CHARACTER ATTRIBUTE EXTRACTION FROM MOVIE SCRIPTS USING LLMS

Sabyasachee Baruah, Shrikanth Narayanan

Signal Analysis and Interpretation Laboratory
University of Southern California
{sbaruah, shri}@usc.edu

ABSTRACT

Narrative understanding is an integrative task of studying characters, plots, events, and relations in a story. It involves natural language processing tasks such as named entity recognition and coreference resolution to identify the characters, semantic role labeling and argument mining to find character actions and events, information extraction and question answering to describe character attributes, causal analysis to relate different events, and summarization to find the main storyline. In this work, we aim to formally operationalize the task of character attribute extraction, motivated by analyzing inclusive character representations and portrayals. We focus on a mix of static and dynamic attribute types that require varying context sizes for their accurate retrieval. We use automated screenplay parsing, entity recognition, and external knowledge bases to collect character descriptions from movie scripts, and explore different prompting strategies (zero-shot, few-shot, and chain-of-thought) to leverage large language models for attribute extraction. [1]

***Index Terms*—** Information Extraction, Character Attributes, Movie Screenplays

1. INTRODUCTION

Narrative understanding presents a complex and challenging task for natural language processing. Several studies have tried to answer the question of what it means to understand a story. Piper et al [1] defined narrative understanding as the task of finding character interactions (dialogue, action, or mention) and their spatiotemporal coordinates. Bamman et al [2] reported on learning persona representations of characters engaged in discourse. Labatut and Bost [3] defined a narrative as a collection of events and how it is told, and studied it from the point of view of character networks. Mostafazadeh et al [4] equated narrative understanding to the story *cloze* test, wherein the task is to complete a story given some sentences as context. Lal et al [5] rephrased narrative understanding as a *Why*-question answering task of explaining the story event. Ammanabrolu et al [6] and Zhang et al [7] focused on extracting the salient event chains. Such a diverse array of approaches shows the lack of consensus on a computational abstraction of the narrative understanding task.

PASSAGE

The SPORTS COMMENTATOR is at the airport and about to interview the world heavyweight boxing champion, APOLLO CREED. Creed is twenty-eight years old. He is a tall, smooth-muscled black man with barely a scar on his light coffee-colored face...

ATTRIBUTES

Type	Value
Profession	Boxer
Accomplishment	Heavyweight Champion
Age	28 years
Physical Appearance	Tall, Smooth-muscled black man, light coffee-colored face
Race	Black

Fig. 1. Example of character attribute extraction from a passage taken from the movie script of *Rocky, 1976* for the character Apollo Creed.

However, a closer look at these studies reveal that certain elements and concepts occur repeatedly. These elements are *characters, events, attributes* and *relations*. **Characters** interact with each other and drive the plot forward. **Events** mark changes in the story world. **Attributes** describe the character's state, and **Relations** causally or temporally connect the events. Much of prior work has focused on characters, events, and relations [8]. In this work, we aim to study character attributes and propose an operational definition for the attribute extraction task. Fig 1 shows an example where we tabulate different attributes of a character from a story passage.

Character attributes track the physical, emotional, and social status of the character in the story. They help us to understand how the character arc develops and progresses. For example, in the movie *The Godfather* (1972), Michael Corleone's character arc takes a villainous turn as he transforms from a war veteran who wants to stay away from his family's crime business to becoming the kingpin of the mafia world. We can track his character's development by following the changes in his goals, emotions, profession, and accomplishments. Attributing character representations allows us to methodically study on-screen portrayals and character depictions [9, 10]. Automatically tracking character attributes can also provide valuable feedback to the writer about whether

[1] Code and data available at github.com/usc-sail/mica-character-attribute-extraction

newer events are coherent and consistent with their characters' backdrop [11]. Additionally, character attributes provide new ways to cluster, curate, filter, and recommend media content, and systematically compare character portrayals across genre and time [12]. Therefore, the attribute extraction task holds great value for researchers, content creators, media professionals and end users. Our contributions are as follows:

1. We operationalize the character attribute extraction task and constrain it to a question answering task to harness the generative ability of large language models (LLMs).
2. We use a data-driven approach to curate and define a taxonomy of character attribute types.
3. We prompt LLMs using zero-shot, few-shot, and chain-of-thought (CoT) methods to find character attributes and analyze the types of errors made.

2. PROBLEM SETUP

We first discuss character attributes and the different ways of classifying them. We define the character attribute extraction task and formulate it as a question answering task to focus our evaluation on the model's extraction and deduction capability.

2.1. Character Attributes

Character attributes define the state of the character. They can be **static or dynamic**. Static character attributes, such as name and gender, seldom change whereas dynamic attributes, such as attire and emotion, and even age, can vary frequently as the story progresses. Attributes can have **small versus large scope** based on the amount of context required for their determination. For example, we need to read several book chapters to ascertain a character's personality, whereas we can find their profession from a single sentence which mentions their job title. In this work, we focus on a fixed set of mostly dynamic and small-scope attributes.

2.2. Task Definition

Character attribute extraction is a form of *open information extraction*. Given a story document, the character attribute extraction task is to produce a set of tuples, each describing some attribute of a specific character. Each tuple has four elements: passage, character, attribute type, and attribute value. **Passage** is the story segment from the document that contains information about the attribute for **character**. **Attribute type** is the type of the attribute and **attribute value** is the particular value of that attribute type portrayed by the character.

For example, given the passage P = "Julia re-appears from the kitchen holding a birthday cake, 17 candles on top. She brings it to John. He eyes her before blowing out the candles.", we can infer that it is John's seventeenth birthday because his birthday cake has seventeen candles. Therefore, a valid character attribute tuple is (P, John, age, 17).

2.3. Question Answering

We constrain the character attribute extraction task to a question answering task by providing the passage, character name, and attribute type as inputs to the model instead of the full story document. The task then becomes to find the attribute value. This can be posed as the question "What is/are the <attribute type> of <character>?" with the passage as context. For the birthday example in Sec 2.2, the question becomes "What is the age of John?".

We constrain the problem so we can only focus on the *extraction* part of the attribute extraction task. We eliminate the covariates of locating the relevant passage from the story document and deciding the valid attribute types. We could use a solution to the question answering task for the broader task by repeatedly applying it to each combination of story segment, character, and attribute type, although it will be computationally expensive. We leave the evaluation of the more general task for future work.

3. CHARACTER ATTRIBUTE EXTRACTION

In this section, we describe the pipeline for collecting the story segments, deciding the attribute types, and extracting the attribute values.

3.1. Collecting Story Segments

We choose movie scripts as our story documents because their semi-structured nature easily lends into creating story segments, and we can find their character list from external knowledge bases. We download publicly available movie scripts from the ScriptsonScreen website[2]. Each script is mapped to its IMDB[3] page that contains the cast information.

We parse the movie scripts to find scene descriptions, dialogue segments, and sluglines, using the screenplay parser of Baruah et al [13]. Scene descriptions are short segments, between 50-200 words, that describe characters and their actions. The content between sluglines defines a movie scene. These are longer segments, between 200-500 words, containing multiple action and dialogue segments. We use the short **description segments** to find small-scope attributes, and use the longer **scene segments** to extract large-scope attributes

3.2. Determining Character Attributes

We use a data-driven approach to decide the attribute types. First, we select description segments that contain the first named mention of a character listed in the top three names

[2]https://scripts-onscreen.com/
[3]https://imdb.com/

of the IMDB cast list. The description segment that first mentions the character usually also describes their attributes. Second, we prompt GPT-3.5[4] in a few-shot manner to list the attribute types described in the description segments. Third, we collect the attribute types from GPT's completions and retain those occurring in the top-90% probability mass.

Attribute	Definition
accomplishments	Significant positive achievements
age	Current age, usually expressed in years
attire	Type of clothes worn, excluding possessions
attitude	Attitude, opinion or evaluation towards something
demeanor	Demeanor, manners, bearing,or outward behavior
emotion	Emotions, feelings, or mental state.
eyes	Eye color, shape or other eye attribute
goal	Goal or motive, what is the character trying to achieve
hair	Hair color, type or other hair attribute
profession	Job title or profession
qualities	Special unique qualities, skills, or abilities
race	Race or ethnicity
voice	Quality, tone, pitch or other voice attribute

Table 1. Definition of attribute types

We end up with 13 attribute types: accomplishments, age, attire, attitude, demeanor, emotion, eyes, goal, hair, profession, qualities, race, and voice. Table 1 contains the brief definitions of each attribute type. All the attribute types have small scope, except for goal. We prompt the larger scene segments to find the character's goal.

3.3. Soft Labeling

Having finalized the attribute types and story segments, we can prompt GPT for the attribute values of the characters mentioned in the segments. However, the attribute distribution is very sparse because most segments describe few or none of the attributes. Directly prompting GPT on a random sample will mostly yield null values because the attribute is not described. We also do not want to prompt all the segments because it will be too expensive. Therefore, following FrugalGPT [14], we use a cascade of two LLMs, the open-source Flan-T5 [15] and GPT-3.5 models, to lower inference costs.

We prompt Flan-T5 in a zero-shot manner to query all <story segment, attribute type, character> triples to answer yes/no if the story segment describes that particular attribute type of the character. We only retain the "yes" triples. The word probability of "yes" gives a soft label between 0 and 1 for the likelihood of the segment describing the character's attribute. It is an empirical measure of the implicitness of the attribute in the segment. Lower the score, higher is the implicitness and harder it is to find the attribute value.

[4]We use the text-davinci-003 GPT-3.5 model

3.4. Prompting

We sample the "yes" triples by the genre of the movie script containing the story segment and its soft label. For each triple, we prompt GPT-3.5 for the attribute value in a zero-shot, few-shot, and CoT manner. Following AnnoLM [16], we self-generate the CoT explanations by prompting GPT-3.5 in a zero-shot manner. The CoT explanations summarize the relevant part of the story segment that describes the attribute, and then conclude with the sentence: "Therefore, the answer is <attribute value>".

4. EVALUATION AND RESULTS

We evaluate all three prompting methods on 680 <story segment, attribute type, character> triples (about 52 triples per attribute type) with the help of four trained raters. We ask the raters if the predicted attribute value is correct or wrong, and to provide an explanation if they choose the latter. The first author double-checked all ratings.

4.1. Prompting Methods

Unlike in reasoning tasks, CoT does not improve performance over few-shot for the attribute extraction task, as shown in Table 2. When the attribute value is clearly evident from the text, stating the answer upfront might be better than reasoning through intermediate steps. CoT might *overextend* its contextual inference, deducing conditions not implied by the story.

Unsurprisingly, both few-shot and CoT perform much better than zero-shot. Therefore, presenting targeted examples besides the attribute definition helps the extraction task. Zero-shot shows good performance (>90%) for age and emotion extraction, suggesting that GPT has a very good understanding of these attributes.

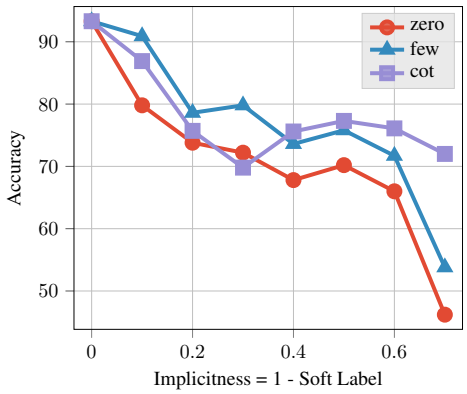

Fig. 2. Accuracy vs Implicitness for different prompting methods. We define implicitness as 1 - soft label.

	Attributes													Micro Acc.	Error Type			
	accomp.	age	attire	attitude	demeanor	emotion	eyes	goal	hair	profession	qualities	race	voice		NF	DC	DA	WM
n	45	53	49	46	50	51	50	48	57	47	58	67	59	680				
zero-shot	64.4^a	**98.1**^{a}	85.7^a	86.7^a	72.0^a	**90.2**^{a}	40.0^b	62.5^a	73.2^b	44.7^b	**73.6**^{a}	70.1^a	64.3^a	71.3^b	**11.5**^{a}	1.9^a	1.8^b	**11.8**^{a}
few-shot	71.1^a	86.8^a	**91.8**^{a}	**93.5**^{a}	**84.0**^{a}	84.3^a	**82.0**^{a}	**66.7**^{a}	70.2^b	**73.9**^{a}	72.2^a	**71.6**^{a}	69.6^a	**78.0**^{a}	4.0^b	2.1^a	**2.5**^{a}	11.6^a
CoT	**71.1**^{a}	90.6^a	87.8^a	93.3^a	78.0^a	74.5^a	81.6^a	60.4^a	**87.7**^{a}	65.2^a	72.2^a	68.7^a	**71.7**^{a}	77.0^a	6.9^c	**2.2**^{a}	1.0^b	11.2^a

Table 2. Per-attribute accuracy and percentage of different error types for zero-shot, few-shot, and CoT. *accomp.* stands for accomplishments. First row contains the counts for each attribute. Values with different superscript letters in a column are significantly different (*t*-test, $p < 0.05$, Bonferroni Correction $n = 3$).

4.2. Attribute Implicitness

CoT does not improve performance over few-shot, but it is still useful. As shown in Fig 2, accuracy decreases for all three prompting methods as the attribute becomes more implicit (defined as $1 -$ soft label, see Sec 3.3). However, CoT is more robust and performs better than few-shot in the highly implicit examples which require a deeper contextual understanding to infer the attribute value. Therefore, an effective ensemble approach would be to switch from few-shot to CoT as the attribute becomes more implicit.

4.3. Error Analysis

We code the explanations provided by the raters when they disagree with the model's response into the following error types: Not Found (NF), Different Character (DC), Different Attribute (DA), and Wrong/Missing (WM). NF means the model is unable to find the attribute value and provides a null response. DC means the model describes some attribute of a different character, and DA denotes that the model describes a different attribute of the given character. WM implies that the model describes the relevant attribute of the given character, but wrongly infers some parts of the attribute value or misses some important facts. The last four columns of Table 2 shows the percentage of each error type for the three prompting methods.

The percentage of DC and WM errors are not significantly different for all three prompting methods. CoT makes more NF-type errors than few-shot but decreases the DA-type errors. This suggests that CoT is stricter than few-shot in providing an answer, but is more faithful to the query and extracts the relevant attribute type. Therefore, CoT might suit applications that are more sensitive to false positive errors better than few-shot, even though it does not improve overall performance.

5. RELATED WORK

Information and relation extraction tasks have traditionally been tackled using tagging-based models [17]. The recent surge of pretrained LLMs have sparked novel generative approaches to extract relation tuples [18]. Sainz et al [19] recast argument extraction as textual entailment, alleviating annotation needs and schema dependencies. Jun et al [20] modeled relation extraction as a semantic matching task by pairing samples with the relation definitions. Wei et al [21] used ChatGPT in a question answering framework for zero-shot information extraction. To the best of our knowledge, similar approaches have not been explored for character attribution in the domain of narrative texts, which is the focus of our study.

6. CONCLUSION

We define the character attribute extraction task, and evaluate different prompting methods using LLMs. CoT does not improve performance over few-shot for attribute extraction, but is more robust to attribute implicitness and remains more faithful to the queried attribute type. Future work should scale the evaluation to more samples, and relax the question-answering constraint to evaluate the long story understanding capability of LLMs.

7. REFERENCES

[1] Andrew Piper, Richard Jean So, and David Bamman, "Narrative theory for computational narrative understanding," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, Nov. 2021, pp. 298–311, Association for Computational Linguistics.

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

[3] Vincent Labatut and Xavier Bost, "Extraction and analysis of fictional character networks: A survey," *ACM*

Computing Surveys (CSUR), vol. 52, no. 5, pp. 1–40, 2019.

[4] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, June 2016, pp. 839–849, Association for Computational Linguistics.

[5] Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian, "TellMeWhy: A dataset for answering why-questions in narratives," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, Aug. 2021, pp. 596–610, Association for Computational Linguistics.

[6] Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J. Martin, and Mark O. Riedl, "Story Realization: Expanding Plot Events into Sentences," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 7375–7382, Apr. 2020.

[7] Xiyang Zhang, Muhao Chen, and Jonathan May, "Salience-aware event chain modeling for narrative understanding," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, Nov. 2021, pp. 1418–1428, Association for Computational Linguistics.

[8] Priyanka Ranade, Sanorita Dey, Anupam Joshi, and Tim Finin, "Computational understanding of narratives: A survey," *IEEE Access*, vol. 10, pp. 101575–101594, 2022.

[9] Yisi Sang, Xiangyang Mou, Mo Yu, Dakuo Wang, Jing Li, and Jeffrey Stanton, "MBTI personality prediction for fictional characters using movie scripts," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 6715–6724, Association for Computational Linguistics.

[10] Victor Martinez, Krishna Somandepalli, Yalda Tehranian-Uhls, and Shrikanth Narayanan, "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)*, Online, Nov. 2020, pp. 4780–4790, Association for Computational Linguistics.

[11] Yijiang River Dong, Lara J Martin, and Chris Callison-Burch, "Corrpus: Detecting story inconsistencies via codex-bootstrapped neurosymbolic reasoning," *arXiv preprint arXiv:2212.10754*, 2022.

[12] Sabyasachee Baruah, Krishna Somandepalli, and Shrikanth Narayanan, "Representation of professions in entertainment media: Insights into frequency and sentiment trends through computational text analysis," *Plos one*, vol. 17, no. 5, pp. e0267812, 2022.

[13] Sabyasachee Baruah, Sandeep Nallan Chakravarthula, and Shrikanth Narayanan, "Annotation and evaluation of coreference resolution in screenplays," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, Aug. 2021, pp. 2004–2010, Association for Computational Linguistics.

[14] Lingjiao Chen, Matei Zaharia, and James Zou, "Frugalgpt: How to use large language models while reducing cost and improving performance," *arXiv preprint arXiv:2305.05176*, 2023.

[15] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al., "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[16] Xingwei He, Zhenghao Lin, Yeyun Gong, A Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al., "Annollm: Making large language models to be better crowdsourced annotators," *arXiv preprint arXiv:2303.16854*, 2023.

[17] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh, "A survey on open information extraction," in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, Aug. 2018, pp. 3866–3878, Association for Computational Linguistics.

[18] Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi, "Gpt-re: In-context learning for relation extraction using large language models," *arXiv preprint arXiv:2305.02105*, 2023.

[19] Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre, "Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning," in *Findings of the Association for Computational Linguistics: NAACL 2022*, Seattle, United States, July 2022, pp. 2439–2455, Association for Computational Linguistics.

[20] Zhao Jun, Hu Yuan, Xu Nuo, Gui Tao, Zhang Qi, Chen Yunwen, and Gao Xiang, "An exploration of prompt-based zero-shot relation extraction method," in *Proceedings of the 21st Chinese National Conference on Computational Linguistics*, Nanchang, China, Oct. 2022, pp. 786–797, Chinese Information Processing Society of China.

[21] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al., "Zero-shot information extraction via chatting with chatgpt," *arXiv preprint arXiv:2302.10205*, 2023.