Annotation and Evaluation of Coreference Resolution in Screenplays

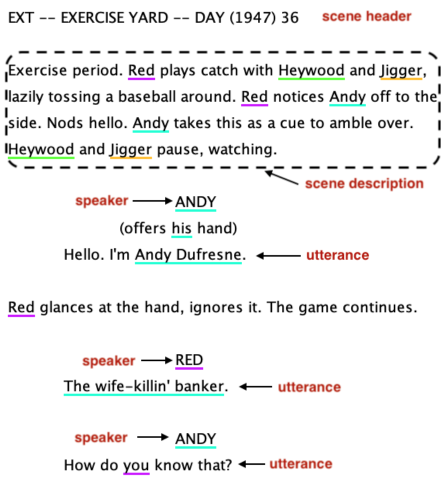
Sabyasachee Baruah sbaruah@usc.edu
Sandeep Nallan Chakravarthula nallanch@usc.edu
Shrikanth Narayanan shri@ee.usc.edu
Signal Analysis and Interpretation Laboratory
University of Southern California

Abstract

Screenplays refer to characters using different names, pronouns, and nominal expressions. We need to resolve these mentions to the correct referent character for better story understanding and holistic research in computational narratology. Coreference resolution of character mentions in screenplays becomes challenging because of the large document lengths, unique structural features like scene headers, interleaving of action and speech passages, and reliance on the accompanying video. In this work, we first adapt widely-used annotation guidelines to address domain-specific issues in screenplays. We develop an automatic screenplay parser to extract the structural information and design coreference rules based upon the structure. Our model exploits these structural features and outperforms a benchmark coreference model on the screenplay coreference resolution task.

1 Introduction

Screenplays are semi-structured text documents containing the dialogue and directions of a film. Automated screenplay analysis provides an opportunity early in the creative process to offer insights into character representations and portrayals (of who interacts with whom, about what, and how), including from a diversity, inclusion, and social impact perspective (Ramakrishna et al., 2017; Shafaei et al., 2020; Martinez et al., 2020). A typical screenplay contains indented blocks of text that can be classified into scene headers, scene descriptions, speakers, and utterances, as shown in Figure 1 (Agarwal et al., 2014). A scene header starts a new scene and provides location and temporal information. Scene descriptions describe the characters and their actions, and the speaker and utterance blocks contain the characters' names and speech.


EXT -- EXERCISE YARD -- DAY (1947) 36

Exercise period. Red plays catch with Heywood and Jigger, lazily tossing a baseball around. Red notices Andy off to the side. Nods hello. Andy takes this as a cue to amble over. Heywood and Jigger pause, watching.

ANDY
(offers his hand)
Hello. I'm Andy Dufresne.

Red glances at the hand, ignores it. The game continues.

RED
The wife-killin' banker.

ANDY
How do you know that?

Figure 1: Coreference-annotated screenplay excerpt from the movie *The Shawshank Redemption (1994)*. Mentions of the same character are underlined with the same color.

Screenplays can refer to a character by different names, pronouns, and nominal expressions. For example, the screenplay excerpt shown in Figure 1 refers to the character *Andy* by the mentions – *Andy Dufresne* (name), *The wife-killin' banker* (nominal), and *his* (pronoun). Many downstream tasks need to find and map all such mentions to the correct *referent*. For example, Gorinski and Lapata (2015) resolved pronominal mentions to their correct antecedent (prior co-referring mention) to find speaker-listener and semantic relations between characters for the movie-summarization task. Chen and Choi (2016) crowdsourced character-mention labels in TV show transcripts for automatic character identification. Deleris et al. (2018) used direct and indirect character references in utterances to study social relationships between characters. Thus, resolving character mentions in screenplays is an essential subtask in many applications. In NLP literature, this task is formally called coreference resolution (Jurafsky and Martin, 2009) and has been studied extensively (Sukthanker et al., 2020; Stylianou and Vlahavas, 2021).

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2004–2010
August 1–6, 2021. ©2021 Association for Computational Linguistics

Most existing coreference datasets focus on news and web text (Pradhan et al., 2012; Webster et al., 2018), but do not include screenplays that have distinct content and structure. First, the document lengths of screenplays are much larger than news articles (Gorinski and Lapata, 2015), increasing the computational complexity of antecedent scoring (Lee et al., 2017). Second, scene headers alter the story's context affecting coreference between mentions of different scenes. Lastly, coreference annotation of some mentions requires the knowledge of the accompanying movie or TV clip because textual descriptions may not capture all the post-production visual details of a scene. Thus, two main challenges for coreference resolution in screenplays are: (1) the lack of suitable annotation rules to handle domain-specific issues and (2) finding methods to leverage the unique screenplay structure to improve coreference resolution.

Our objective in this work is to address the coreference resolution of characters in screenplays. We only focus on characters because most modern narrative studies are centered around their role as agents driving the plot (Bamman et al., 2013; Labatut and Bost, 2019). Our contributions are 1) we establish coreference annotation guidelines for screenplays and use them to label screenplay excerpts, 2) we develop a screenplay parser to convert the semi-structured text into a machine-readable format, and 3) we use the structural information of screenplays to design coreference rules, which improves the performance of coreference resolution when combined with a benchmark coreference resolution model (Lee et al., 2018).

2 Related Work

Screenplay Parsing: Weng et al. (2009) motivated the need for screenplay parsing for social network analysis. Agarwal et al. (2014) formalized the screenplay parsing task and developed an SVM-based parser. Winer and Young (2017) extended it to extract fine-grained information from scene headers. Our parser uses a rule-based algorithm to achieve comparable performance.

Coreference Resolution: OntoNotes 5 is the benchmark dataset for English coreference resolution, containing documents from newswire, broadcast news, telephone conversations, and weblogs (Pradhan et al., 2012). However, it does not contain screenplay texts. The closest work to screenplay coreference is the *LitBank* dataset (Bamman et al., 2020), which contains coreference annotations of 100 works of fiction taken from Project Gutenberg (Lahiri, 2014).

Few previous works address coreference annotation in screenplays. Chen and Choi (2016) labeled character mentions for two TV shows: *Friends* and *The Big Bang Theory*. Zhou and Choi (2018) later extended this dataset by including plural mentions, but mainly focused on character utterances and did not consider action notes. Ramanathan et al. (2014) created a dataset of 19 TV episodes for joint coreference resolution in visual and textual media content. Gorinski and Lapata (2015) created the *ScriptBase* corpus of movie screenplays, which included coreference labels. However, they found the labels using the Stanford CoreNLP system (Lee et al., 2011), which has not been evaluated for screenplay coreference.

3 Annotation

We annotate screenplays with character mentions for coreference resolution (see Figure 1). Following OntoNotes 5 annotation guidelines, we mark the maximal extent of noun phrases, pronouns, and possessives that refer to some character (Pradhan et al., 2012). Characters do not have to be persons; consider for example, the spider *Aragog* in the *Harry Potter* movies. We include singleton character entities. We do not label mentions that refer to multiple characters because finding the correct antecedent of plural mentions often requires the accompanying video clip's aid (Zhou and Choi, 2018). For example, it is difficult to decide whether *They* refers to *Vosen and agents*, *Others*, or both in the following lines without watching the movie.

> [Vosen and agents]$_x$ come running out of the front door. [Others]$_y$ leave through a side entrance. [They] jump in sedans. (*Bourne Ultimatum, 2007*)

We follow OntoNotes 5 annotation guidelines to handle appositions (adjacent noun phrases separated by comma, colon, or parentheses), copula (noun phrases connected by linking verbs, for example, *is*, *look*, etc.), and generic *you* mentions (Pradhan et al., 2012). If a mention's referent is revealed to be identical with another character as the story progresses, we tag the mention with the latter character (Bamman et al., 2020). The screenplay sometimes contains references to the reader or the camera's point of view. We tag such instances with a special READER entity, for example:

Movie	Tokens	Characters	Names	Pronouns	Nominals
The Shawshank Redemption (1994)	7,734	44	329	448	111
Bourne Ultimatum (2007)	7,722	39	491	344	76
Inglourious Basterds (2009)	7,284	23	426	458	124
TOTAL	22,740	106	1,246	1,250	311

Table 1: Statistics of the coreference annotated data.

> In the background, [we]$_{READER}$ see, [our]$_{READER}$ three counterfeit German Officers, Hicox, Wicki, and Stiglitz, enter the basement tavern. They obviously see the five German soldiers, but their too far away for [us (the audience)]$_{READER}$ to read their face. (*Inglourious Basterds, 2009*)

Screenplay Coreference Dataset: We annotated screenplay excerpts of three movies: *The Shawshank Redemption*, *Bourne Ultimatum*, and *Inglourious Basterds*. We downloaded the screenplay documents from IMSDb [1]. We chose these movies because the annotators were familiar with them, and they cover a wide range of genres (drama, action, thriller, and war). Three doctoral students were each assigned one screenplay, which they annotated for coreference according to the guidelines of section 3. The lead author checked the annotations independently for labeling errors. Less than 1% of the mentions required correction, suggesting high overall agreement. Table 1 describes some statistics of the labeled data. The corpus contains $2,807$ mentions in total, covering 106 characters. More than 44% of the mentions are pronouns, and about 11% are nominal mentions.

4 Model

Our coreference model consists of two parts: 1) a screenplay parser to extract structural information, and 2) coreference rules to resolve mentions occurring in speaker blocks.

4.1 Screenplay Parser

The screenplay parser reads raw screenplay text documents and assigns a structural tag – scene header, scene description, speaker, utterance or *other* (see Figure 1) to each line, following the tagset of Agarwal et al. (2014). The parser uses regular expressions to assign the structural tags. Thus, it is a rule-based 5-way multiclass classifier. The *other* tag includes all lines that do not fall in the other four categories, for example, camera transitions (*CUT TO*, *FADE IN*, etc.), dialogue metadata expressions (*O.S.*, *V.O.*, *(shouting)*, etc.), etc. The parser removes blank lines and whitespace indents.

4.2 Coreference Resolution

We use the following strategy to find the coreference clusters of characters in screenplays.

Add *says*: Given a screenplay, we parse it using our screenplay parser and collect all lines tagged as scene header, scene description, speaker, or utterance. We add the word *says* after speaker-tagged lines and concatenate all lines, separated by a newline delimiter. This lexical addition tells the model that the character mentioned in the speaker-tagged line speaks the succeeding utterance-tagged lines.

Keep speakers: We apply a coreference resolution model, pre-trained on OntoNotes 5, to the concatenated text to find coreference clusters. Since OntoNotes 5 annotates for unrestricted coreference (find coreference clusters of all ACE entities and events), we need to prune the clusters to keep only those containing character mentions. We keep clusters that contain any mention which appears in a speaker-tagged line.

Merge clusters: Due to the large document size (see Table 1), long coreference chains, like those of main characters, sometimes get segmented and occur as separate clusters. We merge the segmented clusters using speaker information. Screenplays usually refer to a character by a unique name in the speaker-tagged lines. If two speaker-tagged lines belonging to separate clusters contain identical names, we merge the corresponding clusters.

5 Experiments and Results

5.1 Screenplay Parser Evaluation

We annotated lines of 39 movie screenplay excerpts for the structural tags. These movies were different from the ones annotated for coreference. Three annotators, all doctoral students, labeled 9,758 lines, with a Krippendorff's inter-rater reliability score of 0.983 (strong agreement). We parsed the annotated excerpts using our rule-based parser and evaluated its classification performance. Table 3 shows the

[1] `http://www.imsdb.com`

Model	MUC			B^3			CEAF_e			Avg		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline	83.7	62.0	70.8	55.3	33.9	41.6	28.1	23.6	23.8	55.7	39.8	45.4
Our Model	**87.3**	**79.5**	**83.2**	**65.8**	**61.2**	**63.2**	**50.2**	**40.6**	**41.6**	**67.7**	**60.4**	**62.7**
-Add says	-1.0	-2.5	-1.9	-9.3	-15.0	-12.7	-5.1	-3.6	-4.3	-5.1	-7.0	-6.3
-Keep speakers	-2.5	-2.2	-2.4	-2.4	-2.5	-2.4	-17.7	+3.5	-6.5	-7.5	-0.4	-3.7
-Merge clusters	-0.2	-2.9	-1.7	+3.8	-9.8	-5.3	-1.1	-6.3	-3.8	+0.8	-6.3	-3.6

Table 2: Coreference resolution performance. The last three rows show the change in scores when the corresponding rule is removed from our model.

precision, recall, and F1 scores of the parser on the annotated data. The performance is comparable with the parser developed by Agarwal et al. (2014).

Structure Tag	lines	P	R	F1
Scene Header	343	94.5	90.4	92.4
Scene Description	3074	88.2	87.6	87.9
Speaker	1833	95.4	96.1	95.7
Utterance	3754	90.5	94.5	92.5

Table 3: Screenplay parser's classification performance

5.2 Screenplay Coreference Evaluation

We evaluated our coreference model on the labeled screenplay data collected in section 3. We used the model of Lee et al. (2018), a widely used benchmark for coreference resolution, as the pre-trained model for our method. Lee et al. (2018) calculates mention scores for each span, retains the top-scoring ones and finds antecedent scores of mention pairs. We replaced the model's original GloVe vectors with SpanBERT (large) embeddings. Joshi et al. (2020) showed that the SpanBERT-adapted model obtains 79.6 avg. F1 on the OntoNotes 5 test set for the unrestricted coreference task.

Baseline: Given a screenplay, we find the structural tags for each line using our parser and retain the lines tagged as scene header, scene description, speaker, or utterance. We input the text to the coreference resolution model of Lee et al. (2018), pre-trained on OntoNotes 5. We keep clusters that contain any mention whose named entity tag is *PERSON*. We use spaCy (Honnibal et al., 2020) for named entity recognition.

Metrics: Following the official evaluation framework of CoNLL-2012 multilingual coreference task (Pradhan et al., 2014), we used the MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998) and CEAF_e (Luo, 2005) measures to evaluate our model. We also used the NEC score (Agarwal et al., 2019) to evaluate coreference resolution by mention-type – name, pronoun and nominal. Tables 2 and 5 show the performance of our model and baseline in coreference resolution and character mention identification respectively. Table 4 shows the NEC scores for different mention types.

	Name	Pronoun	Nominal
Baseline	39.1	39.8	48.8
Our Model	**47.6**	**62.1**	**55.4**

Table 4: NEC F1 by mention type

	P	R	F1
Baseline	86.5	63.9	73.1
Our Model	**88.7**	**80.8**	**84.5**

Table 5: Mention identification performance.

Ablation Study: We study how each coreference rule – *Add says*, *Keep speakers*, and *Merge clusters* – described in section 4.2, contributes to the model's performance. Table 2 shows the results of the ablation experiments. *-Add says* means that we do not add *says* after speaker-tagged lines, *-Keep speakers* implies that we retain clusters that contain any mention whose named entity tag is *PERSON* instead of those that contain any mention appearing in speaker-tagged lines, and *-Merge clusters* denotes that we do not merge clusters.

6 Discussion

The results of Table 2 suggests that inputting the raw screenplay directly to the pre-trained coreference model performs poorly. The performance substantially improves when we use the coreference rules (+17.3 avg. F1). The improvement is largest for pronouns (+22.3 NEC F1), as shown in Table 4, possibly because of the *Add says* rule that helps the model to find the antecedent of personal pronouns in utterance-tagged lines. The rule adds

6.3 avg. F1 towards the overall performance (Table 2). Coreference resolution of named mentions also improves greatly (+8.5 NEC F1), probably because of the *Merge clusters* rule that joins clusters if they contain mentions in speaker-tagged lines that have identical names. It contributes 3.6 avg. F1 to the final score (Table 2). The *Keep speakers* rule adds 3.7 avg. F1, which suggests that retaining clusters containing speaker-tagged mentions is better than keeping those containing *PERSON*-tagged (NER) mentions to retrieve character references.

7 Applications

We show two applications of coreference resolution in computational narratology: finding mention-type interactions and character actions.

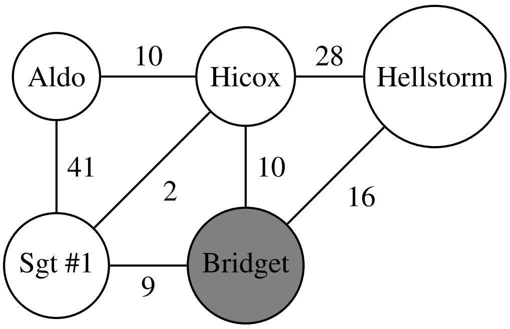


Figure 2: Character network of speech interactions. Edge weight is the number of interactions. Bridget has the **lowest** weighted degree.

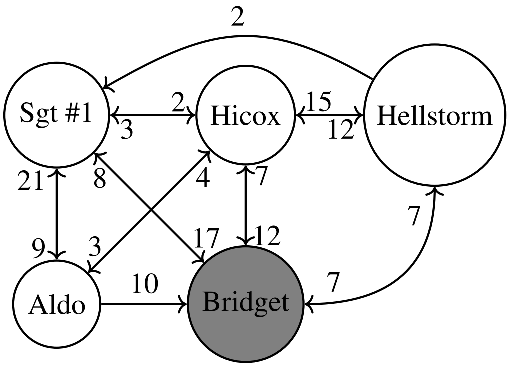


Figure 3: Character network of mention interactions. Edge weight is the number of mentions. Bridget has the **highest** weighted in-degree.

Mention-type Interactions: Figures 2 and 3 show character networks of the top five speaking characters from the movie *Inglourious Basterds (2009)*, capturing speech and mention-type interactions respectively. The edge weight between characters A and B in the speech network (Fig 2) is the number of times A speaks right after B or vice versa (Ramakrishna et al., 2017). The directed edge weight from character A to character B in the mention network (Fig 3) is the number of times A mentions B in their speech. We used the structural tags and coreference annotations to create these networks.

We observe that the two character networks provide different insights. Using degree centrality, *Bridget* is the least 'important' character in terms of speech interactions, but is most mentioned by other characters. This supports the movie plot which contains a scene where the Basterds discuss their plans of meeting *Bridget*, without her being there.

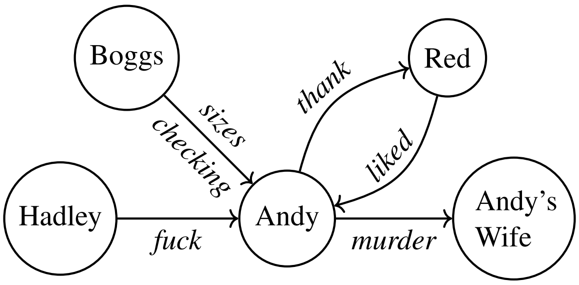


Figure 4: Character network of action interactions. Edge labels are actions. Head node of edge is the agent, and tail node of edge is the patient of the action.

Character Actions: We can use semantic role labeling and coreference resolution to find character actions. Figure 4 shows a subgraph of the character action network of *The Shawshank Redemption (1994)* movie. The directed edge label is the action, the head node is the agent (ARG0), and the tail node is the patient (ARG1) of the action. We applied the SRL model of Shi and Lin (2019) to the sreenplay's sentences, and then substituted the semantic roles with their referred character, wherever possible. From figure 4, we observe that *Andy* had positive interactions with *Red*, but was negatively treated by *Boggs* and *Hadley*, which is in line with the movie plot.

8 Summary and Future Work

We presented a coreference annotation guideline for screenplays and developed rules based on the screenplay's structure to improve coreference resolution performance. Our work can facilitate future annotation and modeling of coreference resolution in screenplays to support computational narratology studies. We plan to label more screenplays to train an end-to-end coreference model and study character interactions using coreference clusters. The data is available in the supplementary material.

References

Apoorv Agarwal, Sriramkumar Balasubramanian, Jiehan Zheng, and Sarthak Dash. 2014. Parsing screenplays for extracting social networks from movies. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 50–58, Gothenburg, Sweden. Association for Computational Linguistics.

Oshin Agarwal, Sanjay Subramanian, Ani Nenkova, and Dan Roth. 2019. Evaluation of named entity coreference. In *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 1–7, Minneapolis, USA. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.

Yu-Hsin Chen and Jinho D. Choi. 2016. Character identification on multiparty conversation: Identifying mentions of characters in TV shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles. Association for Computational Linguistics.

Léa Deleris, Francesca Bonin, Elizabeth Daly, Stéphane Deparis, Yufang Hou, Charles Jochim, Yassine Lassoued, and Killian Levacher. 2018. Know who your Friends are: Understanding social connections from unstructured text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 76–80, New Orleans, Louisiana. Association for Computational Linguistics.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40.

Shibamouli Lahiri. 2014. Complexity of word collocation networks: A preliminary structural analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–105, Gothenburg, Sweden. Association for Computational Linguistics.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, Oregon, USA. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Victor Martinez, Krishna Somandepalli, Yalda Tehranian-Uhls, and Shrikanth Narayanan. 2020. Joint estimation and analysis of risk behavior ratings in movie scripts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4780–4790, Online. Association for Computational Linguistics.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Anil Ramakrishna, Victor R. Martínez, Nikolaos Malandrakis, Karan Singla, and Shrikanth Narayanan. 2017. Linguistic analysis of differences in portrayal of movie characters. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1669–1678, Vancouver, Canada. Association for Computational Linguistics.

Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei. 2014. Linking people in videos with “their” names using coreference resolution. In *European conference on computer vision*, pages 95–110. Springer.

Mahsa Shafaei, Niloofar Safi Samghabadi, Sudipta Kar, and Thamar Solorio. 2020. Age suitability rating: Predicting the MPAA rating based on movie dialogues. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1327–1335, Marseille, France. European Language Resources Association.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Nikolaos Stylianou and Ioannis Vlahavas. 2021. A neural entity coreference resolution review. *Expert Systems with Applications*, 168:114466.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Chung-Yi Weng, Wei-Ta Chu, and Ja-Ling Wu. 2009. Rolenet: Movie analysis from the perspective of social networks. *IEEE Transactions on Multimedia*, 11(2):256–271.

David Winer and R Young. 2017. Automated screenplay annotation for extracting storytelling knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 13.

Ethan Zhou and Jinho D. Choi. 2018. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 24–34, Santa Fe, New Mexico, USA. Association for Computational Linguistics.