

Generating Character Descriptions for Automatic Summarization of Fiction

Weiwei Zhang,¹ Jackie Chi Kit Cheung,^{1,2} Joel Oren^{3*}

¹McGill University, Montreal, Canada

²Mila, Montreal, Canada

³Yahoo! Research, Haifa, Israel

weiwei.zhang@mail.mcgill.ca, jcheung@cs.mcgill.ca, joren@yahoo-inc.com

Abstract

Summaries of fictional stories allow readers to quickly decide whether or not a story catches their interest. A major challenge in automatic summarization of fiction is the lack of standardized evaluation methodology or high-quality datasets for experimentation. In this work, we take a bottom-up approach to this problem by assuming that story authors are uniquely qualified to inform such decisions. We collect a dataset of one million fiction stories with accompanying author-written summaries from Wattpad, an online story sharing platform. We identify commonly occurring summary components, of which a description of the main characters is the most frequent, and elicit descriptions of main characters directly from the authors for a sample of the stories. We propose two approaches to generate character descriptions, one based on ranking attributes found in the story text, the other based on classifying into a list of pre-defined attributes. We find that the classification-based approach performs the best in predicting character descriptions.

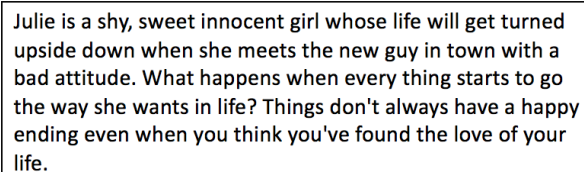
1 Introduction

Reading fiction online has become popular because of its convenience and low cost. Online publishing communities such as Wattpad and Scribd allow users to post and share written works. Readers on such platforms face the problem of deciding which story to read from a large collection of varying quality. One solution is to provide readers with a short summary (a.k.a., synopsis) of the story.

The first research challenge in developing an automatic summarization system for fiction is to establish a framework which defines the expected inputs and desired outputs of the system, as well as a method to evaluate the output quality. A closely related concern is to collect a dataset such that experiments can be carried out under this framework.

There is to our knowledge little work on basic methodological issues in summarizing fiction, and no large-scale datasets of stories and accompanying summaries that encompass large numbers of authors and sub-genres of fiction. The most relevant dataset was created by Mihalcea and Ceylan (2007), consisting of 50 books, each with two manually created summaries. Mihalcea and Ceylan (2007)

*Work done while the author was working at Wattpad
Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



Julie is a shy, sweet innocent girl whose life will get turned upside down when she meets the new guy in town with a bad attitude. What happens when every thing starts to go the way she wants in life? Things don't always have a happy ending even when you think you've found the love of your life.

Figure 1: Sample summary for the story “Starting Out”

adapted summarizers from the news domain to this genre, evaluating using ROUGE scores.

Rather than adapting models and evaluation methods from news summarization, we instead take a bottom-up, data-driven view towards these issues. We start by observing how authors write summaries of their own stories, then use that to inform our modeling and dataset construction decisions. We collected a million stories with accompanying author-written summaries from Wattpad, an online story sharing platform. An example of a summary is shown in Figure 1.

We first analyze a sample of summaries drawn from the dataset to examine the frequency of components that are characteristic of fiction, such as CHARACTER DESCRIPTION and SETTING. We find that character descriptions are the most common component found in summaries, occurring in 81% of the summaries.

Based on this result, we decided to focus on generating character attributes as the first step towards a full fictional summary generation system. We define a framework for generating character descriptions using the story text as input. We examine two sources of information for generating gold-standard character attributes. First, descriptions of characters can be extracted from the summaries and used as a form of gold standard, especially for training supervised machine learning models of attribute inference. However, the extraction process can be noisy, and the labels incomplete, as there are usually other attributes that apply to the characters which are not explicitly mentioned in the summaries. Thus, we surveyed the authors of a sample of stories in order to elicit a set of character attributes to use as a higher-quality gold standard for evaluation.

We compared two models for generating these character descriptions. The first is an extractive model which ranks

the attributes used to describe a character as extracted from its mentions in the story text. The second is an abstractive model that classifies a character into a list of common character attributes, which are not necessarily found in the story text. We examined the use of several feature sets based on the context around character mentions in the story, as well as features relating to the attributes and meta-data about the story. Our results show that the abstractive attribute classifier obtains the highest level of accuracy in generating character descriptions that were important according to story authors, and outperforms a SVM-based method used as the baseline method.

2 Related Work

Automatic summarization of fiction needs deep understanding of fictional stories. Kennedy and Gioia (1983) analyzed an abundance of literature (52 stories, 376 poems, 11 plays) and summarized the elements of fiction. In general, Character, Plot, Setting, Theme, and Point of View are the main elements that fiction writers use to develop a story (Kennedy and Gioia 1983; Stanton 1965; Card 1999; Sartre 1988). There has been a wide span of work on modeling fiction computationally including contributions to literary diction (Underwood and Sellers 2012), character annotation and frames (Elsner 2012; Bamman, Underwood, and Smith 2014; Vala et al. 2015), narrative analysis (Halpin, Moore, and Robertson 2004; Piper 2015), topic modeling (Jockers and Mimno 2013; Goldstone and Underwood 2014), geographic imagination (Wilkens 2013), and folkloristics (Broadwell and Tangherlini 2017).

Character analysis focuses on describing either character personas or their relationships. Bamman, Underwood, and Smith (2014) used a graphical model to infer latent character types (or “personas”), such as clusters of descriptive phrases or actions. Their work provides the possibility of considering other structural and formal elements of narration by adding them into the hierarchical Bayesian model as a separate effect, such as adding the narrative point of view to distinguish first-person narrators and other characters. Flekova and Gurevych (2015) incorporated a range of semantic features with the extracted phrases to predict fictional character personality. Herbelot (2015) constructed representations of named entities in fiction by using a distributional approach that is reweighted by a character’s named entity type. Iyyer et al. (2016) proposed an unsupervised neural network to model the changes of relationships between two characters over the story.

Unlike summarization of news, which can benefit from datasets published by annual Document Understanding Conferences (DUC) evaluations, we are not aware of any large datasets that are publicly available for evaluating methods in fictional text summarization. Project Gutenberg¹ contains a large number of works (both fiction and non-fiction), but does not include summaries. As discussed above, Mihalcea and Ceylan (2007) released a book dataset; however, the dataset does not entirely consist of fiction stories and includes a limited number of books (50 books). Ceylan and

Mihalcea (2009) analyzed the extent to which book summaries can be derived from the book text using cut-and-paste operations. Kazantseva and Szpakowicz (2010) created a system to extract salient descriptive sentences based on syntactic information and shallow semantics for summarizing literary short stories, but the structure of summaries was neglected by the system.

At a high level, similar work exists which aim to infer the structure of descriptions of people in other kinds of texts, including movie scripts (Bamman, O’Connor, and Smith 2014), biographies (Bamman and Smith 2014), Wikipedia articles (Li, Jiang, and Wang 2010) and tweets (Chen et al. 2015). Our work is also related to Guided Summarization (Owczarzak and Dang 2011), in which a domain-specific template helps guide the summarization process, in that we derive a template of summary components for the genre of fiction.

3 An Analysis of Author-Written Summaries of Fiction

In order to better understand the structure of summaries in the domain of fiction, we collected a dataset of stories with author-written summaries, and conducted a manual analysis of the summaries to determine the most frequently occurring components.

3.1 Data Collection

We obtained a dataset² of stories and author-written summaries from Wattpad, a popular online story sharing community. This dataset contains 1,036,965 stories and 942,218 summaries provided by authors.

The average story length is 15,600 words, while the average summary length is 82 words. Wattpad also provided us with a list of the most popular stories, together with metadata about the story (e.g., story ID, story title, general category of the story).

We preprocessed the stories and summaries using BookNLP (Bamman, Underwood, and Smith 2014), in order to extract character mentions and to parse the corpus into dependency trees. We then extracted a list of character attributes and character contexts for each character from their mentions in the story.

Attribute extraction. We define attributes as short descriptive phrases that describe a character in the story. We follow previous work by defining heuristics that use the structure of the dependency parse tree of a sentence containing a character mention to extract attributes (Flekova and Gurevych 2015; Ceylan and Mihalcea 2009). Character mentions were extracted and dependency parsing was performed using BookNLP (Bamman, Underwood, and Smith 2014).

We extract attributes from the context of character mentions where the character is the subject of a copular construc-

¹<https://www.gutenberg.org/>

²Wattpad offers the dataset under a non-commercial academic licence.

tion. For example, in “Bill is not evil”, “is” is the copula, and “not evil” is an attribute describing “Bill”.

Specifically, the attribute extractor extracts the dependency relation COP and its relevant modifiers, including NEG, AMOD, and ADMOD from stories. It identifies instances in which the main character appears with a copula as its NSUBJ argument. For example, “is” is the identified copula in the above example sentence, because “Bill” is a mention of the main character and they have the head word, “evil”. The extractor then extracts the relevant modifiers, indicated by the relations NEG, AMOD, and ADMOD, and the head word of the copular. In the example, “not” and “evil” are extracted. Finally, these words are concatenated to be an attribute according to their original order.

Character context extraction. We also extract other contexts in which the main character is mentioned. In particular, we extracted those sentences where the mention is the argument of a NSUBJ or DOBJ relation. For example, in the sentence “Ethan replied with no emotion whatsoever”, “Ethan” is a main character and the extracted context is “_ replied with no emotion whatsoever”, where _ represents the location of the character mention.

3.2 An Analysis of Summary Components

From the above dataset, we drew a sample of 140 summaries of popular stories from 7 different categories, which we manually annotated with six common components are found in fictional summaries. We pre-collected a set of potential summary components according to the main elements that fiction writers use to develop a story, such as the character, plot, setting, theme etc. (Kennedy and Gioia 1983; Stanton 1965; Card 1999; Sartre 1988). The set of summary components was also refined while we were annotating summaries. From the list of the most popular stories, there are 7 common categories including “Fan Fiction”, “Teen Fiction”, “Romance”, “Werewolf”, “Random”, “General Fiction”, “Science Fiction” and “Others.” We drew 20 summaries of popular stories from each category at random, which we manually annotated with six common summary components.

Type	Ch	Se	Ev	Ho	Co	Ot
Romance	16	1	11	13	0	9
Werewolf	16	0	14	15	0	5
ScienceFiction	16	2	13	11	0	10
Random	18	0	18	19	0	3
FanFiction	15	1	14	16	1	7
GeneralFiction	16	0	14	16	0	6
TeenFiction	17	4	11	12	0	6
Total (140)	114	8	95	102	1	46

Table 1: Frequencies of common summary components: Description of the main characters (*Ch*), Setting (*Se*), Foundational event (*Ev*), Hook (*Ho*), Conclusion (*Co*) and Others (*Ot*) in the summaries of 10 popular stories in each category

Characters: Most summaries contained a description of the

main characters, which included basic characteristics of the characters such as gender, age, and appearance.

Molly is a beautiful intelligent girl who is full of potential.

Setting: The setting refers to the society or general environment in which the main characters are situated.

It has taken countless years and billions of lives, but the Earth has finally achieved a tentative peace. Ruled by a group known as The Council, humanity tries to return to everyday life.

Foundational event: The *foundational event* is an event that happens to the main characters either before the beginning or at the start of the story. It usually appears in the middle of summaries to set up the main character’s background or motivation.

When Alex Heart’s parents die in a tragic car accident he is sent to live with his late parents closest friends the Gately’s in San Diego California.

Hook: A literary *hook* is used to arouse readers’ interest by asking a rhetorical question or by describing an emergency.

Who knows? Maybe Alex will find love unexpectedly in this sunny coastal town.

Conclusion: The conclusion is the ending of the story. Usually, the authors of stories do not provide the conclusion in the summary, because it is considered to be a “spoiler”.

She grows to learn that the world isn[sp] as black and whit[sp]

Others: This category includes other information such as acknowledgments, contact, or copyright information. They often do not relate to the story contents directly.

Credits to the artist. ^_^

The description of the main characters, the foundational event, and the hook are identified as the most common summary components on the basis of the frequency of occurrence (Table 1). It is not surprising that “Conclusion” only appears once, because authors usually do not want the readers to be spoiled. Interestingly, the frequency of “setting” is only eight out of 140, but 4 of them are from the category “TeenFiction.” The reason may be that fiction for teens with a special setting is more attractive.

4 A Framework for Character Attribute Inference

We decided to concentrate on generating summaries started from Character Description, because it is the most common summary component (81.43% of analyzed summaries contained character descriptions) and always appears at the beginning of summaries, preceding the other components. The description of a character typically contains the name and the salient attributes of the character. For example, in the story “Starting Out” presented in Figure 1, the character named Julie has the following salient attributes: *shy, sweet, innocent* and *girl*. Based on this, we propose a framework for predicting attributes for characters in a story.

Given a character in the dataset, c , we generate a set of N candidate attributes, $A = \{a_1, \dots, a_i, \dots, a_N\}$. Note that the set of candidate attributes may be different for different characters. The summarization problem then is to select a subset of A as salient attributes to describe character c .

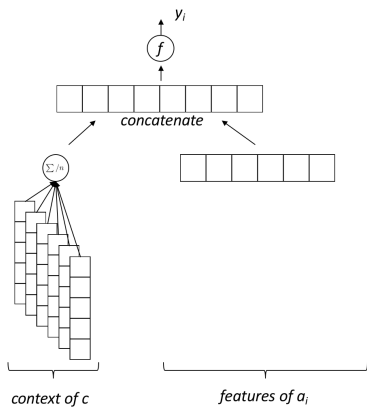


Figure 2: The structure of regression models

In our models, we will formulate this as a machine learning problem, in which the goal is to learn a function, $y_i = f(c, a_i; \theta)$, where y_i represents the predicted salience score of attribute a_i for character c , and θ represents the model parameters. The attributes with the highest scores for each character will be selected.

The methods we describe below differ according to how they generate the candidate set A , the form of the function f , as well as the input features given to f . For all models, we use attributes derived from the author-written summary to derive target scores \hat{y}_i for supervised training of f . For evaluation at test time, we use the attributes we obtained from the questionnaires in Section 6.1 “An Author-Annotated Dataset” as the gold standard.

5 Models

We now describe two models for generating character descriptions. The first is an extractive method which mines the story text for possible descriptions from the context around a character mention. The second is an abstractive method which produces predictions about each character using a fixed list of attributes.

5.1 Model 1: Extractive Attribute Ranking

Our first model casts character description generation as an extractive attribute ranking problem. The candidate set of attributes A for one character c is extracted from the preprocessed story text as well as the author-written summaries described in Section 3.1 “Data Collection”. We can then train a regression model $y_i = f(c, a_i; \theta)$, in which a predicted score close to 1 represents a high likelihood of that attribute being salient, and a score close to 0 that the attribute is not salient. The attributes are then ranked by y_i , and the top k are selected.

Model structure and feature extraction. Our model is a regression model which takes in information about the character, c , and the candidate attribute, a_i , and produces a salience score of that candidate attribute. The model structure is shown in Figure 2.

We feed two types of features as input. The first is a representation of the contexts around the mentions of the character in the story. This representation is constructed by averaging the word embeddings in all of the extracted contexts around mentions of the character into an overall context embedding of c . We describe the context extraction process in more detail in Section 3.1 “Data Collection”.

The second is a set of features related to the candidate attribute a_i extracted from the source story as follows:

- **Dependency relations:** The sequence of incoming dependency relations of the words in the attribute (such as “advmod amod” and “amod nn”) is a syntactic description of the attribute.
- **Frequency of the attribute in current story**
- **Frequency of the attribute in all stories**
- **Frequency of the attribute in all summaries:** This feature shows that how common an attribute is in all summaries in the training set.
- **Chapter number:** This feature represents the position of an attribute in the story.
- **Negation:** The negation shows that whether the attribute contains negation words, such as “not,” “never.” To illustrate, the negation feature of the attribute “not a scientist” is true.
- **Embedding of the attribute:** The attribute embedding, which is the average of word embeddings from the attribute (an attribute may consist of several words), is used to represent the attribute.
- **Embedding of the word before the attribute**
- **Embedding of the word after the attribute**

Target score. We use the character descriptions found in the human-written summaries to construct the target scores for training. If the attribute is found in the summary, it is given a target score of 1.0. Otherwise, the target score is the cosine similarity between the candidate attribute and the most similar attribute in the summary based on word embeddings. If an attribute consists of multiple words, its embedding is the average of all word embeddings in the attribute. Figure 3 shows an example of how we create the target ranking set.

Model details. We use Gradient Boosting Decision Tree (GBDT) as the regression model (Friedman 2002). GBDT is a gradient boosting model, which builds an ensemble of many regression trees the target variable of regression trees can be continuous values with a very limited depth and supports a nonlinear feature combination. We use a squared-error loss: $L = (y - \hat{y})^2$. Word embeddings are pre-trained using word2vec on the entire set of stories (Mikolov et al. 2013).

5.2 Model 2: Abstractive Attribute Classification

A limitation of the extractive approach is that only 16% of the attributes in the author-written summaries can be found

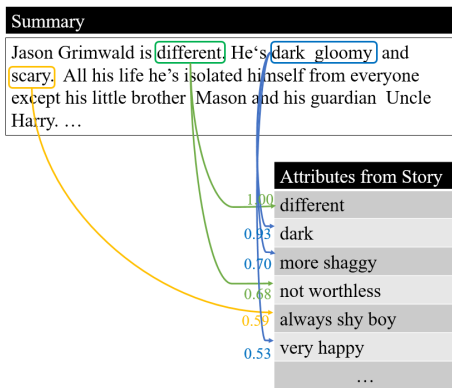


Figure 3: An example of constructing the annotated ranking set

in the story text. In order to address this problem, our second model is an abstractive system that uses a list of candidate attributes drawn from the entire corpus of stories.

Model structure and feature extraction. The structure of the model is shown in Figure 4. The model encodes multiple sources of knowledge about the source story and the attribute into vector representations, and feeds the resulting concatenated vectors into a binary classifier, *CL*, which decides whether the candidate attribute is salient. The inputs of the model are:

- **Full contexts of c :** The first set of input features is the average of word embeddings from all extracted sentences surrounding character mentions of character c , as in the extractive model.
- **Context relevant to a_i :** Unlike the extractive model, a_i may not appear in the context of c . Because the mentions relevant to a_i are more informative than irrelevant mentions, we propose another set of features which is the representation of just the most relevant sentence to the attribute. The most relevant sentence is selected from the extracted sentences according to a similarity measure we define between the attribute and the sentence in Equation 1, $\text{sent_sim}(a_i, \text{sent})$. It is the maximum cosine similarity score between the attribute embedding and the word embeddings from the sentence. We experimented with using the top- k most similar contexts, but initial validation results showed that using just the most similar sentence achieved the best performance:

$$\text{sentence_sim}(a_i, \text{sentence}) = \max_{w \in \text{sentence}} (\text{sim}(a_i, w)) \quad (1)$$

This sentence is then encoded using a long short-term memory (LSTM) network. The hidden layer of the last time step is used as the feature representation.

- **Embedding of a_i :** We use the average embedding of words from the attribute (an attribute may consist of several words).

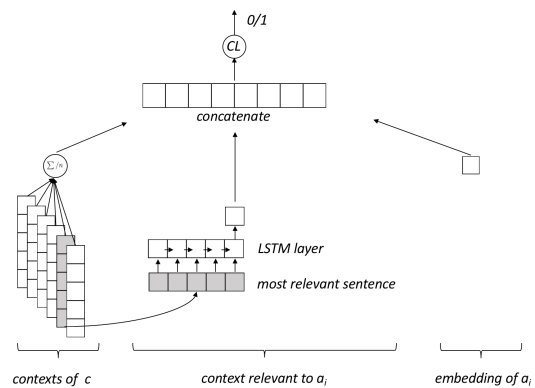


Figure 4: The structure of the abstractive classifier

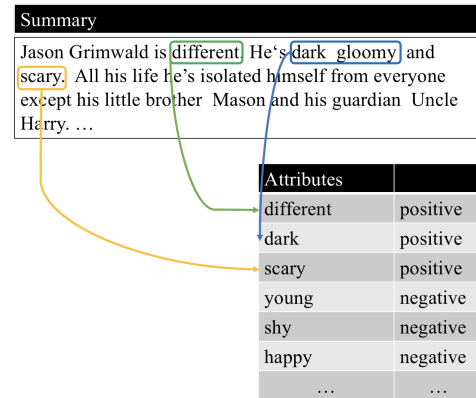


Figure 5: An example of constructing the annotated classification set.

Target score. We chose to frame the abstractive method as a classification task rather than a regression task. The reason for this choice is that when using attributes not found in the story text, we found that similarity to an attribute in the summary was too unreliable as a target score and produced false positive labels. So, an attribute is given a label of POSITIVE if it appears in the summary, and NEGATIVE otherwise. An example is shown in Figure 5.

Model details. We trained the LSTM together with the binary classifier using back-propagation, using a FNN (Feed-forward Neural Network) as the binary classifier *CL*.

6 Experiments

We trained our models using the automatically extracted dataset from the summaries and original articles, and performed final evaluation against the high-quality set of character attributes collected from the story authors, as described in Section 6.1 “An Author-Annotated Dataset”. Because the authors gave each character 4.08 salient attributes on average, we evaluated the top 4 attributes of all automatic methods on the basis of their predicted scores for each character.

Training details. For the extractive models, the automatically created target set contains 97,982 samples (29,818 attributes and 18,100 main characters). The ranking models are prepared in two steps. First, the hyperparameters (the learning rate α , the number of boosting stages $n_estimators$ and the maximum depth of a tree max_depth) of GBDT were tuned on a grid of values ($\alpha \in [0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]$, $n_estimators \in [40, 60, 80, 100, 120, 140, 160, 180, 200]$ and $max_depth \in [3, 5, 7, 9, 11]$) by cross validation. Then, GBDT was trained on the entire automatically extracted ranking set with $\alpha = 0.1$, $n_estimators = 100$ and $max_depth = 5$.

For the abstractive models, the automatically created target classification set contains 52,584 samples (200 attributes and 18,100 main characters). The output of LSTM has 64 dimensions and FNN contains two hidden layers with 64 hidden nodes. The classification model was trained for up to 100 epochs, then we picked the model (5 epochs) with the best performance on the validation data (10% of the automatically extracted classification set). For both models, the word embedding size is set to 64 and pre-trained using word2vec on the entire set of stories.

Baseline. As a baseline method, we trained a SVM to classify the attributes found in the source story associated with the character based on their contexts and then selected four terms according to the score. In the preprocessing stage of the contexts, we lower-cased all words, removed punctuations and stop-words, and selected 10,000 words based on TF-IDF.

Evaluation measure. We use $Recall@K$, macro-averaged across characters, as the measure to evaluate the list of predicted attributes. Suppose the top K salient attributes, A , is selected by a method, while B is the correct set. Then $R = \frac{|A \cap B|}{|B|}$.

6.1 An Author-Annotated Dataset

We used questionnaires to collect our own gold-standard set of character descriptions, because there could be multiple summary-worthy attributes which are not described in the summary by the author. Furthermore, the attribute extractor can introduce errors due to cascading errors in the preprocessing pipeline. We provide these extracted attributes to authors and let them choose the correct ones, or to provide new attributes. We designed and sent questionnaires to more than 2,000 authors of the Wattpad stories we collected. Each questionnaire contained 6 questions about the same main character, as follows:

- **Gender:** This question is single-selection with fixed options, including “female” and “male.”
- **Age:** This question is single-selection with a fixed set of choices, including “child”, “teenager”, “adult” and “elderly.”
- **Species:** This question is single-selection, and the choices are species-related, which are extracted both from the

story and the summary, such as wolf, vampire, and human.

- **Role/Occupation:** This question is single-selection, and its answer choices are role/occupation-related, extracted both from the story and the summary. These choices include teacher, captain, and father.
- **Other:** The remaining choices are not related to any of the previous questions. This question is multiple-selection, consisting of around 20 choices, including alone, rich, and quiet.
- **Additional Features:** This open question allows authors to give us more salient attributes, which are not listed in previous questions.

Interviewees select the options (attributes) that they think are correct and answer a further question as to how likely they would use the selection in the summary (salient attributes).

To generate the questionnaires, we first built an attribute dictionary which consists of 200 high-frequency salient attributes collected from all summaries and manually categorized into different questions. Second, we applied the attribute extractor both to a story and to its summary to extract attributes and assigned the attributes to different questions according to the attribute dictionary. Finally, the questionnaire pages were generated and sent to the authors.

We received 100 valid responses, with each response having 8.07 attributes and 4.08 salient attributes on average. Therefore, we set $K = 4$ in the evaluation metric.

6.2 Results

The recall results of the methods are shown in Table 2. The baseline method only considers the context of candidate attributes. Consequently, it selects the attributes which are correct but indistinctive, such as “person” and “kind” in Table 4.

The extractive models perform at the same level whether using context or not. We speculate that it may be because information is lost due to averaging word embeddings. Because only 16% of the salient attributes appear in the candidates from story, it is not surprising that the extractive models do not perform well.

The abstractive model performs best when both the relevant context and the full contexts are used. The relevant context (relevant sentences) can pinpoint the attributes, such as “alpha” in Table 3. Meanwhile, the full contexts (all context sentences) can handle the overall impression of a character. However, details are missing because the full contexts are the average information of all the extracted sentences. Among the missing are the sentence structure and low frequency, an important piece of detail. Consequently, the predictions only based on full contexts are highly similar.

Our models face several limitations. One is that it selects attributes independently of each other. This results in redundant attributes such as “smart” and “very smart” in Table 3 or conflicts such as “clueless person” and “smart” in Table 4. The other limitation is the reliance of our methods on word relatedness word embedding models such as word2vec, which do not allow reasoning about related by

Models	Features	Recall@4
Baseline	Full contexts	0.0505
Extractive	Attribute	0.0631
	Attribute & Full contexts	0.0631
Abstractive	Attribute & Relevant context	0.1193
	Attribute & Full contexts	0.1104
	Attribute & Relevant context & Full contexts	0.1266

Table 2: The performance of different methods by Recall@4.

Model	Features	Predictions
Baseline	Full contexts	wolf; werewolf ; very smart; kind
Extractive	Attribute	werewolf ; weakling; smart; very smart
	Attribute & Full contexts	werewolf ; weakling; smart; very smart
Abstractive	Attribute & Relevant context	immortal; wolf; different; guy
	Attribute & Full contexts	guy; different; student; werewolf
	Attribute & Relevant context & Full contexts	wolf; different; werewolf ; alpha
Gold	-	perfect ; werewolf ; sweet ; alpha ; happy

Table 3: The predictions and true salient attributes of “Timmy” from the story “Timmy”

Model	Features	Predictions
Baseline	Full contexts	person; kind; afraid; social person
Extractive	Attribute	social person; someone; clueless person; very intelligent
	Attribute & Full contexts	social person; someone; clueless person; smart
Abstractive	Attribute & Relevant context	different; guy; beautiful ; student
	Attribute & Full contexts	student; guy; different; popular
	Attribute & Relevant context & Full contexts	student; different; beautiful ; guy
Gold	-	beautiful ; intelligent ; happy

Table 4: The predictions and true salient attributes of “Hyemi” from the story “I’m Okay With Love”

mutually exclusive attributes. To illustrate, “wolf,” “werewolf,” and “vampire” in Table 3 have such similar representations that they are all wrongly selected for a character.

The recall results of the methods are shown in Table 2. The baseline method only considers the context of candidate attributes. Consequently, it selects the attributes which are correct but indistinctive, such as “person” and “kind” in Table 4.

7 Conclusion

This paper addressed the problem of summarizing online fictional stories. Rather than directly producing an entire summary, we conducted an in-depth investigation of the structure of fictional summaries, created a data set consisting of a large number of stories, and finally proposed several approaches to generate character descriptions.

Our approach focuses on inferring salient attributes to generate the description of main characters. We design and experiment with two different models: one extracts attributes from the source story by ranking candidates; the other classifies using a set list of attributes abstractively. The results show that the abstractive model works better than the extractive model, and both outperform a SVM-based baseline.

While generating character descriptions is a good first step towards full summarization of fiction, we have not tackled other aspects of the process, such as extracting foundational events. Our approach can be extended to these other aspects; however, the machine learning models and feature extraction techniques will need to be specially designed. The last important component, the hook, presents new challenges, which prompt research on user understanding and natural language generation techniques.

Acknowledgments

We thank Wattpad for help in gathering the annotations, as well as our anonymous reviewers for their insightful comments. This work was funded by Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

- Bamman, D., and Smith, N. A. 2014. Unsupervised discovery of biographical structure from text. *Transactions of the Association for Computational Linguistics* 2:363–376.
- Bamman, D.; O’Connor, B.; and Smith, N. A. 2014. Learning latent personas of film characters. In *Proceedings of the*

- Annual Meeting of the Association for Computational Linguistics (ACL)*, 352.
- Bamman, D.; Underwood, T.; and Smith, N. A. 2014. A bayesian mixed effects model of literary character. *ACL (1)* 370–379.
- Broadwell, P. M., and Tangherlini, T. R. 2017. Ghostscope: Conceptual mapping of supernatural phenomena in a large folklore corpus. In *Maths Meets Myths: Quantitative Approaches to Ancient Narratives*. Springer. 131–157.
- Card, O. S. 1999. *Elements of Fiction Writing-Characters & Viewpoint*. Writer's Digest Books.
- Ceylan, H., and Mihalcea, R. 2009. The decomposition of human-written book summaries. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 582–593. Springer.
- Chen, X.; Wang, Y.; Agichtein, E.; and Wang, F. 2015. A comparative study of demographic attribute inference in twitter. *ICWSM* 15:590–593.
- Elsner, M. 2012. Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 634–644. Association for Computational Linguistics.
- Flekova, L., and Gurevych, I. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *EMNLP*, 1805–1816.
- Friedman, J. H. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38(4):367–378.
- Goldstone, A., and Underwood, T. 2014. The quiet transformations of literary studies: What thirteen thousand scholars could tell us. *New Literary History* 45(3):359–384.
- Halpin, H.; Moore, J. D.; and Robertson, J. 2004. Automatic analysis of plot for story rewriting. In *EMNLP*, 127–133.
- Herbelot, A. 2015. Mr Darcy and Mr Toad, gentlemen: distributional names and their kinds. In *Proceedings of the 11th International Conference on Computational Semantics*, 151–161.
- Iyyer, M.; Guha, A.; Chaturvedi, S.; Boyd-Graber, J.; and Daumé III, H. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1534–1544.
- Jockers, M. L., and Mimno, D. 2013. Significant themes in 19th-century literature. *Poetics* 41(6):750–769.
- Kazantseva, A., and Szpakowicz, S. 2010. Summarizing short stories. *Computational Linguistics* 36(1):71–109.
- Kennedy, X., and Gioia, D. 1983. Literature: An introduction to fiction. *Poetry, Drama, and writing*.
- Li, P.; Jiang, J.; and Wang, Y. 2010. Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, 640–649. Association for Computational Linguistics.
- Mihalcea, R., and Ceylan, H. 2007. Explorations in automatic book summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 380–389.
- Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Owczarzak, K., and Dang, H. T. 2011. Overview of the tac 2011 summarization track: Guided task and AESOP task. In *Proceedings of the Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland, USA, November.
- Piper, A. 2015. Novel devotions: Conversional reading, computational modeling, and the modern novel. *New Literary History* 46(1):63–98.
- Sartre, J.-P. 1988. "What is literature?" and other essays. Harvard University Press.
- Stanton, R. 1965. *An introduction to fiction*. Holt, Rinehart and Winston.
- Underwood, T., and Sellers, J. 2012. The emergence of literary diction. *Journal of Digital Humanities* 1(2):1–2.
- Vala, H.; Jurgens, D.; Piper, A.; and Ruths, D. 2015. Mr. Bennet, his coachman, and the archbishop walk into a bar but only one of them gets recognized: On the difficulty of detecting characters in literary texts. In *EMNLP*, 769–774.
- Wilkens, M. 2013. The geographic imagination of civil war-era american fiction. *American Literary History* 25(4):803–840.