 and e_1 are comparable. With parameters $t_{4,r} = 0.2$, $t_{4,m} = 0.3$ and $t_4 = \text{ctable}_4[e_0.type][e_1.type]$, we follow the same procedure as described for the third merging step to calculate a similarity matrix S (2), a similarity value $\text{sim}(V, W)$(3), and, finally, merge two entities if (3) holds. Afterwards, we check if V is transitively similar to other entities.

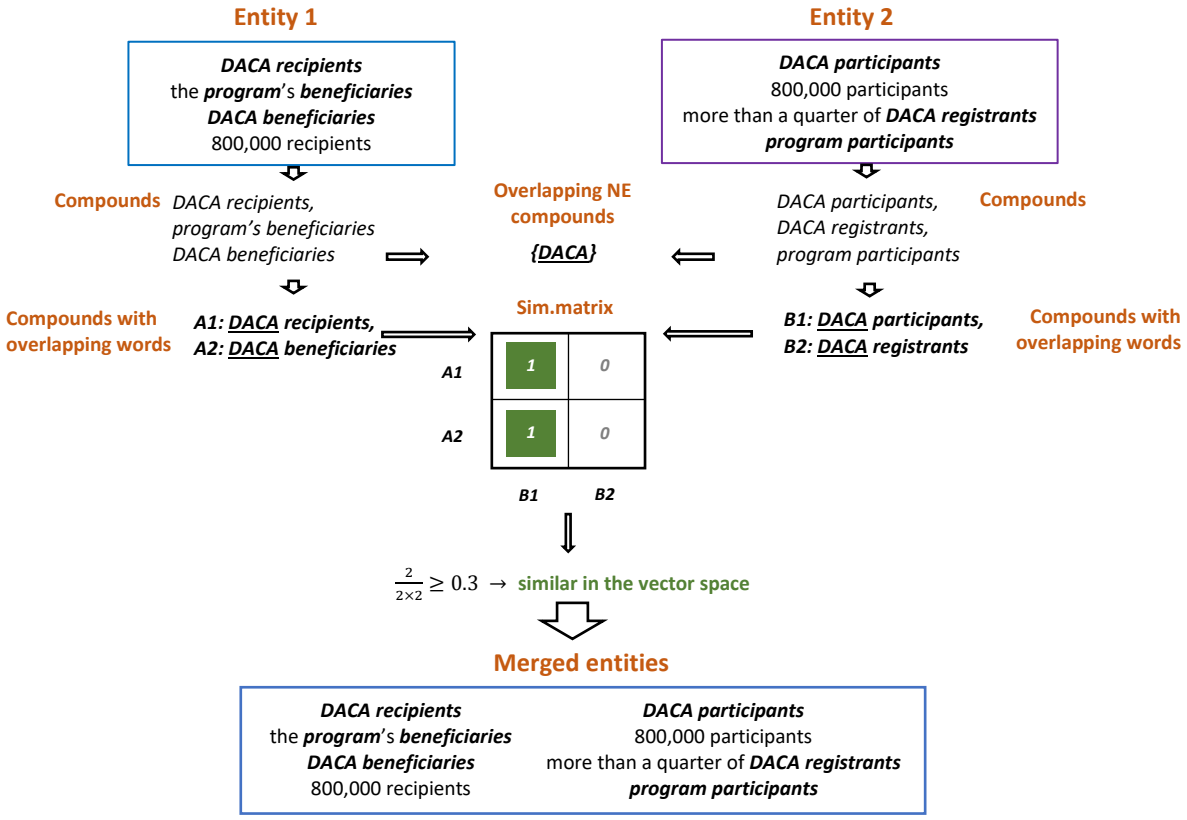


Figure 18: Fourth step: merging with common compounds

4.9. Merging using representative frequent wordsets

The goal of merging using *representative frequent wordsets* is to represent an entity with the most frequent word patterns appearing throughout the entity members. A *wordset* is an itemset where items are words, and the order of the words is ignored. We consider a frequent wordset an attribute that comprises the meaning distributed over the entity members into a representative composite entity meaning. Figure 19 depicts the fifth merging step.

To calculate frequent wordsets, we remove stopwords from each entity member, convert each phrase into a set of words, and employ Apriori algorithm [5] with minimum support $\text{supp} = 4$ to extract frequent wordsets. Additionally, among all the frequent itemsets, we select only maximal

itemsets [5]. To select the most representative wordsets from all maximal itemsets, we introduce *a representativeness score* $\mathrm{r}(w)$:

$$\mathrm{r}(w) = \log\big(1 + \mathrm{l}(w)\big) \times \log\big(\mathrm{f}(w)\big) \qquad (4)$$

where w is the current itemset, $l(w)$ the number of words in the itemset, and $f(w)$ the frequency of the itemset in the current entity, i.e., the calculated support of an itemset.

The representativeness score balances two factors: first, the *descriptiveness of a wordset*, i.e., the more words an itemset contains, the more comprehensively it describes its meaning; second, *the importance of a wordset*, i.e., the more often the itemset occurs in phrases of the entity, the more relevant the itemset is. We then select as the representative wordsets the N itemsets with the highest representativeness score, where $N = \min\left(6, f_p(e)\right)$ and $f_p\ (e)$ is the number of entity members. If a word appears in more than $\mathrm{rs}_5 = 0.9$ of all entity members but is not present as a separate itemset among all the maximal itemsets, then we select only $N - 1$ representative wordsets and add a one-item wordset with the frequently occurring word to the representative wordsets.

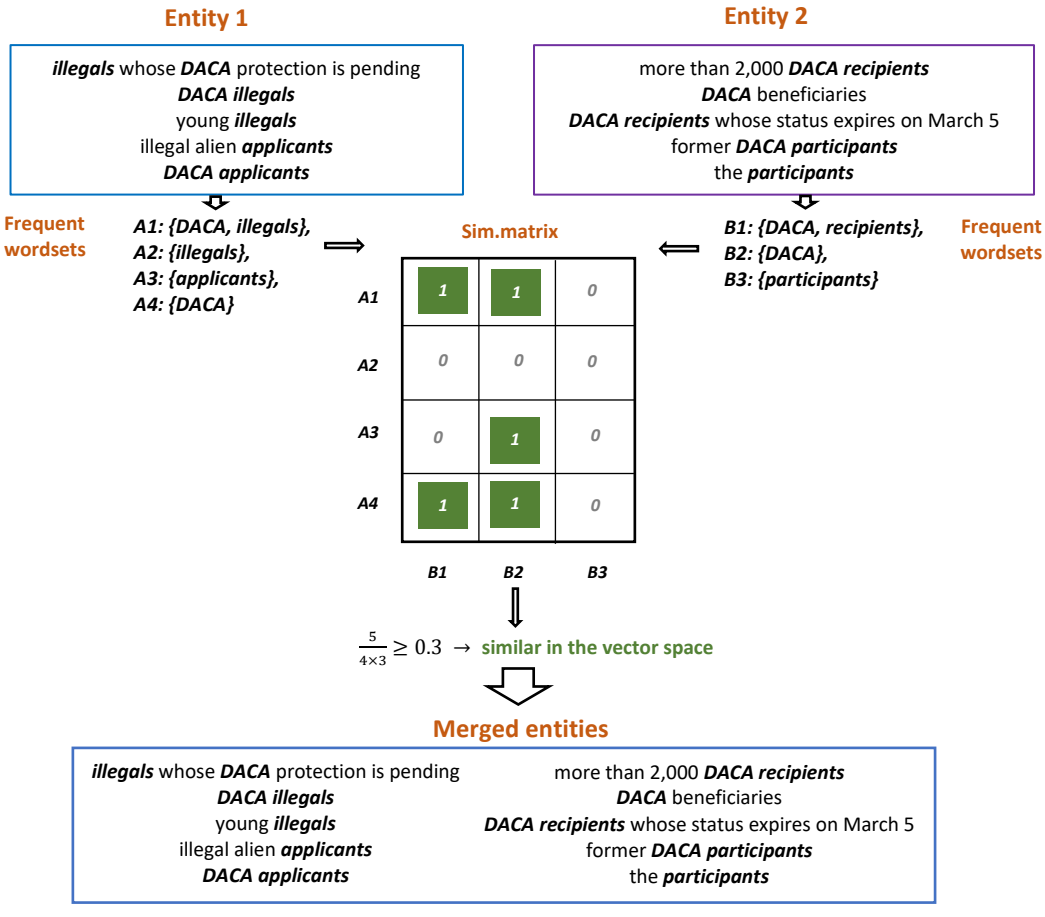


Figure 19: Fifth step: merging using representative frequent wordsets

Before we proceed with merging, we try to find phrases in the model vocabulary by looking up different combinations of words within representative wordsets. The goal is to retrieve vectors that cover the specific meaning of a phrase better than a calculated vector as a combination of the meaning of the constituting words. The meaning of multiword expressions (MWE) can be different from the composition of the meanings of multiple words [51], e.g., an MWE "United_States" found in the word vector model is not located in the same vector space position as the word composition "United+States." Moreover, the MWE "United_States" is situated much closer in the vector space to "U.S." than "United+States." If an MWE is a part of the word vector model, we retrieve a vector of the MWE, then vectorize all other words in a representative wordset and then compute the mean vector of the wordset. If an MWE is not a part of the model, then we vectorize all words and compute a mean vector.

Then, to determine the similarity of two entities, we compute a similarity matrix $S(V, W)$ spanned by the representative wordsets $v_i \in V$ of e_0, $w_j \in W$ of e_1, $|V| \times |W| \geq 1$, where e_0 and e_1 are comparable. Unlike second and third merging steps, V and W can by 1×1 matrixes if the only frequent wordset contains one word or found in the vector model MWE, e.g., "United_States" and "U.S.". With parameters $t_{5,r} = 0.2$, $t_{5,m} = 0.3$ and $t_5 = \text{ctable}_5[e_0.type][e_1.type]$, we repeat the same calculations from the third merging step (Section 4.7) to obtain values for the similarity matrix S (2), a similarity value $\text{sim}(V, W)$(3) and, finally, we merge two entities if the condition of (3) is fulfilled. Before proceeding with comparing V to all other entities, similarly to the third merge step, we first check if V is transitively similar to other entities through W.

4.10. Merging using representative frequent phrases

Representative frequent phrases are conceptually similar to representative frequent wordsets (Section 4.9), but instead of retrieving frequent word patterns regardless of the word order, frequent phrases account the word order. The sixth merging step targets MWEs, especially NEs, where two MWEs have a different set of headwords and a word order property was a missing component to merge MWEs at the fifth step. Figure 20 depicts the sixth merging step.

To calculate frequent phrases, we, first, remove stopwords from the entity members' phrases. Then, we iterate over the list of preprocessed phrases, determine the intersecting phrases, and calculate the number of their occurrence. The procedure is similar to the Information Retrieval (IR) operation of querying intersecting postings for two terms [49], but unlike the IR approach, we do not retrieve all intersecting words but intersecting sequences of words. Then, to choose representative phrases, we calculate a representativeness score of a phrase p similar to the score (4) introduced in Section 4.9:

$$\text{r}(p) = \log\big(1 + \text{l}(p)\big) \times \log\big(\text{f}(p)\big) \quad (5)$$

where $\text{l}(p)$ is a number of words in a phrase p and $\text{f}(p)$ the frequency of p in the entity. We then select as the representative frequent phrases the N phrases with the highest representative score, where $N = \min(6, \text{f}_\text{p}(e))$.

Then, to determine the similarity of two entities e_0 and e_1 in the sixth merge step, we compute a similarity matrix (V, W) spanned by the representative phrases $v_i \in V$ of e_0, $w_j \in W$ of e_1, $|V| \times |W| \geq 2$, where e_0 and e_1 are comparable. For each cell we compute:

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

where $\text{levend}(v_i, w_j)$ is a normalized Levenshtein distance [34][49], $t_{6,r} = 0.2$, and $t_6 = \text{ctable}_6[e_0.type][e_1.type]$.

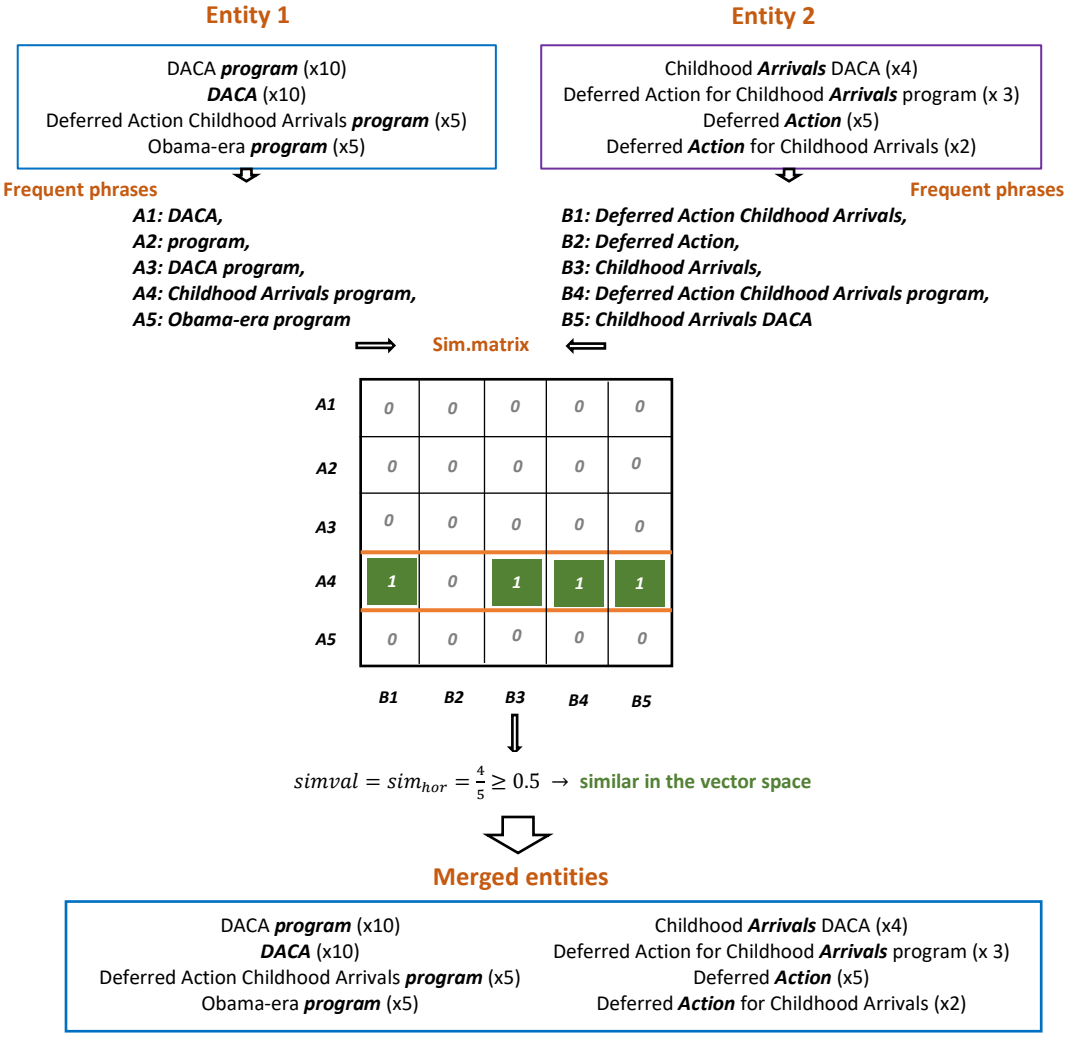


Figure 20: Sixth step: merging using representative frequent phrases

Then, over all rows j we find the maximum sum of similarity scores sim_{hor}, and likewise sim_{vert} over all columns i:

$$\mathrm{sim}_{\mathrm{hor}} = \max_{0 \le i < |W|} \left(\sum_{j=0}^{|V|} s_{i,j} \right) / |W| \quad (7)$$

$$\mathrm{sim}_{\mathrm{vert}} = \max_{0 \le j < |V|} \left(\sum_{i=0}^{|W|} s_{i,j} \right) / |V| \quad (8)$$

We compute a similarity score for the matrix:

$$\mathrm{simval}(V, W) = \begin{cases} \mathrm{sim}_{\mathrm{hor}}, \text{if } \mathrm{sim}_{\mathrm{hor}} \ge \mathrm{sim}_{\mathrm{vert}} \wedge |W| > 1 \\ \mathrm{sim}_{\mathrm{vert}}, \text{else if } |V| > 1 \\ 0, \text{else} \end{cases} \quad (9)$$

Finally, we merge entities e_0 and e_1 if $\mathrm{simval}(V, W) \ge t_{6,m} = 0.5$. If entities $U_1, \dots, U_k$ are transitively similar to V through W (see Section 4.7), then we merge these entities to V as well.

4.11. Summary

In Chapter 4, we presented the multi-step merging approach that compares and merges entities not only by one criterion, e.g., semantic similarity of mean vectorized words of the comprised phrases, but first represents each entity with a set of attributes, which focus on specific aggregated properties of entity members, and then use these attributes for different merging steps.

The proposed MSMA starts with the high entity granularity reducing the number of entities by collapsing similar entities, therefore, minimizing the number of entities at the comparison for each following merging step and increasing the algorithm performance (Section 4.2). Moreover, the introduced entity types decrease the number of comparisons for every merging step from $O(n^2)$ approaching to $O(n \log n)$ depending on the configuration parameters (Section 4.4).

We showed that the complexity of the employed entity attributes proportionally grows with the decreasing granularity of entities. Merging using the simpler core meaning attributes, i.e., phrases' heads, allows accumulating entities with the similar meaning before merging the entities using the more complex attributes, i.e., core meaning modifiers, that, finally, collects enough entity members to merge entities based on the aggregated shared meaning, i.e., frequent word collections. In other words, a reduced number of increased in size entities led to the necessity of representing rather big entities with meaning consolidating attributes.

The proposed approach consists of six consecutive merging steps that use: (1) representative phrases' heads, (2) sets of phrases' heads, (3) representative labeling phrases, (4) compound phrases, (5) representative frequent wordsets, and (6) representative frequent phrases.

Merging using representative phrases' heads (Section 4.5) employs an attribute from the output of coreference resolution and merges NEs by the string comparison thus merging President *Trump* and Donald *Trump*. The method performs well only on NEs and, therefore, we ignore other non-NE entities.

Merging using sets of phrases' heads (Section 4.6) addresses entities' core meaning comprised in the sets of phrases' heads and merges semantically similar sets, e.g., {the *president*, President *Trump*} and {*billionaire*}. The merging step applied to all types of entities. Some words in the sets

of phrases' heads do not contain a specific meaning, i.e., such a meaning that a word vector model could reflect well in the vector space and merge a general-vocabulary entity to the entities with more domain-specific vocabulary.

Merging using representative labeling phrases (Section 4.7) represents each entity with the most prominent adjective-noun NPs contained in its entity members by determining intra- and cross-entity labelling patterns. We managed to merge cases such as "*illegal* immigrants" and "*undocumented* workers." Adjectives belong to the class of core meaning modifiers, but they do not cover all cases of the modifiers.

Merging using compound phrases (Section 4.8) considers the similarity of entities between the noun-to-noun NPs, where the left-side noun is a core meaning modifier, and it is identical for two entities. The merging step determines similarities in cases such as "*DACA* applicant" and "*DACA* recipient", but the method cannot merge longer MWEs where more than two words reflect an entity's meaning.

Merging using representative frequent wordsets (Section 4.9) represents an entity as a collection of frequently used wording patterns incorporated in entity members. After discarding some phrasing fluctuations, we managed to merge entities when extracted "*U.S."* and "*United States."* Because wordsets ignore the order of the words, some word patterns cannot be identified by the fifth merging step.

Merging using representative frequent phrases (Section 4.10) considers the frequent wording pattern of an entity as a sequence of words used throughout the entity members. Considering the word order, we merged long MWEs such as "Deferred Action of *Childhood Arrivals* program" and "*Childhood Arrivals."* The method performs well if entity members contain extensive repetitive wording.

5. Evaluation

Although we implement multiple modules of the WCL analysis system, the evaluation of the system focuses on two aspects: quantitative evaluation of the effectiveness of the proposed MSMA for the candidate alignment task (RT4, Section 5.1) and a case study to demonstrate the functionality of the usability prototype (RT2, Section 5.2). All other modules are considered as the environment and excluded from the evaluation.

5.1. Quantitative evaluation

We perform a quantitative evaluation of the MSMA to assess the effectiveness of the approach. The chapter explains an evaluation experiment setup (Section 5.1.1), gives an overview of the datasets and annotation methodology (Section 5.1.2), introduces evaluation metrics (Section 5.1.3), presents comparison baselines (Section 5.1.4), and reports the performance results obtained in multiple experiments (Sections 5.1.5 – 5.1.8).

5.1.1. Experiment setup

Evaluation of the MSMA aims at estimating of the approach effectiveness by comparing manually annotated concepts to the determined entities. Each candidate phrase has a true label indicating a manually coded concept and a predicted label, i.e., an entity's name that contains this candidate phrase.

Unlike the similar evaluation performed by Hamborg et al. [30], we focus only on the evaluation of the MSMA and control for other factors such as the performance of the candidate extraction. That is, we only evaluate how well the approach categorizes the extracted candidates into the relevant concepts, and we regard that candidate extraction retrieves candidates of sufficient quality and quantity.

The entity extraction module extracts more candidates that are afterward manually annotated as frequent concepts. A concept is considered frequent if then number of phrases referring to it is greater than 1% of the overall number of extracted candidates. To focus only on the relevant candidates, for all evaluation calculations we retain extracted entities that contain at least one manually annotated candidate phrase and exclude the other entities.

When calculating an aggregated F1-score, we compute a weighted average F1-score. We use a support value, i.e., a number of true candidates in a manually annotated concept, to weight the F1-score of each coded concept.

We structure our experiments as follows: first, we evaluate the general performance of the MSMA compared to the baselines (Section 5.1.5), then examine if there is a relation between performance and candidate phrasing complexity employed to refer to the coded concepts (Section 5.1.6). We proceed with observing the performance development regarding the consecutively applied merging steps (Section 5.1.7), and lastly, compare the performance on a topic with a large number of articles to the smaller topic consisting of a subset of the original articles from a bigger dataset (Section 5.1.8).

5.1.2. Dataset overview

To evaluate the MSMA, we used a dataset of eleven topics: we used ten topics from NewSWCL50 dataset introduced in [30] and created an additional eleventh topic, thus obtaining an extended NewsWCL50 dataset for evaluation. Table 3 shows an overview of the dataset. To create the eleventh topic, we followed the article selection principle explained in [30] and selected the equal number of news articles from online news outlets representing the political and ideological spectrum of the US publishers. This way we selected five articles from Breitbart (far right, abbreviation *RR*), Fox News (right, *R*), Washington Post (medium, *M*), CNN (left, *L*), and The New York Times (far left, *LL*) resulting in a topic of twenty-five articles. The articles were selected within the time frame of September 4-6, 2017.

Topics	# articles	# coded phrases
0_CIADirectorMikePompeoMeetingNorthKorea	5	427
1_ComeyMemo	5	419
2_NorthKoreaNuclearStopAnnouncement	5	468
3_DemocratsSueRUTrump	5	580
4_TrumpDealIran	5	413
5_TrumpVisitUnitedKingdom	5	334
6_Asylum-SeekingMigrantCaravan	5	447
7_TrumpDelaysTariff	5	369
8_MuellerQuestionsTrump	5	497
9_Iranfiles	5	383
10_TrumpCancelsDACA25	25	2072

Table 3: Overview of the datasets used for the evaluation of the multi-step merging approach

The coding book for the *content analysis* (CA) of Hamborg et al. [30] describes ten concept types that refer to frequently appearing concepts such as actors, events, actions, etc. The coding book broadly defines target concepts: it allows coding VPs, grouping phrases such as reaction on something into separate concepts, and forming concepts consisting of several individua. In our implementation of the WCL analysis system, we focus only on the entity extraction thus limiting the types of identified semantic concepts.

To evaluate specifically entity identification ignoring more complex concepts, we adapted the CA coding book and created a *simplified CA* coding book. We dropped semantically complex or too abstract concept types and reconsidered concept codes obtained in NewsWCL50. Moreover, as stated in Section 5.1.1, we annotated only extracted candidate phrases to adhere to the evaluation only of the MSMA. We performed the following steps to adapt CA codes from the NewsWCL50:

- Drop complex semantic codes "[…]-I"
- Do not annotate candidates that belong to the complex original concepts such as "Reaction on…";
- Collapse original concept types "Event," "Object," and "Misc" into a simplified concept type "Misc";

- To annotate a concept of a simplified "Country" concept type, use only a country name, country references, and names of organizations and ignore the job positions related to the government organizations;
- If a complex semantic code "[…]-Misc" is similar to "[…]-I", then drop the code;
- If a "[…]-Misc" target concept is related to a specific group of people, e.g., "Democrats," create a simplified coded concept of the "Group" concept type and annotate the candidates with this coded concept;
- If a target concept consists of both NPs and VPs, retain a code name and apply the simplified code to annotate extracted NP-based candidates;
- If a target concept consists of VPs, convert the code into one or multiple simplified NP-based codes. That is, if two originally coded as "Peace negotiation" phrases, "to negotiate about the peace" and "to discuss the end of the war" result in the extracted candidate phrases such as "the peace" and "the end of the war", annotate the extractions with a simplified code "Peace";
- If among the original "[Country]" and "[Country]-I" target concepts there is a group of official representatives that act as one entity, code them as a simplified concept of a "Group" concept type;
- If none of the above is applicable, reuse code of the original target concept;
- In simplified CA, annotate only frequent concepts, i.e., concepts that consist of phrases which number is at least 1% of all extracted candidates.

To code the extracted candidate phrases, we followed a two-step procedure. First, we converted the original codes of all topics to simplified CA following the described conversion rules. Second, we annotated the extracted candidates. To do so, we created lists of all phrases' heads and exemplary phrases for the new simplified codes, which we obtained from the originally coded phrases. We annotated the extracted phrases if they matched phrases' heads or were semantically similar to exemplary phrases. For example, two codes have identical phrase's head, e.g., "meeting," but one phrase is "Inter-Korean *meeting*" and the second one "Trump and Kim's *meeting*." According to the exemplary phrases, the phrases will be annotated as two different codes.

Table 3 shows the number of annotations per topic, and Appendix A1 gives an overview of all codes of CA and simplifies CA at the comparison.

5.1.3. Metrics

Precision, recall, and F1-score are the state-of-the-art evaluation metrics for coreference resolution borrowed from IR [9]. We selected three types of metrics to evaluate the MSMA: relevance metric (precision, recall, and F1-score), clustering quality metric (homogeneity, completeness, and V-measure), and phrasing complexity metric (WCL-metric). We describe each metric type in this section.

Precision, recall, and *F1-score* are state-of-the-art IR metrics to evaluate the relevance of the extracted information and especially extracted coreferential chains [49]. We use F1-score to assess the quality of the *best matching entity* (BME), i.e., an entity with the majority of coded concept phrases. In other words, we want to estimate how well a BME represents a coded concept.

Given the candidate alignment task, we specify the usual meaning of precision and recall. The meaning of precision is the level of representativeness of a BME, i.e., the proportion of coded concept's phrases to all phrases of the BME, and the meaning of recall is the level of completeness of the BME, i.e., the proportion of the phrases in a considered coded concept to those merged into the BME. Figure 21 depicts a confusion matrix for the candidate alignment task. In the evaluation, we allow multiple coded concepts belong to one BME.

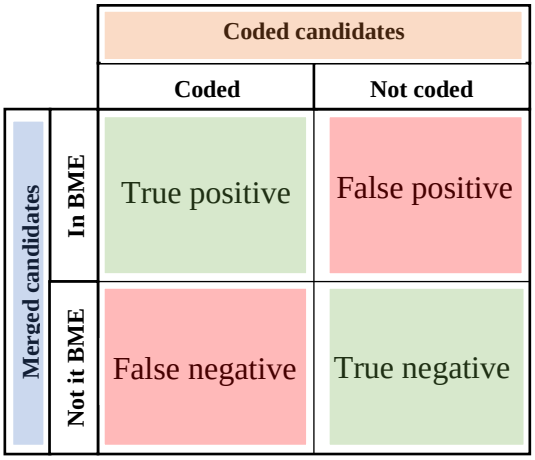


Figure 21: Confusion matrix for the candidate alignment task: the evaluation of correctly merged entities is based on the BME ignoring other smaller entities of the candidate of the same coded concept

We use *homogeneity*, *completeness*, and *v-measure* as a supportive metric to F1 to estimate the overall degree of clustering quality if the algorithm yields not one representative BMEs, but multiple smaller entities [47]. Homogeneity estimates how homogeneous a cluster is and what is a class composition within a cluster. That is, high homogeneity means that points of only one class belong to a cluster. Completeness measures how complete a cluster is, and across how many clusters points of one class were distributed. Thereby, high completeness means that the points of one class are concentrated in a small number of clusters. Figure 22 depicts the principles of homogeneity and completeness. V-measure is a harmonic mean of homogeneity and completeness and shows how well a clustering algorithm performs.

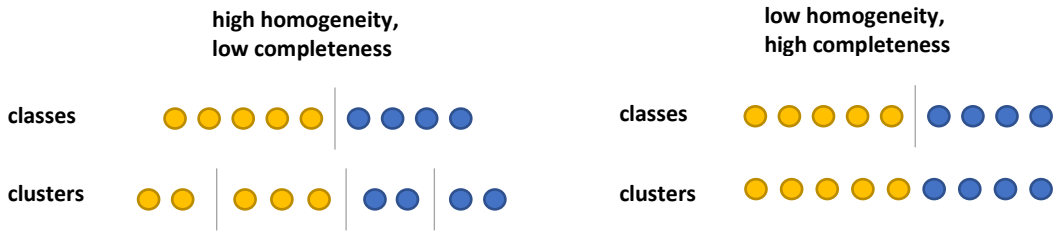


Figure 22: Illustration of principles of homogeneity and completeness

To measure phrasing complexity of a coded concept, we introduce a *WCL-metric* that seeks to capture the variety of word choice of the phrases comprised in a concept. To the best of our knowledge, such phrasing complexity metric is proposed for the first time. The phrasing diversity is based on the number of distinct phrases' heads and takes into account phrasing fluctuation, e.g., labeling:

$$WCL = \sum_{h \in H} \frac{|S_h|}{|L_h|} \quad (10)$$

where H is a set of phrases' heads in a code, S_h is a set of unique phrases with a phrase's head h, and L_h is a list of non-unique phrases with a phrase's head h.

Another way to estimate phrasing diversity of a coded concept is to count the initial number of entities of which each coded concept consisted: the larger the initial number of entities, the higher the phrasing complexity. After an entity-preprocessing step, entities referring to NEs are partially grouped by coreference resolution. Coreference resolution results in fewer initial entities of NE-based concept codes than concept codes referring to more complex non-NE-based concepts.

To evaluate the correctness of the proposed WCL-metric, we test a hypothesis that the more complex concepts, i.e., concepts that have a larger number of entities over which a coded concept was initially distributed, also have higher phrasing complexity consolidated in the WCL-metric. We examine if there is a linear regression between the initial number of entities and the WCL-metric and if any, calculate a coefficient of determination of a model R^2 [21] to measure the quality of the linear model. For every coded concept in the dataset, we extracted a number of initial entities and calculated a WCL-metric. Figure 23 depicts a positive linear trend between two variables; Pearson's correlation coefficient [4] is equal to 0.458 confirming the positive trend observed on the plot. Calculated $R^2 = 0.733$ shows that the model explains 73% of the total variability of WCL-metric and, therefore, has a good fit. The linear model confirms that WCL-metric can be used the explain the complexity of the phrasing used in the coded concepts.

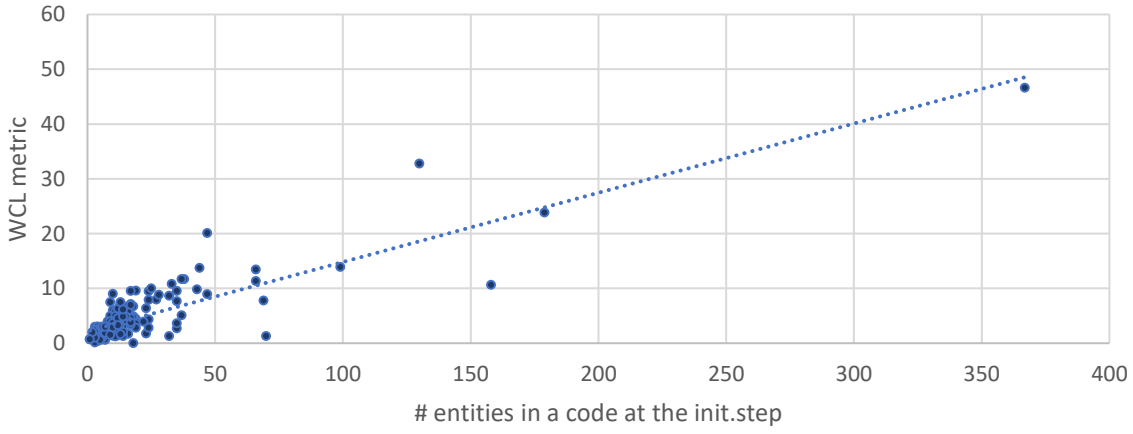


Figure 23: Positive linear relation between the initial number of entities and WCL-metric of phrasing complexity

5.1.4. Baselines

We compare the MSMA to three baselines: *random baseline* (B1), *coreference resolution baseline* (B2), and *clustering baseline* (B3). A random baseline is random guessing of coded concepts, coreference resolution is a state-of-the-art CoreNLP coreference resolution [11], and clustering baseline is one of the first implementations of candidate alignment task that clusters NPs in the word vector space [29].

For the random guessing, we uniformly assign concept codes to all extracted candidates. For the coreference resolution, we use only coreferential groups of candidates and ignore the candidates extracted as additional NPs.

We performed clustering on all extracted candidates by vectorizing the phrases in the word vector space. We removed stopwords from each phrase, retrieved word vectors for each word in a phrase, and then obtained a final vector by averaging the retrieved vectors. We applied affinity propagation [7] to cluster the vectorized candidates, and considered the obtained clusters as entities.

5.1.5. F1-score results

Figure 22 depicts the evaluation results of the MSMA compared to the three baselines and shows that the overall F1 score of the proposed approach $F1_M = 0.84$ is twice as high as the best performing baseline B3 $F1_{B3} = 0.42$ and almost thrice as high as coreference resolution $F1_{B2} = 0.27$. The random baseline B1 performs the worst $F1_{B1} = 0.12$, meaning that random guessing could not address the candidate alignment task.

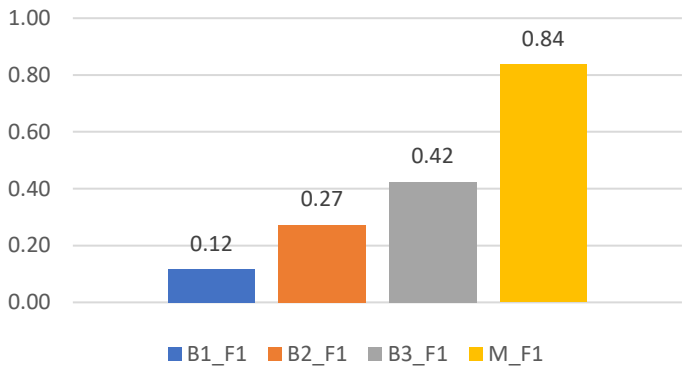


Figure 24: Comparison of the F1-score of the multi-step merging approach to the baselines: the multi-step merging approach outperforms the best performing baseline by 100%

Figure 25 and Table 4 show that the proposed approach outperforms the best performing baseline B3 in all concept types and differs the most compared to B3 in "Actor" and "Group" concept types by $\Delta_{F1\_Actor} = 0.52$ and $\Delta_{F1\_Group} = 0.49$ respectively.

Although coreference resolution is integrated into the merging approach, Figure 25 demonstrates that the highest performance of coreference resolution baseline B2 in "Actor" and "Country" concept types does not systematically yield high performance of the MSMA. Both B2 baseline and the MSMA perform the best in the "Actor" concept type. The highest F1-score of B2 $F1_{B2\_Actor} = 0.41$ yield the highest F1-score of the approach $F1_{M\_Actor} = 0.97$. On the contrary, despite baseline B2 performing the second-best on the "Country" concept type $F1_{B2\_Country} = 0.30$, the proposed approach performs the worst on this concept type $F1_{M\_Country} = 0.74$. Both baselines B2 and B3 performs the worst on the "Group" concept type $F1_{B2\_Group} = 0.14$ and $F1_{B3\_Group} = 0.29$ but the MSMA demonstrates significant performance improvement $F1_{M\_Group} = 0.78$.

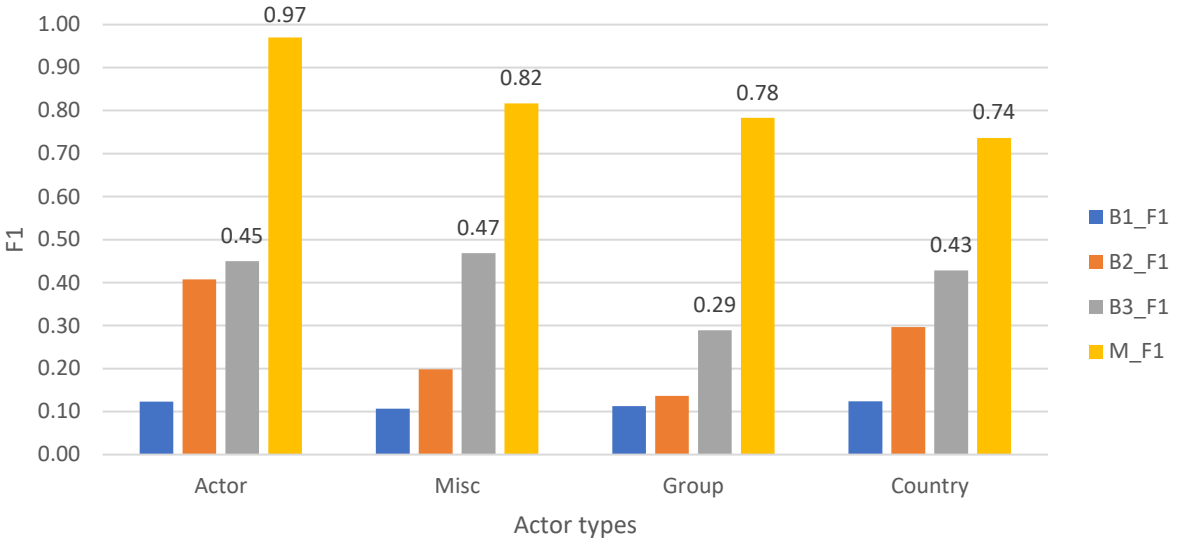


Figure 25: Performance on the different concept types: all concept types outperform the best performing baseline

	Baselines			Multi-step merging			
Actor type	B1_F1	B2_F1	B3_F1	Precision	Recall	F1	Support
Actor	0.12	***0.41***	**0.45**	0.97	0.97	**0.97**	1894
Misc	0.11	0.20	0.47	0.89	0.80	0.82	1955
Group	0.11	0.14	**0.29**	0.82	0.83	**0.78**	1037
Country	0.12	***0.30***	0.43	0.93	0.66	0.74	1523

Table 4: Performance on the different concept types: all concept types outperform the best performing baseline

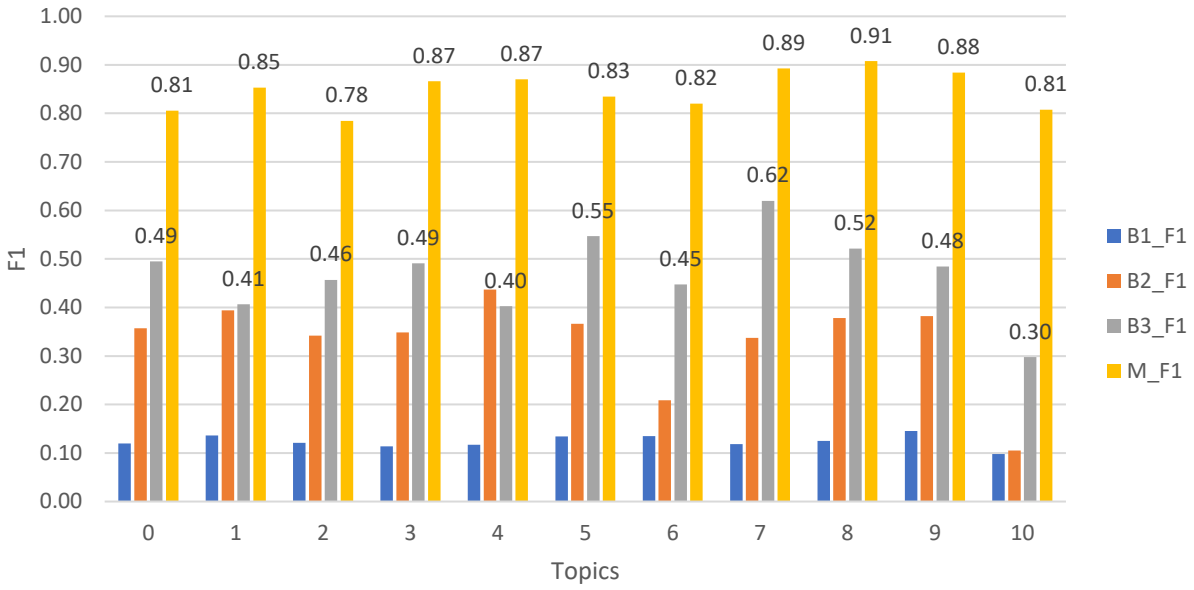


Figure 26: Performance on different topics: all topics outperform the best performing baseline

	Baselines			Multi-step merging			
Topic	B1 F1	B2 F1	B3 F1	Precision	Recall	F1	Support
0	0.12	0.36	0.49	0.81	0.88	0.81	427
1	0.14	0.39	0.41	0.93	0.84	0.85	419
2	0.12	0.34	0.46	0.82	0.82	**0.78**	468
3	0.11	0.35	0.49	0.95	0.84	0.87	580
4	0.12	0.44	0.40	0.96	0.84	0.87	413
5	0.13	0.37	0.55	0.86	0.86	0.83	334
6	0.13	0.21	0.45	0.98	0.73	0.82	447
7	0.12	0.34	0.62	0.94	0.87	0.89	369
8	0.12	0.38	0.52	0.97	0.87	**0.91**	497
9	0.15	0.38	0.48	0.99	0.84	0.88	383
10	**0.10**	**0.11**	**0.30**	0.90	0.80	**0.81**	2072

Table 5: Performance details on different topics: all topics outperform the best performing baseline

Figure 26 and Table 5 compare the performance across the topics and show that the MSMA outperforms the best performing baseline B3 on all topics. The approach performs the best on topic 8 $F1_8 = 0.91$ and the worst on topic 2 $F1_2 = 0.78$. The largest by the number of articles topic 10 performs slightly worse but comparable to an average F1-score ($F1_{M\_10} = 0.81$ against $F1_{avg} = 0.84$) though having the lowest performing baselines among all topics $F1_{B2\_10} = 0.11$ and $F1_{B3\_10} = 0.3$.

5.1.6. F1 results from the perspective of WCL complexity

The goal of WCL analysis is to resolve coreferential phrases of a broader sense, i.e., the phrases that cannot be resolved with the state-of-the-art methods of coreference resolution or NER. The following experiment is designed (1) to test if a definition of broadly phrased and abstract concepts is related to the WCL complexity of different concept types and (2) to test how the WCL complexity of different concept types is related to the model performance.

In Section 5.1.3, we introduced a WCL-metric and showed that this metric describes a coded concept from the perspective of anaphora phrasing complexity. We proved that the phrasing complexity depends on the degree of coreference resolution involved in the pre-merging step. Coded concepts consisting of fewer coreferential groups form coded concepts with more diverse phrasing. Concept type "Misc" refers to abstract concepts such as event, object or action whereas "Group" concept type refers to a group of people with broad phrasing. Figure *27* (left) illustrates that "Misc" and "Group" have the highest WCL complexity $WCL_{Misc} = 5.67$ and $WCL_{Group} = 9.20$. Figure *27* (right) depicts a decreasing logarithm trend between the WCL-metric and F1-score: the higher the WCL-metric leads to the lower the performance, i.e., is harder to identify. The highest F1-score $F1_{Actor} = 0.97$ corresponds to the lowest WCL complexity of "Actor" concept type $WCL_{Actor} = 2.1$, whereas the lowest F1-score $F1_{Group} = 0.78$ belongs to the highest WCL metric of "Group" $WCL_{Group} = 9.20$.

Actor Type	WCL	F1	Symbol
Actor	2.10	0.97	
Country	4.49	0.74	
Misc	**5.67**	0.82	
Group	**9.20**	0.78	

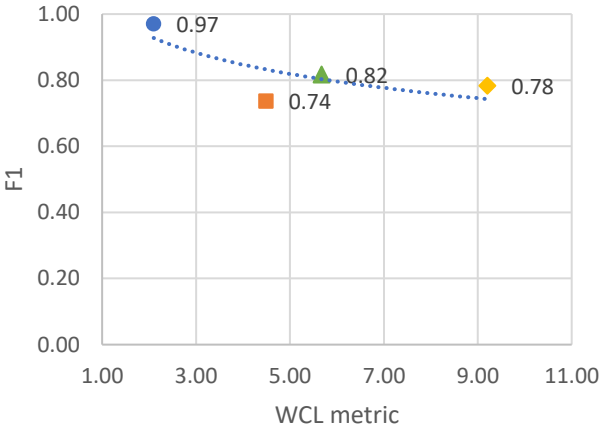


Figure 27: Dependency of performance from WCL-metric from a concept type perspective: a decreasing logarithm trend between WCL metric and F1-score

Similarly, we test if the same relation holds between the algorithm performance and WCL-complexity level across the topics. Figure 28 depicts the logarithm trend between these variables. Though not as notable as in Figure *27*, we observe a decreasing tendency. Topics 6 and 10 with the highest WCL complexity $WCL_6 = 8.37$ and $WCL_{10} = 12.71$ yield performances $F1_6 = 0.82$ and $F1_{10} = 0.81$, which are comparable to the average F1-score $F1_{avg} = 0.84$.

Topic	WCL	F1	Symbol
8	2.84	0.91	
7	2.89	0.89	
5	3.31	0.83	
4	3.54	0.87	
1	3.63	0.85	
3	3.95	0.87	
0	3.99	0.81	
9	4.63	0.88	
2	5.44	0.78	
6	**8.37**	**0.82**	
10	**12.71**	**0.81**	

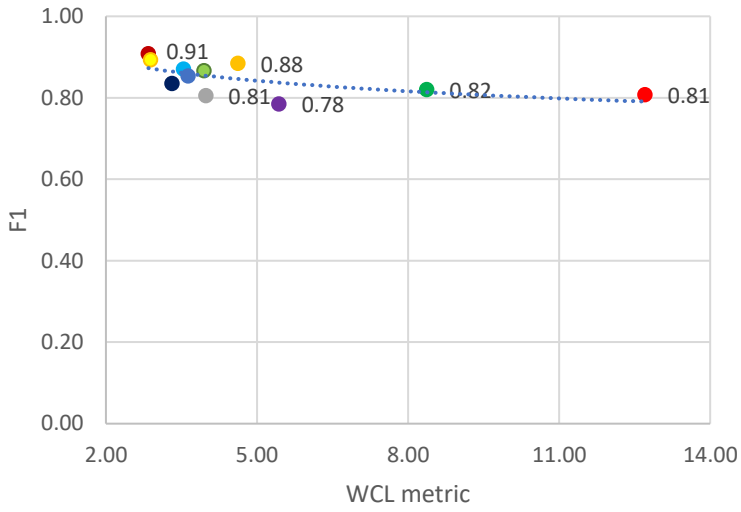


Figure 28: Dependency of performance from WCL-metric from a topic perspective: the topics with the highest WCL value perform comparably to the average F1-score

5.1.7. Performance of the merging steps

Performance of the MSMA consists of the sum of impacts of the consecutive merging steps, which search for similarities between entities given specific attributes. While Section 5.1.5 covers a comparison of the overall performance of the approach compared to the baselines, the current section focuses on performance development at each merging step.

To evaluate the merging steps, we calculate F1-score and v-measure (see Section 5.1.3). The difference between two metrics is that F1-score evaluates the approach focusing on the quality of BMEs, whereas v-measure assesses general clustering quality based on all identified entities. In other words, if a coded concept is distributed over multiple entities of equal size, F1-score of this coded concept will be low, but the v-measure will be higher thus showing the generally acceptable quality of the determined entities.

Table 6 shows that precision, recall, and F1-score of the initial merging step are comparable to the baseline B2 and the first merging step outperforms the best performing baseline B3 in both F1-score and v-measure. F1-score of the MSMA rises from 0.276 to 0.838 ($\Delta_{F1\_M} = 0.562$), rapidly boosting on the first steps, which focus on the core meaning, and then gradually increasing with the more advanced merging steps. The increase of V-measure is not that significant. V-measure changes from 0.632 to 0.840 ($\Delta_{V\_M} = 0.208$) because at the initial state coreferential chains yielded high completeness and entity preprocessing resulted in high homogeneity.

	Effectiveness			Clustering quality		
	Precision	***Recall***	***F1***	***Homogeneity***	***Completeness***	***V-measure***
B1	0.117	0.149	0.117	0.089	0.080	0.084
B2	0.968	0.173	0.273	0.682	**0.477**	0.560
B3	0.865	0.324	**0.424**	0.809	0.487	**0.604**
Init.	0.983	0.175	0.276	0.983	**0.468**	0.632
Step 1	0.982	0.439	**0.535**	0.979	0.548	**0.698**
Step 2	0.948	0.688	0.755	0.924	0.713	0.803
Step 3	0.929	0.759	0.802	0.917	0.730	0.812
Step 4	0.925	0.791	0.818	0.911	0.758	0.826
Step 5	0.915	0.820	0.835	0.903	0.785	0.839
Step 6	0.914	0.824	0.838	0.903	0.788	0.840

Table 6: Effectiveness and clustering quality of merging steps: starting the first merging step the multi-step merging approach outperforms the best performing baseline

The previously observed general trend in the performance difference in the various concept types presumes contrasting performance of the concept types at the merging steps as well. Table 7 shows that the initial step outperforms baseline B2 not evenly: the performance in "Group" and "Misc" types increased by 0.003 and 0.005 respectively compared to B2 while on the other concept types the increase is only 0.001. Step 1 performs better than a baseline B3 on NE-based "Actor" and "Country" types, and step 2 outperforms the baseline on non-NE-based "Misc" and "Group" concept types. The "Group" concept type shows the biggest performance boost $\Delta_{Group} = 0.643$ whereas the "Country" type demonstrates the smallest performance boost $\Delta_{Country} = 0.439$.

Steps	**Actor**	**Country**	**Misc**	**Group**
B1	0.123	0.124	0.107	0.112
B2	0.407	0.297	0.198	0.137
B3	0.450	0.428	0.468	0.289
Init.	0.408	0.298	0.204	0.140
Step 1	0.872	0.634	0.298	0.222
Step 2	0.927	0.685	0.779	0.502
Step 3	0.927	0.685	0.803	0.744
Step 4	0.970	0.700	0.803	0.744
Step 5	0.970	0.736	0.808	0.783
Step 6	0.970	0.736	0.817	0.783

Table 7: Increase of performance with each merging step across concept types

Table 8 shows the development of F1-score with each merging step on different topics. We observe the most significant performance increase on topics 6 and 10 by $\Delta_6 = 0.607$ and $\Delta_{10} = 0.702$. Having the lowest performance at the initial step, the final performance on these topics is

comparable to the average. Topic 8 is the best performing topic $F1_8 = 0.908$ and topic 2 performs the worst $F1_2 = 0.785$.

	Topics										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
B1	0.119	0.136	0.121	0.113	0.117	0.134	0.135	0.118	0.125	0.146	0.098
B2	0.357	0.394	0.342	0.349	0.437	0.367	**0.209**	0.337	0.378	0.382	**0.105**
B3	0.495	0.407	0.457	0.491	0.403	0.547	0.447	0.619	0.521	0.485	0.297
Init.	0.367	0.394	0.351	0.352	0.444	0.371	**0.213**	0.338	0.378	0.382	**0.105**
Step 1	0.648	0.592	0.538	0.530	0.705	0.630	0.313	0.574	0.670	0.667	0.436
Step 2	0.746	0.835	***0.738***	0.773	0.859	0.718	0.586	***0.860***	0.883	0.854	0.695
Step 3	0.748	0.845	***0.738***	0.780	0.862	0.718	0.746	***0.860***	0.883	0.854	0.799
Step 4	0.794	0.850	0.775	0.793	***0.870***	0.795	0.750	0.885	0.903	0.868	0.800
Step 5	0.805	0.853	0.774	0.866	***0.870***	0.835	0.820	0.893	0.908	0.884	***0.801***
Step 6	0.805	0.853	0.785	0.866	***0.870***	0.835	0.820	0.893	0.908	0.884	***0.807***

Table 8: Increase of *F1-score* with merging steps

When analyzing the results of the F1 score in Table 8, some merging steps seem to not affect the performance results, e.g., Step 3 of topics 2 and 3. Because F1-score evaluates performance only of BMEs, the metric does not reveal any change if smaller relevant to coded concept entities mere merged with themselves and not with a BME. Table 9 shows that completeness acts as a supportive metric to highlight the effectiveness of the merging step also when the merging does not yield an instant performance improvement. For example, the completeness on topics 2 and 7 shows intermediate clustering improvement at Step 2, whereas topics 4 and 10 demonstrate the increase of completeness at the steps from 4 to 6. While conducting the experiments, we observed that the merging steps tend to form smaller entities and later merge them into BMEs.

	Topics										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
B1	0.125	0.077	0.107	0.089	0.085	0.080	0.045	0.105	0.094	0.060	0.015
B2	0.516	0.482	0.513	0.495	0.494	0.481	0.361	0.536	0.529	0.498	0.345
B3	0.527	0.446	0.504	0.529	0.435	0.555	0.408	0.632	0.494	0.467	0.354
Init.	0.513	0.472	0.495	0.487	0.495	0.488	0.350	0.511	0.522	0.484	0.334
Step 1	0.620	0.531	0.550	0.550	0.599	0.572	0.370	0.585	0.640	0.607	0.403
Step 2	0.738	0.720	**0.665**	0.705	0.760	0.706	0.541	**0.784**	0.827	0.781	0.616
Step 3	0.742	0.734	**0.667**	0.716	0.765	0.706	0.608	**0.788**	0.827	0.781	0.701
Step 4	0.800	0.743	0.695	0.736	**0.785**	0.777	0.616	0.816	0.859	0.806	0.708
Step 5	0.827	0.746	0.718	0.792	**0.787**	0.813	0.671	0.834	0.876	0.839	**0.738**
Step 6	0.827	0.746	0.723	0.792	**0.789**	0.813	0.671	0.834	0.876	0.839	**0.757**

Table 9: Increase of *completeness* with merging steps

5.1.8. Big vs. small dataset analysis

Among the small five-article topics of the dataset, topic 10 consists of twenty-five news articles. This topic was created as an addon to the NewsWCL50 dataset to evaluate the performance of the MSMA on a larger topic compared to the performance on the smaller topics. As discussed in Sections 5.1.5 – 5.1.7, the performance on the topic 10 is slightly worse but comparable to the average F1-score, but the performance development from the initial pre-merging step to the sixth step is 25% higher than the average performance development.

We derived the conclusions obtained in the previous sections based on comparison to the topics with the different concept type composition, i.e., the concept types were not always matching, and each concept type was represented by a different number of included concepts. As a result, we lacked proper experimental setup to conclude that the larger topics perform somewhat worse than the smaller topics.

To fulfill a requirement of the appropriate experiment setup, we extract three subset topics of five articles from the big original topic of twenty-five articles. While a comparison of a big topic to one subset topic does not seem to be sufficient and evaluation of all 5^5 subsets is not feasible, we extract three subsets and average their performance. We created three subsets with stratified sampling [4]. That is, we selected with uniform distribution one article per an ideology group and ensured that all concepts covered in the twenty-five-article topic are present in the subset topics. Then we executed the MSMA on all subset topics with the same run configuration parameters used to execute the big topic. Finally, we calculated a mean F1-score of the three subsets.

Table 10 shows that F1-score of the big topic is higher than a mean F1-score of the three small topics $F1_{big} = 0.81 \gg F1_{avg\ small} = 0.72$. When analyzing the results across the concept types, we see that the results are identical on "Actor" type, F1-score of the small topics is higher on "Country" type, and the performance of the subset topics is smaller on "Group" and "Misc" types.

Concept type	F1 score				
	Random 5			Random 5: mean	All 25
	1	2	3		
Actor	0.94	0.94	1.00	0.96	0.96
Misc	0.64	0.62	*0.75*	***0.67***	***0.88***
Group	*0.73*	0.54	0.62	***0.63***	***0.75***
Country	0.65	0.67	0.65	0.66	0.59
All	0.73	0.68	0.77	**0.72**	**0.81**

Table 10: Evaluation results of an original DACA25 dataset compared to the DACA5 subsets

5.2. A case study on the usability prototype

To enable interactive exploration of the obtained entities, in Section 3.6 we introduced a developed usability prototype that includes a visualization with four views. In this section, we

describe two cases demonstrating how to use the visualization and explore phrasing composition of the determined entities.

5.2.1. Script 1: an exploration of the phrasing complexity

WCL-metric is a proposed metric to estimate complexity of the phrasing referring to an entity. Since the WCL analysis resolves phrases of a broader meaning, the metric plays a role of interestingness criterion and hence suggests structuring exploration of the entities in the order of decreasing phrasing diversity.

Figure 29 depicts a bar chart view where entities are sorted in the decreasing order of their size. “Caravan” entity has the highest WCL value among the entities and, given the order of the entities, is also the largest hence the most prominent entity in topic 6. To obtain details on phrasing composition of the entity, we switch to the candidate view by clicking on the entity’s bar or name.

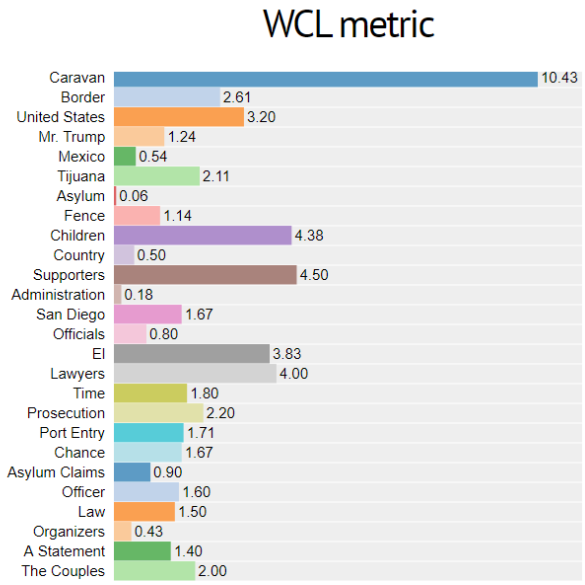
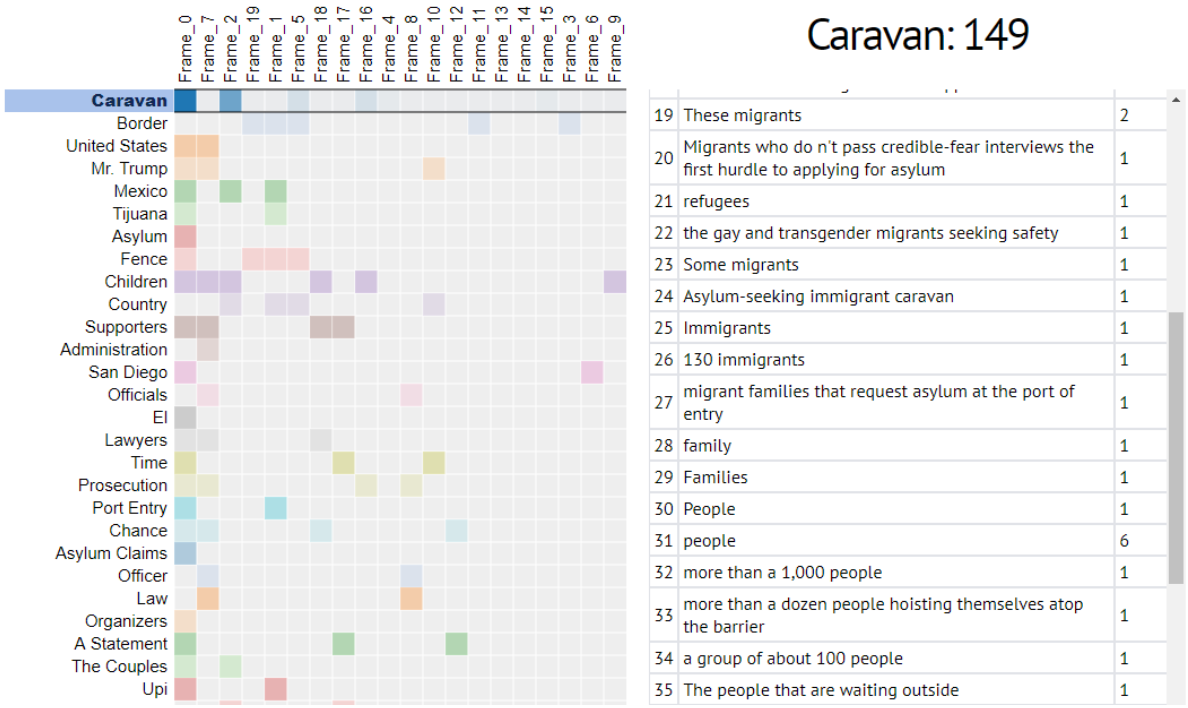


Figure 29: Visual exploration of the results starting with the highest WCL-metric

Activated candidate view in the selection mode, depicted in Figure 30, lists the entity members and the number of their occurrence in the topic. For example, the view shows that throughout the authors of the article when describing or referring to the “Caravan” entity, use phrases as “asylum-seeking immigrant caravan”, “families”, “refugees”, “the gay and transgender migrants seeking safety”, etc.

Figure 31 depicts an article view with highlighted candidates of the selected entity. While switching between articles using the arrows on the both sides of an article title, users can read and explore the candidates’ context and compare various word choices used by different outlets.



19	These migrants	2
20	Migrants who do n't pass credible-fear interviews the first hurdle to applying for asylum	1
21	refugees	1
22	the gay and transgender migrants seeking safety	1
23	Some migrants	1
24	Asylum-seeking immigrant caravan	1
25	Immigrants	1
26	130 immigrants	1
27	migrant families that request asylum at the port of entry	1
28	family	1
29	Families	1
30	People	1
31	people	6
32	more than a 1,000 people	1
33	more than a dozen people hoisting themselves atop the barrier	1
34	a group of about 100 people	1
35	The people that are waiting outside	1

Figure 30: Matrix and candidate view when exploring details on “Caravan” entity

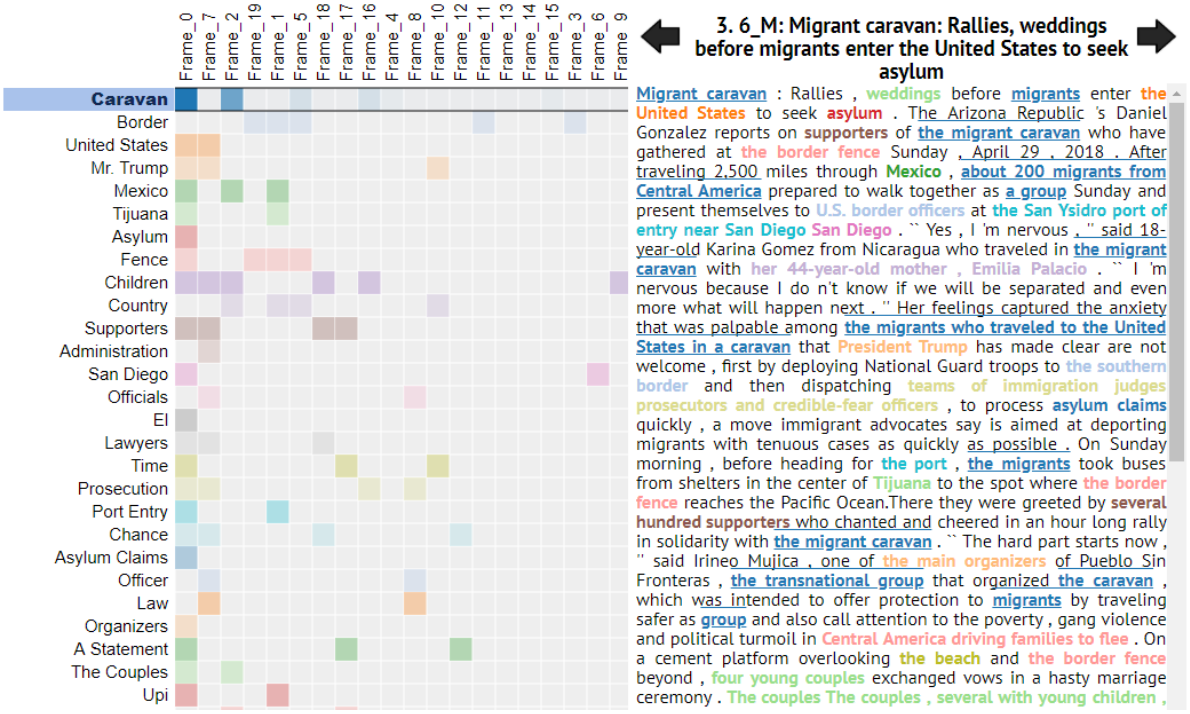
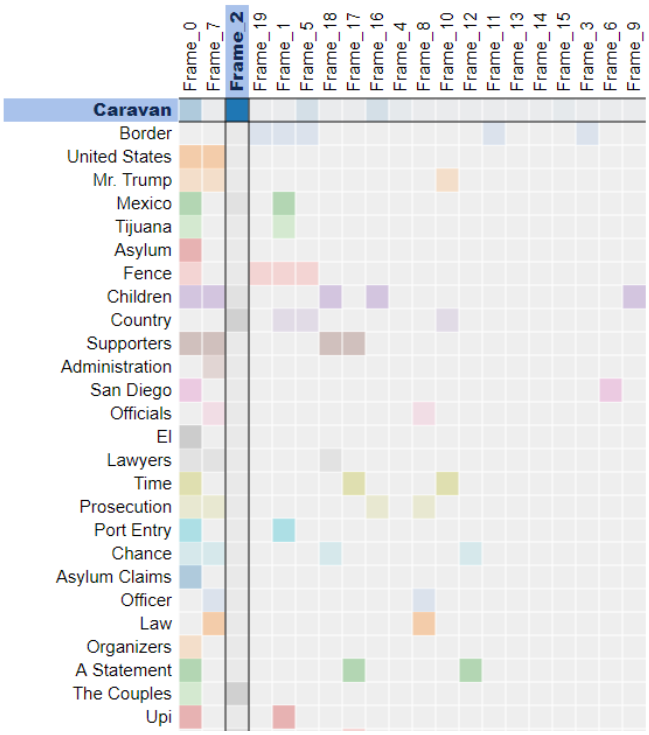


Figure 31: Matrix and article view when exploring details on “Caravan” entity



Caravan » Frame_2

#	Candidate	Number
1	a group	1
2	a group of about 150 of the caravan 's members	1
3	The 150 migrants , many of whom were children ,	1
4	the group	2
5	the same group as the Washington-based non-profit that provides computer classes to international residents of the District of Columbia	1
6	Migrants who do n't pass credible-fear interviews the first hurdle to applying for asylum	1
7	The people that are waiting outside	1
8	the past 24-hours several groups	1
9	ranking member of the House Committee on Homeland Security	1
10	the nonprofit group	1
11	members	1
12	the two groups	1
13	these large `` Caravans '' of people	1
14	Similar migrant groups	1
15	a particular group	1
16	this group	1

Figure 32: Matrix and candidate view of “Caravan” entity members framed as “Frame_2”

In the selection mode, the matrix view encodes the number of phrases with opacity thus allowing to identify the most frequently occurred emotion frames. Assuming that “Frame_0” is a neutral frame, we proceed exploring the second biggest emotionally similar group of candidate phrases denoted as “Frame_2”.

Figure 32 illustrates matrix and candidate views in the double-selection mode with the selected “Caravan” entity and emotion “Frame_2”. “Frame_2” assembles candidate phrases with word choice related to different groups, e.g., “the people”, “members”, “migrant groups”, etc. By clicking on another colored cell of the “Caravan” entity, we inspect a different group of similarly framed candidates of “Frame_16”. This time, the frame groups candidates related to family, e.g., phrases such as “migrant families that request asylum”, “their families”, etc. In general, we found LIWC dimensions not yielding insightful results and being hard to interpret and consider using another emotion dimensions in the future work, e.g., SEANCE [13].

A click the selected cell or a right-click anywhere on the matrix view resets the matrix and article views and returns the candidate view to the bar chart view. We continue exploring the remaining entities in decreasing order of WCL-metric and decreasing order of entity size hence inspecting the remaining most interesting WCL cases.

5.2.2. Script 2: from a phrase to an entity

The article view converts the texts of news articles from the plain view to a hypertext filled with interlinked colored referenced phrases belonging to the entities. If a phrase is an entity member, it will have a color code of its entity thus allowing to estimate coreferential phrases visually.

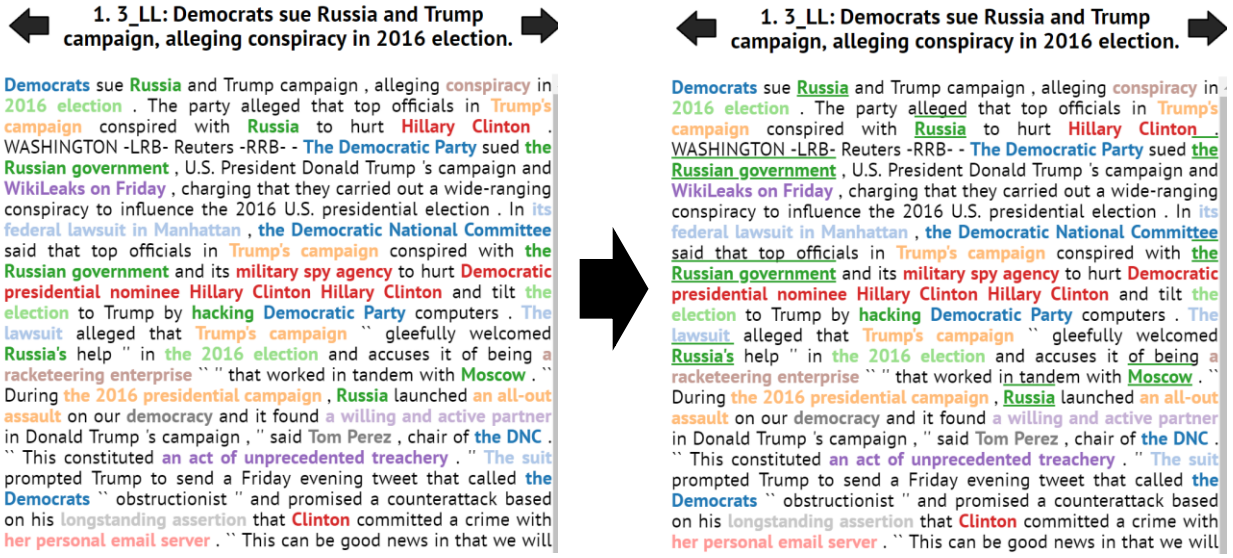


Figure 33: Selection of a phrase in the article view to explore all phrases related to the same entity

If while reading a text of a news article, a user finds a phrase interesting for investigation, then by clicking on a phrase, he or she highlights all other candidates that constitute an entity. Figure 33 depicts the selection of a “Russia” phrase to highlight in the text phrases such as “the Russian government” and “Moscow”. Selection of a phrase activates an entity selection mode on the matrix view and changes a bar plot view to a candidate view to enable further exploration of the entity composition.

6. Discussion and future work

In the following chapter, we discuss the evaluation results from Sections 5.1.5 – 5.1.8, then summarize observations and findings of the MSMA, and, finally, conclude the section with the future work.

6.1. Discussion

The MSMA resulted in the F1-score $F1 = 0.84$, which is $\Delta = 0.42$ higher than its best performing baseline, i.e., clustering of candidate phrases in the word vector space. Among the most prominent results, the evaluation showed that performance on a concept type or a topic depends on the phrasing diversity of included concepts, i.e., the more different phrases are used to refer to a concept, the harder it is to resolve all concept mentions. We revealed that low performance on a "Group" concept type happened due to the small number of articles in the topic. The performance on the "Group" concept type on the bigger topic of twenty-five articles is on average $\Delta = 0.12$ higher that on the smaller topics of five articles ($F1_{Group\_big} = 0.75$ and $F1_{Group\_small} = 0.63$). We concluded that to increase the MSMA's performance on the diversely phrased concepts, we require a larger number of articles to capture repetitive phrasing patterns used across different publishers.

In the following subsections, we discuss the evaluation results and explore the underlying reasons for the different outcomes. We start with the performance analysis across concept types and their influence on the performance on the topics (Section 6.1.1), then we discuss the difference between broadly defined concepts and concepts with diverse phrasing (Section 6.1.2), proceed with the investigation of the performance difference between big and small topics (Section 6.1.3), and, finally, discuss mixed concepts problem (Section 6.1.4). We conclude the discussion section with interesting findings and examples of the determined entities by the WCL analysis system (Section 6.1.5).

6.1.1. Performance on different concept types

While the overall performance indicates the general quality of the approach, the bigger interest of the approach performance lies in the evaluation of the approach given different concept types. In this subsection, we discuss the performance on different concept types regarding their level of abstractness and phrasing complexity. Moreover, we demonstrate that topic composition influences the algorithm performance of the approach, and conclude that the performance on the most phrasing-diverse topics is comparable to the average performance ($F1_6 = 0.82$ and $F1_{10} = 0.81$ compared to $F1_{avg} = 0.84$).

In Section 5.1.6, we showed that the introduced WCL-metric could be used as a numeric interestingness criterion to estimate which coded concept are the hardest to identify in the WCL analysis. Abstract and broadly phrased concepts of "Misc" and "Group" types resulted in having the highest WCL complexity among four concept types. By showing a negative logarithmic trend

between the WCL-metric and the MSMA's performance, we proved that these concept types are the most complicated for identification semantics groups.

In Section 5.1.7, we observed that the effectiveness of the merging steps differs depending on the concept type origins, i.e., if contained concepts are mainly NE- or non-NE-based. The initial pre-merging step slightly outperforms the second-best performing baseline B2, which represents state-of-the-art coreference resolution. It happens due to the added NPs, and the higher performance difference is noticed in "Misc" and "Group" concept types. The added NPs show that these concept types contain very few coreferential phrases and can be represented only by additionally extracted NPs.

In Section 5.1.7, we observed that the first merging step outperforms the best performing baseline B3 on "Group" and "Country" concept types, whereas only the second merging step outperforms on the remaining concept types. Such behavior is caused by the comparison tables incorporated into the merging steps. Each comparison table determines the comparability of different entity types and results in merging candidate phrases belonging to specific concept types. This way Step 1 is only applied to the NE-based entity types, thus outperforming the baseline B3 on "Actor" and "Country" concept types. On the contrary, applied to all entity types, Step 2 results in outperforming on the non-NE-based concept types.

Also, in Section 5.1.5, we saw the baselines B2 and B3 perform the worst on the "Group" type and stated that the MSMA gained the second-highest performance increase on this concept type. Whereas the lowest performance value of baseline B2 indicates that the "Group" concept type is not a target of state-of-the-art coreference resolution, the lowest F1-score of the baseline B3 suggests that finding similarity on the average word vector of the entire phrase does not capture specific enough information, which is sufficient to determine the similarity between entities. In contrast, the MSMA extracts attributes targeting specific phrases' properties to compare entities on the extracted attributes.

When comparing the performance of the merging steps in Section 5.1.7, we observed that not only the initial step performs the worst on topics 6 and 10, but also the approach results in the highest performance boost on these topics. Also, we identified that the approach outperformed the baseline B3 only on the second merging step, unlike other topics.

To investigate the reasons for such performance contrast, we examined concept composition of these topics represented in Appendix A1. The examination revealed dominating concepts related to immigrants, which have the highest WCL complexity among all concepts (32.81 and 46.61 in topics 6 and 10 respectively). Both concepts with high WCL complexity belong to the "Group" type. This finding co-aligns with the general performance trend in the "Group" type. In Section 5.1.6, we observed that performance on topics 6 and 10 stepped out of the trend of model performance depending on WCL complexity. Slightly worse than average, the F1-score of both topics showed a significant increase in WCL complexity. The high performance on the topics with high WCL complexity is the goal of the WCL analysis, and we consider the results of the current WCL analysis system satisfactory.

Considering the best and worst performing topics in Section 5.1.6 and Section 5.1.7, we observed that the model performs the best on the topic 8 with the lowest WCL complexity among all topics. The model performs the worst on the topic 2 that contains a concept of "Misc" type with

the highest WCL complexity $WCL_{Misc} = 20.15$ among all "Misc" concepts. The low performance can be related to the broadly defined concept. We discuss this problem in the following Section 6.1.2.

To sum up, the experiments showed that the WCL-metric could be used to describe phrasing complexity used to refer to a coded concept. The topics with high WCL complexity indicate the presence of such complicated concept types as "Group" and "Misc" that represent concepts of abstract nature or are broadly defined. The evaluation showed that performance on the most phrasing-diverse topics is somewhat less than of the average performance ($F1_6 = 0.82$ and $F1_{10} = 0.81$ compared to $F1_{avg} = 0.84$). Moreover, we assume, that both concept types "Group" and "Misc" can be the target of the framing of word choice and labeling [31], hence leading to more biased topic coverage.

6.1.2. Broadly defined concepts vs. concepts with diverse phrasing

While analyzing the lowest performance on "Group" and "Country" concept types, we revealed the different nature of the reasons underlying this low performance. While performance on the "Group" concept type gradually grows with each newly applied merging step and could be improved with more sophisticated merging steps, the performance on "Country" concept type indicates a problem grounding in the basic components of the approach such as word representativeness of the employed word vector model and the semantic complexity of a coded concept.

In Section 5.1.5, we spotted that although coreference resolution is incorporated into the MSMA, it seems to have no proportional influence on the model outcome. The best model performance at the initial step in "Actor" type resulted in the overall best model performance, whereas the second-best performing initial step on the "Country" type yielded the worst model performance.

The approach performs similarly bad both on the "Country" type and the "Group" type ($F1_{Coutry} = 0.74$ and $F1_{Group} = 0.78$), but "Country" type stepped out of the overall logarithmic WCL complexity trend (see Section 5.1.6). Given the different WCL complexity $WCL_{Coutry} = 4.49$ and $WCL_{Group} = 9.2$, we assume distinct reasons underlying the low performance.

In Section 5.1.5, we evaluated precision, recall, and F1-score across the concept types. The evaluation of the "Group" concept type yield balanced weighted precision $P_{Group} = 0.82$ and recall $R_{Group} = 0.83$, but the evaluation of the "Country" concept type result in the second-best precision $P_{Country} = 0.93$ and the smallest among all concept types recall $R_{Country} = 0.66$. Such low recall indicates the approach's inability to merge multiple entities that contain related to the "Country" type candidates.

Table 11 shows the performance increase across concept types and accents the contrast performance boost between "Country" and "Group" types. These concept types have similar performance increase after the merging steps targeting the entity members' core meaning, despite the NE-based nature of the "Country" type that implies performance growth similar to "Actor"

type. The performance increase of the "Group" type almost doubles over the next merging step blocks, whereas the "Country" type gains at most 15% of additional performance increase.

The intended MSMA pipeline implies using the output of the previous merging step to extract more complex attributes to merge similar entities. Each merging step should capture specific properties of the diverse phrasing thus allowing to identify semantic similarities across multiple small entities. Low performance on the concepts with rich phrasing indicates the necessity of the MSMA improvement, specifically, development of the new attributes and new merging steps that could capture wording semantic peculiarities. In contrast, the low performance of the concepts with less phrasing variety indicates another set of problems lying in the basic approach components.

Merging step types	Actor	Country	Misc	Group
Core meaning (Steps 1 & 2)	0.519	**0.388**	0.575	**0.362**
Core modifiers (Steps 3 & 4)	0.043	0.014	0.024	0.242
Word patterns (Steps 5 & 6)	0.000	0.037	0.014	0.039
Overall	0.562	**0.439**	0.613	**0.643**

Table 11: Difference of performance increase of merging steps across concept types: the "Group" concept type got the largest performance increase

During the algorithm design phase, we presumed the overall performance on the "Country" type to be comparable to that on the "Actor" type due to the NE-based nature of both types. The high performance of coreference resolution incorporated in the initial step and the design of the merging steps that target core meaning and word patterns should have yielded the second-best performing concept type. However, during the evaluation phase, we realized that the coding scheme of manual annotation of the "Country" type is too broad for the chosen word vector model.

Rather low WCL complexity and a low increase of the F1-score at the core meaning and word pattern merging steps indicate that the word vector model does not capture semantic similarity between country-related names and terms and the names of the organization. Although the semantic similarity between phrases such as "Congress" and "the U.S." is trivial for a human evaluation and it is easy to annotate these phrases as "USA," the model identifies these candidates as members of two different entities.

Usage of a simpler coding scheme to manually annotate extracted phrases could improve the performance of the "Country" concept type and, consequently, lead to the overall performance improvement. To test this hypothesis, we reannotated topic 10, specifically, we split a "USA" code into two codes: "USA" to code references to the country and "USA\Organization" to annotate governmental institutions. The evaluation of the reannotated topic yielded the performance increase on the "Country" concept type from $F1_{Coutry\_old} = 0.59$ to $F1_{Country\_new} = 0.85$.

Performance comparison of "Country" and "Group" concept types revealed the contrasting reasons underlying the low performance: (1) lack of attributes and merging steps capable of capturing cross-entity peculiarities, (2) not accurate representation of words and phrases in the

semantic vector space, and (3) complex and broadly defined coding scheme. To increase the performance of the MSMA, we need to address at least one of these problems.

6.1.3. Reasons for differences in the performance on big and small topics

The approach performance on a bigger topic exceeds performance on the smaller topic by $\Delta = 0.09$, showing better results on the "Misc" and "Group" concept types. Unlike an expected trend of decreasing performance with increasing WCL complexity, the evaluation results demonstrated that the more diverse repetitive phrasing enables more efficient concept identification on these concept types.

In Section 5.1.8, we compared the performance of a big topic to its subset topics. Due to the high performance of the "Misc" and "Group" concept types, the overall performance of the big topic was higher than the average of its subsets. To investigate the reasons why the MSMA outperforms on "Misc" and "Group" concept types, we calculated WCL-metric for each concept type. For the smaller topics, we averaged the values to obtain a mean WCL-metric value per a concept type.

Figure 34 and Table 12 depict the inverse proportion of WCL-metric compared to F1-score. Aligning with the previous observations from Section 5.1.6, almost equal WCL-metric yields similar F1-score on "Actor" type and bigger value of WCL-metric leads to the lower model performance, as the results of "Country" performance suggests. On the contrary, lower WCL-metric values of "Misc" and "Group" types yield lower F1-scores, hence contradicting our previous conclusions.

We investigate unexpected performance results by looking into the performance details at each merging step shown in Table 13. Table 13 demonstrates that the third merging step, merging using representative labeling phrases, increases the performance on the big topic by $\Delta_{big} = 0.538$ and on the smaller topics on average by only $\Delta_{small} = 0.198$. That is, the third merging step is almost three times more efficient on the big topic rather than on the small ones. Moreover, in half of the "Misc" type concepts (highlighted with bold borders), the third merging step boosts the performance on the big topic yielding the maximum values (shown in bold) and has little or no impact on the performance on the smaller topics.

The high performance of the merging step using representative labeling phrases on the big topic happens due to the larger semantically related variety and repetition of labeling phrases used to refer to the prominent concepts. This repetitive labeling provides additional ground to capture similarity between entities. Table 10 shows that performance on "Group" and "Misc" concept types on topics Random5-1 and Random5-3 respectively reached the performance of the All25 topic. This comparable performance could be obtained only in case of some successful article combinations in a subset topic that could lead to the repetitive enough labeling phrases.

The comparison experiment demonstrated that the approach performance on a topic with a bigger number of articles exceeds the performance on its subset topics, thus showing that the performance of the MSMA does not drop in case of the scaled-up topic volume. The comparison across the concept types showed that MSMA performs the best on the big dataset on the complex

concept types “Group” and “Misc.” These concept types represent the major WCL analysis interest as they cannot be identified by coreference resolution and other related research areas.

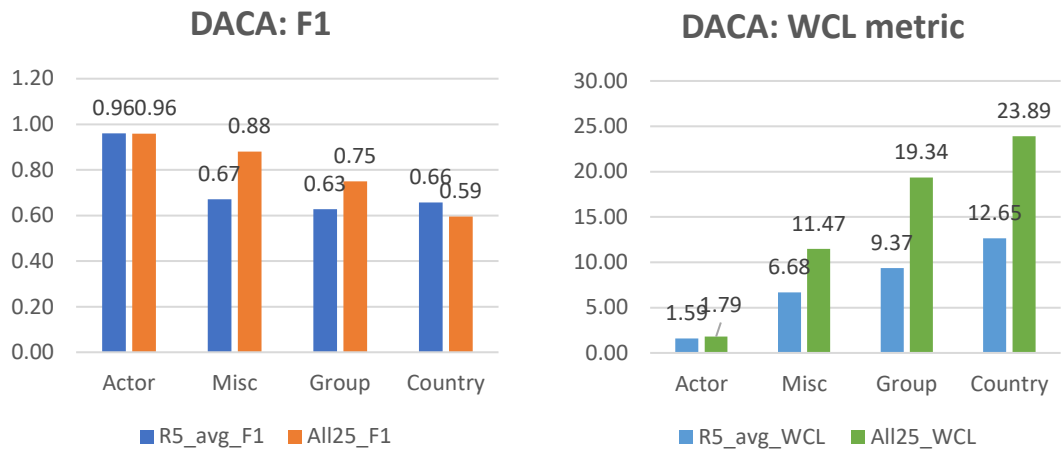


Figure 34: Evaluation results of an original DACA25 dataset compared to the DACA5 subsets: the more diverse WCL of “Misc” and “Group” concept types leads the better performance

Concept type	F1 score					WCL metric				
	Random 5			Random 5: mean	All 25	Random 5			Random 5: mean	All 25
	1	2	3			1	2	3		
Actor	0.94	0.94	1.00	0.96	0.96	1.98	1.69	1.11	1.59	1.79
Misc	0.64	0.62	*0.75*	***0.67***	***0.88***	8.05	6.71	5.28	***6.68***	***11.47***
Group	*0.73*	0.54	0.62	***0.63***	***0.75***	11.06	8.93	8.13	***9.37***	***19.34***
Country	0.65	0.67	0.65	0.66	0.59	11.17	13.58	13.19	12.65	23.89
All	0.73	0.68	0.77	**0.72**	**0.81**	7.80	6.77	5.84	6.8	12.71

Table 12: Evaluation results of an original DACA25 dataset compared to the DACA5 subsets: the more diverse WCL of “Misc” and “Group” concept types leads the better performance

6.1.4. Mixed concepts

The MSMA seeks to locate coreferential phrases of broader sense and unite them into semantically related entities. Given a word embedding model as a language model to represent the meaning of the words in the vector space, the approach merges similar words and phrases based on the representation of the words’ semantics in the word vector model. In Section 6.1.2, we discussed that the approach does not yield merging of the manually annotated phrases in some cases because of the reasons related to the inefficient entity attributes, poor representation of the words in the vector space, or complex coding system.

		Group				Misc			
Topic	**Step**	**Immigrants**	**USA\ Americans**	**USA\ Lawmakers**	**USA\ Officials**	**DACA program**	**DACA status**	**Decision to close DACA**	**New immigr. reform**
All25	Init	0.050	0.098	0.081	0.105	0.073	0.056	0.122	0.055
	Step 1	0.050	0.098	0.526	0.105	0.608	0.056	0.122	0.055
	Step 2	0.283	*0.519*	0.963	*0.676*	0.905	0.449	0.932	0.560
	Step 3	0.821	*0.166*	0.963	*0.155*	0.905	0.692	0.932	0.749
	Step 4	0.821	0.166	0.963	0.155	0.905	0.692	0.932	0.749
	Step 5	0.823	0.162	0.971	0.151	0.905	0.692	0.932	0.749
	Step 6	**0.823**	0.162	**0.971**	0.151	**0.946**	**0.692**	**0.932**	**0.732**
R5-1	Init	0.171	0.200	0.125	0.182	0.250	0.160	0.267	0.188
	Step 1	0.171	0.200	0.636	0.182	0.496	0.160	0.267	0.188
	Step 2	0.338	*0.500*	0.983	0.625	0.622	0.467	0.960	0.533
	Step 3	0.699	*0.136*	0.983	0.625	0.622	0.516	0.960	0.533
	Step 4	0.699	0.136	0.983	0.625	0.622	0.516	0.960	0.533
	Step 5	0.684	0.196	1.000	0.857	0.622	0.516	0.960	0.533
	Step 6	0.684	0.196	1.000	0.857	0.622	0.516	0.960	0.533
R5-2	Init	0.049	0.667	0.190	0.286	0.256	0.200	0.286	0.114
	Step 1	0.049	0.667	0.643	0.286	0.571	0.200	0.286	0.114
	Step 2	0.367	0.500	0.973	0.667	0.649	0.480	0.655	0.596
	Step 3	0.491	0.500	0.973	0.372	0.649	0.480	0.667	0.596
	Step 4	0.491	*0.500*	0.973	0.372	0.649	0.480	0.667	0.596
	Step 5	0.471	*0.098*	1.000	0.314	0.649	0.480	0.667	0.596
	Step 6	0.471	0.098	1.000	0.314	0.649	0.480	0.667	0.596
R5-3	Init	0.091	0.333	0.222	0.222	0.413	0.167	0.364	0.087
	Step 1	0.091	0.333	0.414	0.222	0.413	0.167	0.364	0.087
	Step 2	0.333	*0.800*	0.905	0.571	0.716	0.625	0.957	0.424
	Step 3	0.442	*0.286*	0.905	0.571	0.716	0.625	0.957	0.542
	Step 4	0.442	0.286	0.905	0.444	0.716	0.625	0.957	0.542
	Step 5	0.586	0.195	0.864	0.444	0.716	0.625	0.957	0.542
	Step 6	0.586	0.195	0.864	0.444	0.716	0.625	0.957	0.542

Table 13: Performance of the approach on the big topic vs. its subsets for "Group" and "Misc" concept types (solid black box shows the performance improvement of a big topic over small topics and vice versa)

In contrast to the problem of not merging all entities related to one concept, during the close examination of the evaluation results we revealed cases of erroneously merged entities with similar phrasing. To spot such cases, we need to examine the performance trend of each concept and look for a performance drop at some merging step. For example, Table 13 shows that the performance on the coded concept "USA\Americans" at step 3 plunges from 0.519 to 0.166 compared to step 2 on the "All25" topic. Similarly, the performance on another concept "USA\Officials" also drops from 0.676 to 0.155 at the same merging step. The detailed investigation revealed that these

concepts were merged with the entity referring to the "Immigrants" concept. Although these concepts have a high recall value, the precision value drops in both cases because these falsely merged concepts share the best matching entity with the "Immigrants" concept.

In the described case, such false merging happens either due to similar labeling used within two concepts, e.g., "young immigrants" and "young Americans", or because the labeling phrases are too closed in the vector space, e.g., "undocumented immigrants" and "immigration officials", and the merging step cannot identify their actual dissimilarity. The detailed investigation revealed that some phrasing within the "Immigrants" concept is rather complex: in topic 10, when referring to the former DACA recipients, some articles use phrases like "the parents of American citizens" thereby making the concepts semantically similar and making hard even for a human coder to annotate such phrase with a correct concept code.

Another type of false entity merging is based on the phrases with similar or identical phrases' heads but belonging to different concepts. In topic 0, three concepts are related to meetings or summits: "USA PRK summit", "USA JPN meeting", and "PRK KOR meeting" (see Appendix A1). Although referring to different concepts, the phrases' heads are identical or similar and contain meaning of "meeting", "summit", or "talks". We found it hard to manually annotate phrases related to these concepts and quite often used the phrases' context to disambiguate phrases' meaning. Thereby, we could not expect a current implementation of the MSMA to be capable of differentiating the phrases as different concepts.

6.1.5. Summary

The MSMA performs twice better compared to the best performing baseline B3 ($F1_M = 0.84$ compared to $F1_{B3} = 0.42$). The approach also outperforms the baseline B3 across concept types and topics. In Section 6.1.1, we showed that topic composition influences on the overall topic performance. The topics performing the worst are the topic with dominating "Misc" or "Group" concept types that are the hardest for identification and are the major targets of the WCL analysis. Although, performing the worst, the model differs from the average performance by 10% on these topics.

When inspecting the evaluation results, we uncovered interesting findings among the aligned candidate phrases. For example, referring to one coded concept of DACA recipients (see code 'Immigrants" in topic 10), the following phrases were merged: "undocumented students," "illegal alien applicants," "often dubbed Dreamers," and "these incredible kids." Moreover, when identified, the entities allowed unifying the word choice representation in the articles. Employed entity name as a representative phrase, it helped identify the coupling sentences across various publishers. For example, we extracted "The lengthy statement is among Obama's most forceful since departing office." vs. "Mr. Obama issued a rare public statement in opposition to his successor." and "DNC files a lawsuit over election interference." vs. "DESPERATION: DNC files multi-million-dollar lawsuit against Russia, Trump campaign for collusion." as sentences covering one piece of information with different word choice. Appendix A2 shows more examples of coupling sentences.

The high model performance allows annotating a larger corpus of articles automatically with a so-called "silver standard" quality, i.e., a corpus with less annotation quality compared to the "gold" human annotation [1]. Human coders could easier check the silver dataset and faster correct the mistakenly annotated concepts than annotating new datasets from scratch.

Although we used a word vector model released in 2013 and employed it into the MSMA to resolve coreferential phrases referring to the concepts not present or widely used in the past, e.g., "DACA", "denuclearization", or "President Trump", the word vector model managed to reconstruct the meaning most of these terms used in the news articles dated 2017-2018.

Despite high performance, the MSMA has the following drawbacks: (1) the concepts with similar or identical phrasing are falsely merged together, (2) the chosen word vector model does not treat out-of-vocabulary (OOV) words or represent some phrases correctly, and (3) model parameter optimization depends on the expert domain knowledge. We discuss these disadvantages in the following section.

6.2. Future work

As discussed in Section 6.1.5, the drawbacks of the MSMA are the falsely merging of concepts with similar word choice, inefficient word vector model, and expert-dependent parameter optimization. In this section, we cover the ideas of how to solve these problems, improve the performance of candidate alignment, and discuss the next version of the WCL analysis system.

To address a problem of merging phrases that belong to different concepts, we want to study the methods used for word's sense disambiguation (WSD). The problem of WSD is partially similar to the mixed concepts problem and is employed in the cross-document coreference resolution. In both tasks of concept and word sense disambiguation, a list of candidate phrases or a text corpus is given, but in case of WSD, we need to estimate the word's meaning among those listed in a dictionary, whereas in the mixed concept problem we need to cluster the words without any prior knowledge about the phrases' specific sense. We plan to research the methods of WSD that could be applicable for clustering based on the context similarity. Additionally, context similarity could help resolve topic-specific related phrases such as "Kim Jong Un" and "Little Rocket Man" and enable identification of more complex semantic concepts such as reaction on something or action.

Previously, we have discussed that the chosen Word2Vec model is slightly outdated compared to the news articles release date, which we used in the dataset. Despite being old, the word vector model managed to reconstruct the non-existent or not well covered in the older news concepts. The reconstruction was possible only for the words being a part of the word vector model; the OOV words were discarded. By discarding the OOV words, we lost some phrases' semantics, hence leading to the lower MSMA performance. Additionally, as discussed in Section 6.1.2, the Word2Vec model does not catch all semantic relations incorporated into the coded concepts, e.g., of the "Country" concept type. To solve both problems, we plan to conduct experiments by employing other word embedding models and estimating the best performing model for the WCL analysis task.

Although we suggest a general parameter configuration to execute the model, domain and expert knowledge are required to tune the parameters for the better model performance on a specific topic. To enable automated parameter optimization, we will incorporate supervised learning and train a deductive analysis model given true values of the coded concept.

As an advanced and more automated approach for the candidate alignment task, we consider training a WCL-resolution model based on the sequential neural networks (SNN). To do so, we will need to annotate a big dataset with the MSMA resulting in the silver quality corpus, design features based on the output of the MSMA, and experiment with the SNN architecture. Given a larger annotated corpus, we expect model learn various cases of WCL across multiple topics.

The proposed MSMA can be employed in the applications to assist both readers in the everyday news consumption and the social science researchers to study media bias. In the resonance topics, politicians and journalists use diverse and slanted word choice to refer to the frequent concepts. A news consumer may find it difficult to estimate which narrative phrases refer to which entity. Moreover, when not resolved, these slanted references can trigger emotion evaluation before being consciously identified as bias inducing words. In Section 5.2, we demonstrated that highlighted interlinked concept mentions help a reader to pay attention to the phrasing diversity. If a news aggregator, when presenting the texts of topically related articles, incorporates highlighting of the phrases that refer to the most diverse covered concepts, the users will become aware of the bias by word choice and labeling. Conscious reading of the variously framed articles allows making an inform decision instead of emotional reaction on the news content.

The proposed approach embedded into the media bias analysis systems could allow social sciences to speed up their research by drastically decreasing the amount of manual text annotation. The researchers could focus on the more specific tasks such as origin exploration of the slanted narrative mentions, the evolution of such phrases, and study the effect on the reader's perception. The future work includes evaluation of the WCL analysis system with the social science researchers to test the usability of the tool for the research in practice. The feedback collection will ensure the tool applicability and relevance not only in the computer science research of media bias, but also in the applied research political and media studies.

7. Conclusion

The vast majority of the published news articles exhibit so-called media bias, i.e., present the content of the news differently, thus leading to the distorted perception of the information. The information distortion can appear when some news articles are framed differently due to the word choice and labeling journalists choose to cover the topics. While some of the word choice depends on the journalists' stylistic preferences, most of the influential word choice originates from politicians' speeches or ideology of the media, who want to affect peoples' way of thinking and behaving.

For many years, social science researchers have been applying content and framing analyses to reveal the framing effect and make readers more aware of the media bias. Although being well-developed, the qualitative analyses are time-consuming and cannot be scaled into the larger number of articles. Approaches of automated WCL analysis can be applied to larger text collections and analyze WCL from actor perspective or topic perspective. In both cases, obtained lists of discrepant word choice require qualitative interpretation to make sense of the results. Moreover, none of the approaches analyze small sets of news articles identifying frequently mentioned semantic concepts by resolving broad word choice of referring anaphora.

Unlike existing methods for automated WCL analysis, NLP techniques, e.g., coreference resolution, address the problem of identification of the phrases that refer to the same entities. While most of the NLP techniques resolve anaphora of the common knowledge, e.g., named mentions "Trump" and "Donald Trump", the WCL analysis seeks to resolve coreferential phrases of non-named-entities of broader sense, e.g., "undocumented immigrants" and "illegal aliens."

The four contributions of this thesis are:

1) We proposed a unified methodology of the automated WCL analysis pipeline that analyzes a set of news articles related to one event. We designed and implemented the WCL analysis system's architecture that adheres to the analysis tasks defined in the pipeline. The WCL analysis pipeline combines principles of content and framing analyses and analyzes the word choice difference, first, from the actor perspective, and, then, from the topic perspective, thereby unifying two strategies of automated WCL analysis. We implemented the modules of the WCL analysis system that focus on the identification of the semantic concepts in the news articles.

2) As the main contribution, we proposed and implemented a multi-step merging approach (MSMA) for the candidate alignment task, which resolves coreferential phrases of named mentions and mentions a broader sense, e.g., "undocumented immigrants" and "illegal aliens," referring to non-named-entities occurring in the text. The proposed approach consists of six consecutively employed merging steps that extract different types of attributes and use these attributes to identify similarity between coreferential phrases and merge similar ones. The approach works for different types of entities such as actors, organizations, groups of individuals, objects, events, or more abstract entities.

3) We designed and implemented a usability prototype that enables users to explore the results of the identified entities interactively. The visualization tool includes four views that allow a user to have an overview of entity composition of the news articles, analyze phrasing diversity of

the identified entities, and broaden their perspective while reading news articles with underlined phrases referring to the same entities. We showed exemplar analysis tasks that a user can solve using the proposed visualization.

4) We evaluated proposed MSMA and showed that on average MSMA reaches $F1 = 0.84$ in identification and resolution of entity mentions. MSMA demonstrates a significant improvement over state-of-the-art coreference resolution ($F1_{B2} = 0.27$) and the best performing baseline based on the candidate clustering ($F1_{B3} = 0.42$). The evaluation is based on the extended NewsWCL50, a data corpus of manually annotated coreferential phrases referring to the semantic concepts. Given research focus on the entity identification instead of identification of complex semantic concepts, a simplified content analysis annotation methodology was applied to reannotate the corpus and evaluate MSMA. The evaluation showed better performance on the non-NE-based concept types, e.g., group of individuals, on a topic with a larger number of articles: the more diverse and repetitive WCL is, the more similarities MSMA can identify.

In future work, we will focus on concept mention disambiguation that have similar word choice but relate to different concepts. We also plan to identify more complex semantic concepts, experiment with various word vector models, and train a WCL identification neural network model on the large corpus of the news topics and reveal WCL difference in any topic-related aggregated articles hence allowing readers more informed and bias-aware news consumption.

Appendix

A1: Dataset overview

Table 14: Comparison of coded concepts between original CA and simplified CA

	Original CA			Simplified CA			
Topic	***Types***	***Codes***	***Size***	***Types***	***Codes***	***Size***	***WCL***
0	Country	USA	75	Country	PRK	58	4.92
		PRK	72		USA	35	6.81
		JPN	23		JPN	21	3.57
		KOR	17		KOR	9	3.00
		CHN	7		Koreas	6	3.00
		RUS	1		CHN	5	0.40
	Actor	USA\Trump	83	Actor	USA\Trump	83	1.77
		PRK\Jong Un	47		JPN\Abe	39	1.86
		JPN\Abe	46		PRK\Kim	36	4.45
		USA\Mike Pompeo	25		USA\Pompeo	24	4.25
		KOR\Moon Jae	6	Misc	PRK USA summit	28	6.42
		CHN\Jinping	3		PRK KOR peace	19	5.00
	Actor-I	USA\USA-I	6		PRK KOR war	16	3.78
		USA\USA-Misc	3		JPN USA meeting	13	6.00
		JPN\JPN-I	2		Trip to PRK	12	4.00
		PRK\PRK-Misc	2		Nuclear weapons	10	5.00
	Event	PRK USA Summit	45		PRK KOR meeting	9	5.00
		JPN USA Meeting	16		Denuclearization	4	2.50
		PRK KOR Meeting	8				
	Misc	Denuclearization	19				
	Action	Trip to PRK	28				
		Negotiate about the peace	19				
1	Country	USA	104	Country	USA	59	7.99
		RUS	16		RUS	14	0.75
		GBR	5	Actor	USA\Trump	95	1.71
	Actor	USA\Trump	116		USA\Comey	82	1.49
		USA\Comey	115		USA\Flynn	17	0.71
		USA\Flynn	24		RUS\Putin	7	1.67
		RUS\Putin	9	Group	USA\Democrats	8	1.83
	Actor-I	USA\USA-I	32	Misc	Comey memos	64	8.67
		USA\USA-Misc	14		Steele dossier	26	4.72
	Event	Comey Memos\ Russian Investigation	15		Interactions with Trump	18	7.17
		Comey Memos\ Flynn investigation	13		RUS interference	11	5.67
		RUS interference	12		Russian Investigation	10	1.78
	Misc	Comey Memos\ Interactions with Trump	23		Flynn investigation	8	3.00
	Object	Comey Memos	81				
		Comey Memos\ Steele dossier	26				
2	Country	PRK	112	Country	PRK	92	9.50
		USA	47		USA	29	7.30
		KOR	46		KOR	13	4.00
		JPN	7		Koreas	13	9.00
		RUS	2	Actor	PRK\Kim	89	5.89
		CHN	1		USA\Trump	41	1.40
		GUM	1		KOR\Moon	16	3.00

Table 14 (continued)

2	Actor	PRK\Kim	96	Misc	Nuclear weapons	49	20.15
		USA\Trump	27		Nuclear tests	27	4.35
		KOR\Moon	21		Kim's announcement	24	6.73
		CHN\Xi	1		Nuclear test site	14	2.83
	Actor-I	KOR\KOR-I	3		PRK suspension of nuclear tests	12	5.83
		USA\USA-I	2		PRK USA Summit	11	2.80
		USA\USA-Misc	1		PRK KOR relation improvement	10	7.50
		KOR\KOR-Misc	1		PRK KOR hotline	10	2.00
		PRK\PRK-Misc	1		PRK KOR summit	9	2.00
	Event	Kim's announcement	31		Denuclearization	5	0.60
		USA PRK summit	20		PRK KOR war	4	3.00
		PRK KOR summit	10				
	Misc	PRK nuclear tests/capabilities	40				
		Denuclearization	30				
		PRK plans to close down test site	27				
		PRK KOR end of military conflict	7				
	Action	PRK nuclear missile test suspension	46				
	Object	KOR PRK hotline	13				
3	Country	RUS	76	Country	RUS	43	2.51
		USA	53		USA	10	2.67
	Actor	USA\Republicans\Trump	71	Actor	USA\Trump	83	2.20
		USA\Democrats\Clinton	12		USA\Clinton	14	0.71
	Actor-I	USA\USA-Misc	8		USA\Mueller	13	2.78
		USA\USA-I	8		USA\Pascale	10	0.80
		RUS\RUS-I	1		USA\Gates	8	0.75
	Group	USA\Democrats	148		USA\Manafort	8	1.86
		USA\Republicans	119	Group	USA\Democrats	102	9.53
		Wikileaks	27		RUS\Agents	16	4.75
	Event	Trump campaign	66		WikiLeaks	14	0.61
		USA election 2016	32		USA\Republicans	6	1.60
		RUS investigation	22	Misc	Lawsuit	102	9.82
	Misc	Lawsuit	120		Trump campaign	62	3.57
		RUS interference into USA election	94		RUS interference into USA election	47	13.74
					USA elections 2016	28	4.23
4	Country	IRN	67	Country	IRN	86	1.82
		USA	49		USA	44	9.60
		FRA	20		Middle East	14	4.00
		SYR	9		FRA	11	2.22
		PRK	9		SYR	7	0.14
		ISR	3		PRK	4	0.75
		SAU	2	Actor	USA\Trump	73	1.33
	Actor	USA\Trump	122		FRA\Macron	34	3.55
		FRA\Macron	74		USA\Obama	9	1.63
		USA\Obama	12		PRK\Kim	4	3.00
		PRK\Kim	10		IRN\Rouhani	2	1.00
		IRN\Rouhani	2	Misc	IRN nuclear deal	97	11.74
	Actor-I	USA\USA-I	46		Nuclear program	21	5.85
		USA\USA-Misc	14		IRN new deal	7	3.00

Table 14 (continued)

4	Actor-I	IRN\IRN-I	3				
		ISR\ISR-I	3				
	Event	IRN restart nuclear	20				
	Misc	Reactions on deal	52				
	Object	IRN nuclear deal	86				
		IRN new deal	14				
5	Country	GBR	67	Country	GBR	42	4.58
		USA	34		USA	23	6.33
	Actor	USA\Trump	82		GBR	42	4.58
		GBR\May	33	Actor	USA\Trump	98	2.83
		GBR\GBR-Misc\Khan	17		GBR\May	30	2.06
	Actor-I	GBR\GBR-Misc	31		GBR\Khan	12	2.56
		GBR\GBR-I	12		GBR\Queen	10	2.50
		USA\USA-Misc	2	Group	GBR\London	31	3.00
		USA\USA-I	1		GBR\Demonstrators	6	1.50
	Event	Visit Trump UK	62	Misc	Trump's visit to GBR	37	3.69
		State Visit	34		State visit	24	3.94
		Protest	26		Protests	12	2.75
	Object	GBR\London	24		Cancelled visits to GRB	9	4.00
	Action	Break UK visits	19				
6	Country	USA	116	Country	USA	61	10.00
		MEX	30		MEX	28	1.08
	Actor	USA\Trump	24	Actor	USA\Trump	32	2.18
	Actor-I	Migrant caravan\Migrant caravan-I	36	Group	Migrant caravan	178	32.81
		USA\USA-I	28		USA\Border staff	29	8.88
		Supporters\Supporters-I	26		Caravan supporters	18	9.57
		USA\USA-Misc	4		Caravan organizers	9	1.83
	Group	Migrant caravan	142	Misc	USA MEX border	74	8.96
		Supporters	33		Asylum	18	0.06
		Migrant caravan\People without borders	16				
	Misc	Migration to the USA	40				
	Object	USA-MEX Border	70				
	Action	Migration to the USA\Seek asylum	17				
7	Country	USA	116	Country	USA	59	7.53
		EU	60		CAN	30	1.04
		CAN	33		EU	29	3.61
		CHN	26		MEX	23	1.05
		MEX	25		CHN	14	1.08
		KOR	12	Actor	USA\Trump	50	1.46
		JPN	8	Group	USA\Industry	8	3.00
	Actor	USA\Trump	33	Misc	Trump tariffs	60	3.63
		CAN\Trudeau	3		Permanent exemption from tariffs	28	2.82
		KOR\Moon	2		Metal	18	3.42
		JPN\Abe	2		Negotions about deals for exemption	16	4.21
	Actor-I	USA\USA-I	24				
		EU\EU-I	12				
		USA\USA-Misc	2				
	Group	USA\Steel producers	11				

Table 14 (continued)

7	Misc	Tariff imposition consequences	45				
		Reactions to the permanent exemption	21				
	Object	Tariffs	90				
		New import quotas	21				
		Permanent exemptions	15				
8	Country	USA	56	Country	RUS	31	1.66
		RUS	41		USA	27	3.10
	Actor	USA\Trump	122	Actor	USA\Trump	119	2.06
		USA\Mueller	68		USA\Mueller	86	1.25
		RUS\Putin	6		USA\Flynn	18	1.65
	Actor-I	USA\USA-I	121		Trump's lawyers	17	4.57
		Trumps Lawyers\Trumps Lawyers-I	34		Trump's lawyers\Dowd	17	1.90
		USA\USA-Misc	8		Trump's lawyers\Manafort	14	0.64
		RUS\RUS-I	8		USA\Sessions	6	2.33
	Group	Trumps Lawyers	16		USA\Comey	6	1.00
		USA\Mueller\Investigators	13	Group	Trump's lawyers	17	4.57
	Event	RUS meddling investigation	38		Mueller's team of investigators	12	1.57
		RUS meddling	33	Misc	Questions to Trump	53	5.11
	Object	Inquiries for Trump	66		Mueller's investigation	39	4.92
					RUS meddling	17	7.00
					Trump's campaign	13	3.00
9	Country	IRN	96	Country	IRN	86	3.85
		USA	37		ISR	33	3.29
		ISR	36		USA	27	6.17
	Actor	ISR\Netanyahu	58	Actor	ISR\Netanyahu	57	2.73
		USA\Trump	24		USA\Trump	34	2.88
	Actor-I	USA\USA-I	16		USA\Obama	7	0.71
		ISR\ISR-I	13	Misc	IRN nuclear deal	58	4.85
		IRN\IRN-I	9		IRN nuclear program	43	10.85
		IRN\IRN-Misc	1		Iran files	28	7.92
	Event	Presentation	15		Netanyahu's presentation	10	3.00
	Misc	IRN nuclear activity	60				
		Reaction to IRN deal	24				
	Object	Nuclear deal	57				
		IRN files	38				
10				Country	USA	413	23.89
				Actor	USA\Trump	284	1.30
					USA\Obama	100	1.33
					USA\Sessions	97	2.75
				Group	Immigrants	393	46.61
					USA\Lawmakers	70	11.35
					USA\Americans	39	11.71
					USA\Officials	36	7.69
				Misc	DACA program	341	10.67
					Decision to close DACA	123	7.81
					New immigration reform	107	13.92
					DACA status	69	13.47

A2: News excerpts with similar meaning but different word choice

Table 15: Extracted sentences with similar meaning but different word choice

Topic	#	Publisher	Excerpt
0		LL	CIA Director Mike Pompeo secretly met with North Korean leader Kim Jong Un.
		R	Mike Pompeo met with Kim Jong Un in North Korea last week.
2	1	LL	Pyongyang also said it plans to close down a former test site.
		M	“North Korea has agreed to suspend all Nuclear Tests and close up a major test site,” he said.
	2	L	North Korea’s leader, Kim Jong-un, announced early Saturday that his country no longer needed to test nuclear weapons or long-range missiles and would close a nuclear test site.
		L	Mr. Kim said his country required no further nuclear and long-range missile tests because it had already achieved a nuclear deterrent.
3	1	LL	In its federal lawsuit in Manhattan, the Democratic National Committee said that top officials in Trump’s campaign conspired with the Russian government and its military spy agency to hurt Democratic presidential nominee Hillary Clinton and tilt the election to Trump by hacking Democratic Party computers.
		R	The new suit claims that Trump campaign officials worked in tandem with the Russian government and its military spy agency to bring down Hillary Clinton by hacking into the computer networks of the DNC and spreading stolen material.
		RR	The DNC is alleging, in a complaint filed in federal district court in Manhattan, that top Trump campaign officials \"conspired\" with the Russian government and its military spy agency to hurt Hillary Clinton and help Trump, but hacking the email servers of the Democratic National Committee and disseminating them, according to the Post.
	2	R	DNC files lawsuit over election interference.
		RR`	DESPERATION: DNC files multi-million dollar lawsuit against Russia, Trump campaign for collusion.
	3	LL	“During the 2016 presidential campaign, Russia launched an all-out assault on our democracy and it found a willing and active partner in Donald Trump’s campaign,” said Tom Perez, chair of the DNC.
		R	“During the 2016 presidential campaign, Russia launched an all-out assault on our democracy, and it found a willing and active partner in Donald Trump’s campaign,” said DNC Chairman Tom Perez in a statement, calling the alleged collusion “an act of unprecedented treachery.”
4		LL	The Iran deal is a terrible deal.
		RR	Trump repeated that the deal reached under the Obama administration was a “terrible deal.”

Table 15 (continued)

5		LL	Donald Trump will visit London in July despite the threat of protests, the US ambassador to the UK has insisted.
		LL	Despite the threat of widespread protests, the US ambassador Woody Johnson confirmed Trump would “definitely be coming to London”.
6		LL	But hundreds of participants in the caravan made their way to the U.S.-Mexico border on Sunday anyway.
		R	Members of migrant caravan prepare to reach U.S.-Mexico border.
8	1	LL	The New York Times obtained a list of possible questions for the president.
		RR	The Times obtained a list of questions it said Mueller’s team read over the telephone to Trump’s legal team, which compiled them into a list.
	2	LL	He said in January he was “looking forward” to speaking with the special counsel.
		R	He said he is “looking forward” to eventually being questioned under oath by Mueller.
	3	LL	The Times noted that four people in the president’s orbit have already pleaded guilty to lying to federal investigators.
		L	Four people, including Mr. Flynn, have pleaded guilty to lying to investigators in the Russia inquiry.
10	1	L	Former President Barack Obama on Tuesday bashed his successor's decision to rescind an immigration order shielding some children of undocumented immigrants from deportation, calling the move "cruel" and "self-defeating."
		LL	Former President Barack Obama, who had warned that any threat to the program would prompt him to speak out, called his successor's decision 'wrong,' 'self-defeating' and 'cruel.'
	2	RR	Attorney General Jeff Sessions made the rumored end of Obama's DACA amnesty program official Tuesday.
		LL	Attorney General Jeff Sessions announced on Tuesday the end of an Obama-era immigration program that shielded young immigrants from deportation.
	3	L	I have a love for these people.
		R	"I do not favor punishing children, most of whom are now adults, for the actions of their parents," Trump's statement went on.
	4	M	Such legislation was last voted on in 2010, when it passed the House but fell five votes short in the Senate.
		M	In 2010, the Dream Act passed the House, then controlled by Democrats, but fell five votes short of the 60 needed in the Senate.
	5	L	The lengthy statement is among Obama's most forceful since departing office.
		M	Mr. Obama issued a rare public statement in opposition to his successor.

Bibliography

[1] Joohui An, Seungwoo Lee, and Gary Geunbae Lee. 2003. Automatic acquisition of named entity tagged corpus from world wide web. *In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* 2, (2003), 165–168. DOI:https://doi.org/10.3115/1075178.1075207

[2] Mihael Ankerst, Markus M Breunig, Hans-peter Kriegel, and Jörg Sander. 1999. OPTICS : Ordering Points To Identify the Clustering Structure. *In ACM Sigmod record* 28, 2 (1999), 49–60.

[3] David Arthur and Sergei Vassilvitskii. 2007. k-means ++ : The Advantages of Careful Seeding. *In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (2007), 1027–1035.

[4] Michael R Berthold, Christian Borgelt, Frank Höppner, and Frank Klawonn. 2010. *Guide to intelligent data analysis: how to intelligently make sense of real data*. Springer Science & Business Media.

[5] Christian Borgelt. 2012. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2, 6 (2012), 437–456. DOI:https://doi.org/10.1002/widm.1074

[6] Margaret M Bradley, Peter J Lang, Margaret M Bradley, and Peter J Lang. 1999. Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings. *Technical report C-1, the center for research in psychophysiology, University of Florida* 30, 1 (1999).

[7] Michael J. Brusco and Hans Friedrich Köhn. 2008. Comment on “clustering by passing messages between data points.” *Science 319*, 972–976. DOI:https://doi.org/10.1126/science.1150938

[8] Dallas Card, Justin H Gross, Amber E Boydstun, and Noah A Smith. 2016. Analyzing Framing through the Casts of Characters in the News. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)* (2016), 1410–1420. Retrieved from http://homes.cs.washington.edu/~nasmith/papers/card+gross+boydstun+smith.emnlp16.pdf

[9] Nancy Chinchor and Ph D. 1992. MUC-5 EVALUATION METRIC S Science Applications International Corporatio n 10260 Campus Point Drive , MIS A2-F San Diego , CA 9212 1 Naval Command , Control , and Ocean Surveillance Cente r RDT & E Division (NRaD) Information Access Technology Project Te. *System* (1992), 69–78.

[10] Dennis Chong and James N. Druckman. 2007. Framing Theory. *Annual Review of Political Science* 10, 1 (2007), 103–126. DOI:https://doi.org/10.1146/annurev.polisci.10.072805.103054

[11] Kevin Clark and Christopher D Manning. 2016. Deep Reinforcement Learning for Mention-Ranking Coreference Models. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016), 2256–2262.

DOI:https://doi.org/10.18653/v1/D16-1245

[12] Kevin Clark and Christopher D Manning. 2016. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (2016), 643–653.

[13] Scott A. Crossley, Kristopher Kyle, and Danielle S. McNamara. 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

[14] William H E Day and Herbert Edelsbrunner. 1984. Efficient Algorithms for Agglomerative Hierarchical Clustering Methods. *Journal of classification* 1, 1 (1984), 7–24.

[15] James N Druckman. 2004. Political Preference Formation : Competition and the (Ir) relevance of Framing Effects. *The American Political Science Review* 98, 4 (2004), 671–686.

[16] Sourav Dutta and Gerhard Weikum. 2015. Cross-Document Co-Reference Resolution using Sample-Based Clustering with Knowledge Enrichment. *Transactions of the Association for Computational Linguistics* 3, (2015), 15–28. DOI:https://doi.org/10.1002/pa

[17] Allie Duzett. 2011. Media Bias in Strategic Word Choice. *http://www.aim.org/on-target-blog/media-bias-in-strategic-word-choice/*.

[18] Robert M. Entman. 1993. Framing: Toward Clarification of a Fractured Paradigm. *Journal of Communication* 43, 4 (1993), 51–58. DOI:https://doi.org/10.1111/j.1460-2466.1993.tb01304.x

[19] Robert M. Entman. 2007. Framing bias: Media in the distribution of power. *Journal of Communication* 57, 1 (2007), 163–173. DOI:https://doi.org/10.1111/j.1460-2466.2006.00336.x

[20] Martin Ester, Hans-peter Kriegel, Xiaowei Xu, and D- Miinchen. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *In KDD* 96, 34 (1996), 226–231.

[21] Julian J Faraway. 2016. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press.

[22] Usama Fayyad, Gregory Piatetsky-shapiro, and Padhraic Smyth. 1996. From Data Mining to Knowledge Discovery in databases. *AI magazine* 17, 3 (1996), 37–54.

[23] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)* (2005), 363–370.

[24] Blaz Fortuna, Carolina Galleguillos, and Nello Cristianini. 2009. Detection of bias in media outlets with statistical learning methods. In *Text Mining*. Chapman and Hall/CRC, 57–80.

[25] Dianne M. Garyantes and Priscilla J. Murphy. 2010. Success or chaos?: Framing and ideology in news coverage of the Iraqi national elections. *International Communication Gazette* 72, 2 (2010), 151–170. DOI:https://doi.org/10.1177/1748048509353866

[26] Felix Hamborg, Karsten Donnay, and Bela Gipp. 2018. Automated identification of media bias in news articles : an interdisciplinary literature review. *International Journal on Digital Libraries* (2018). DOI:https://doi.org/10.1007/s00799-018-0261-y

[27] Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please : a Generic News Crawler and Extractor. In *Everything changes, everything stays the same : Understanding Information Spaces; Proceedings of the 15th International Symposium of Information Science (ISI 2017), Berlin, Germany, 13th-15th March 2017* (Schriften zur Informationswissenschaft), 218–223.

[28] Felix Hamborg, Norman Meuschke, and Bela Gipp. 2018. Bias-aware news analysis using matrix-based news aggregation. *International Journal on Digital Libraries* (2018), 1–30. DOI:https://doi.org/10.1007/s00799-018-0239-9

[29] Felix Hamborg, Anastasia Zhukova, and Bela Gipp. 2019. Illegal Aliens or Undocumented Immigrants ? Towards the Automated Identification of Bias by Word Choice and Labeling. *in Proceedings of the iConference 2019* (2019).

[30] Felix Hamborg, Anastasia Zhukova, and Bela Gipp. 2019. Automated Identification of Media Bias by Word Choice and Labeling in News Articles. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (2019), 1–10.

[31] Ariana Hoyer. 2016. Spanish News Framing of the Syrian Refugee Crisis. (2016), 32.

[32] Daniel Kahneman and Amos Tversky. 1984. Choices, values, and frames. *American Psychologist* 39, 4 (1984), 341–350. DOI:https://doi.org/10.1037/0003-066X.39.4.341

[33] Heeyoung Lee. 2017. A Scaffolding Approach to Coreference Resolution Integrating Statistical and Rule-based Models. *Natural Language Engineering* 23, 5 (2017), 733–762.

[34] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, 707–710.

[35] Mark Liberman and Richard Sproat. 1992. Modified Noun Phrases in English. *Lexical matters 24* 24, 131 (1992).

[36] W. Linstrõm, M., & Marais, Margaret Linstrom, and Willemien Marais. 2012. Qualitative News Frame Analysis: A Methodology. *Communitas* 17, 17 (2012), 21–38.

[37] Margaret Linstrom and Willemien Marais. 2012. Qualitative News Frame Analysis: A Methodology. *Communitas* 17, (2012), 21–38.

[38] Christopher D Manning, John Bauer, Jenny Finkel, and Steven J Bethard. 2014. The Stanford CoreNLP Natural Language Processing Toolkit Christopher. *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (2014), 55–60. Retrieved from http://macopolo.cn/mkpl/products.asp

[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Advances in neural information processing systems* (2013), 3111–3119.

[40] George A Miller. 1995. WordNet : A Lexical Database for English. *Communications of the ACM* 38, 11 (1995), 39–41.

[41] James Edward Miller and Jim Miller. 2011. *A critical introduction to syntax*. A&C Black.

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

[43] Viet-an Nguyen, Jordan Boyd-graber, and Philip Resnik. 2013. Lexical and Hierarchical Topic Regression. *Advances in Neural Information Processing Systems 26 (NIPS 2013)* (2013), 1–9.

[44] Zizi Papacharissi and Maria de Fatima Oliveira. 2008. News frames terrorism: A comparative analysis of frames employed in terrorism coverage in U.S. and U.K. newspapers. *International Journal of Press/Politics* 13, 1 (2008), 52–74. DOI:https://doi.org/10.1177/1940161207312676

[45] James W Pennebaker, Roger J Booth, Ryan L Boyd, and Martha E Francis. Linguistic Inquiry and Word Count: LIWC2001. *Mahway: Lawrence Erlbaum Associates 71.2001* 2001 , 1–22.

[46] Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic Models for Analyzing and Detecting Biased Language. *Proceedings of the 51st Annual Meeting on Association for Computational Linguistics* (2013), 1650–1659.

[47] Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language (EMNLP-CoNLL'07)* 1, June (2007), 410–420. DOI:https://doi.org/10.7916/D80V8N84

[48] Margrit Schreier. 2012. *Qualitative content analysis in practice*. Sage publications.

[49] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Cambridge University Press.

[50] Marie Gustafsson Sendén, Sverker Sikström, and Torun Lindholm. 2015. "She" and "He" in news media messages: Pronoun use reflects gender biases in semantic contexts. *Sex Roles* 72, 1 (2015), 40–49. DOI:https://doi.org/10.1007/s11199-014-0437-x

[51] Walid Shalaby, Wlodek Zadrozny, and Hongxia Jin. 2018. Beyond Word Embeddings: Learning Entity and Concept Representations from Large Scale Knowledge Bases. *Information Retrieval Journal* (2018), 1–18. DOI:https://doi.org/10.1007/s10791-018-9340-3

[52] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew Mccallum. 2011. Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* 1, (2011), 793–803.

[53] Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology* 29, 1 (2010), 24–54. DOI:https://doi.org/10.1177/0261927X09351676

[54] Yan Tian and Concetta M. Stewart. 2005. Framing the SARS Crisis: A Computer-

Assisted Text Analysis of CNN and BBC Online News Reports of SARS. *Asian Journal of Communication* 15, 3 (2005), 289–301. DOI:https://doi.org/10.1080/01292980500261605

[55] How many ways are there to tell the same story? *http://umich.edu/~newsbias/wordchoice.html*, 1–5.