Supervised learning of character relations in novels

Ben Kybartas
McGill University
845 Sherbrooke St. W
Montréal, QC, Canada
ben.kybartas@mail.mcgill.ca

Talia Wise
McGill University
845 Sherbrooke St. W
Montréal, QC, Canada
talia.wise@mail.mcgill.ca

Abstract

We use NLP techniques to classify character relationships in novels. Character relationships are key elements of fictional works and analyzing them remains an open challenge in digital humanities. Approaches to this analysis typically involve some form dynamic of sentiment or emotion analysis of character's affinities towards each other. We use the Book-NLP pipeline to extract character instances and use an annotated dataset of character relationships to train a supervised learning classifier. We then compare this bag-of-words classification to a classifier using SentiWordNet as features and to an analysis of classification success on authors not present in the training data. We conclude that sentiment analysis is at least as good at relationships classification as ngrams, and that ngram classification works better than random irregardless of author, but that certain authors use different writing styles and vary in predictability.

1 Introduction

Understanding character relations in text is an important step in natural language understanding for literature. It has been proposed as a form of summarization, stance detection and analysis (Mohammad, 2013), a way to challenge or verify literary analysis (Elson et al., 2010) and is an open problem in digital humanities (Iyyer et al., 2016). Existing work on analyzing characters in narratives has focused on identifying character roles, or sentiments (Mohammad, 2013) but such studies importantly ignore the relations and interactions between characters that form the core of the novel (Srivastava et al., 2015). Additional complexities arrive in the sense that character relations are not necessarily static, and may change significantly over the course of a novel (Chaturvedi et al., 2016).

In this project, we aim to investigate the classification of character relations. We hypothesize that, similar to sentiment analysis (Liu, 2012), the words used in sections of the book where two characters are mentioned can be used as indicators of their relationship and furthermore that relationship types are universally detectable regardless of author. Our proposal is thus to take a labeled set of character relationships and use it to train a classifier. To create our dataset we use the annotated character relation dataset from Massey et. al. (Massey et al., 2015). The dataset classifies character relations from 109 narratives with coarse-grain (professional, social, familial) and fine-grain (brother/sister, lover, friend, etc.) labels. The dataset further provides the characters' affinity towards each other (positive, negative, neutral) and classifies whether the relationship is dynamic or static over the course of the narrative. For our approach, we chose to extract sections from paragraphs where the two characters are referenced, which are then converted into n-grams, and used alongside the labels from the data-set to train a classifier. We further used sentiment analysis techniques to attempt to classify character affinity. And additionally, compared the n-gram results to classification on singled out authors in order to ascertain the impact of authors style on descriptions of character relationships. We use Book-NLP (Bamman et al., 2014) pipeline to resolve character co-references and tokenize/lemmatize the narratives we are testing. Secondly, we map the output Book-NLP onto the dataset and extract the segments of the narrative where the two characters are co-referenced. Last, we convert the segments into n-grams and train a support vector machine with a linear kernel, using several other classifiers for comparison.

2 Related Work

Early work on character analysis in narratives tends to focus on the roles and emotions of individual characters, rather than relationships. In Mohammad's work (Mohammad, 2013), sentiment analysis is used to track the current emotional state of the narrative, which is considered to apply to all characters at that point. As in several other works in this section, Mohammad's work is primarily unsupervised, relying instead on an existing emotion lexicon to analyze segments of text, and their focus is on overall mood rather than character relations.

Chaturvedi et. al. (Chaturvedi et al., 2017) and Iyyer et. al. (Iyyer et al., 2016) also utilize unsupervised learning, but specifically for understanding character relationships and the way those relationships change over the course of the narrative. Chaturvedi et. al. extract semantic information regarding two characters, such as actions, frames, and text, and use a hidden markov model to predict the current relationship between both characters. Iyyer et. al.'s [1] work uses deep-learning techniques on segments of text between two references to characters in order to discover both relationship *descriptions* (types of possible relationships) and *trajectories* (the particular relationships over time between two characters). Our experiment is supervised, however the dataset we use only indicates a simple yes/no of whether a relationship is dynamic or not, and the inter-annotator agreement on this tag is only 20% when corrected for chance. As such, we do not focus on the evolution of relationships.

Work by Srivastava et. al. attempts to extract character relations as a network, referring to the result as a form of "social community" (Srivastava et al., 2015). Their work is unique in that it considers more than two characters and the types of social structures that can occur (examples include love triangles, cliques and common enemies). As above, Srivastava's work is unsupervised, using a clustering model and uses movie summaries instead of novels. This work differs from ours in that we only consider relations between two characters. While we could still formulate a social network from our results, Srivastava's work seems to indicate that taking multiple characters into account can better show certain relationships, for example one character may hate another only because they are both in love with a third character, a feature which may be hard to extract when considering only two characters at once. Similarly, work by Elson et. al. (Elson et al., 2010) also extracted social networks from novels, however their work notes the frequency rather than the type of relationship between characters.

In light of recent work on stance detection, which has been mostly done on twitter data, we show that similar techniques can be used to detect character affinity in novels. Previous work on stance detection such as Mohammad et. Al (Mohammad et al., 2016) and Aker et. Al.(Aker et al., 2017) find that informed feature extraction and standard classifiers work better than other more complicated techniques for this task. In their case, the extra features extracted are information about the twitter user for each post, and extra-textual information about the tweet such as use of emojis, urls and hashtags, none of which are present in novels.

Our work is different from the above works on character relations in its use of the character annotation dataset (Massey et al., 2015) and its use of supervised learning. Likewise, we do not take into account change in character relations over the course of a narrative in our model, because we are only drawing sections of text on paragraph or sentence level where presumably character relationships are static. Because of this, our model can be compared to the work on stance detection, but rather than testing sentences that address specific predetermined target issues, we make the assumption that sections of text with two characters indicate the stance of each character towards the other.

[1] Both of these papers are from the same project.

3 Method

3.1 Pre-processing

Our main focus in pre-processing was to extract the segments of text which contain references to two characters in the novel. The first challenge was the extraction of characters as well as their co-references in each of the novels. We first downloaded all the novels used in Massey et. al.'s dataset from the Project Gutenberg website.[2] We used the Book-NLP (Bamman et al., 2014) pipeline to process each novel individually. Book-NLP is a well-regarded tool (used in almost all papers described in the related works section) that extracts character names, resolves co-references, and handles tokenization, part-of-speech tagging and lemmatization of books. Unfortunately, Book-NLP is often too conservative with co-reference resolution in order to preserve accuracy, causing many occurrences of pronouns not to be resolved.

Following this, we needed to extract the character names from the Book-NLP output and match them to the character names used in the annotated dataset. Book-NLP provides all the references for a character (e.g. Anna, Karenina and Anna Karenina) in a novel, but these occasionally do not match the dataset we used. Thus, we implemented a basic character processor that attempts to find first and last names, and predict the gender of a character given a set of names. We then process both the names retrieved from Book-NLP and the names from the dataset and try to match them. If we cannot match the characters, we discard that entry from the dataset.

As a final step, for each of the character relations we were able to match to the Book-NLP output, we then extract all paragraph level segments of text where two characters are referenced. For each segment, we only take the lemmatized tokens, and do not include the character references to avoid the classifier learning the names of characters. We then labeled each segment with the corresponding data from the dataset. The result, was that we obtained a dataset of lines which relate two characters, classified according to affinity, coarse-grained relationship, fine-grained relationship and change over time. Subsequently, we extracted various authors work from the main dataset in order to create data and tags subdivided by author. Our intent was to test each of the classifiers on each of these four labels, to note if certain types of relationships are easier or harder to classify according to our data. Note that for the affinity classification, we only classified for static relationships, as it was not clear how a dynamic relation changed the affinity, nor was that provided by the annotators.

3.2 N-Gram Analysis

For the first test, we extracted the (1,2)-grams for each phrase, removing stopwords, using tf-idf smoothing and removing any words which occurred less than twice[3]. The results were obtained by testing using 3-fold cross-validation, obtaining the average of the weighted F1 scores as our final result. A uniform random classifier was used as the baseline in all cases and the main results were obtained using a support vector machine with a linear kernel. We also compared these results to Naïve Bayes and Logistic Regression classifiers.

3.3 Sentiment Analysis

The second analysis involved extracting the average sentiment for each phrase using SentiWordNet. For this experiment, we were largely interested in examining the impact of sentiment analysis on classifying the affinity of character relations, and thus ignore its impact on change, coarse and fine-grained character relations. Following the set up in Ohana and Tierney's work (Ohana and Tierney, 2009), we tested using two approaches. For the first approach, for each phrase we obtained the sentiment score for each token by taking the positive sentiment and subtracting the negative sentiment. The final sentiment was the average of each of these scores. Since SentiWordNet is based on word senses, for disambiguation we simply took the most-frequent sense for each word[4]. We then "classified" the result as negative if the average was less than -0.01, positive if the result was greater than 0.01 and neutral otherwise. The bounds were set quite low due to the fact that most phrases used largely neutral words, with only a few words

[2] Note that the dataset we used also had a number of annotated plays, however the Book-NLP pipeline we used could not process these plays so they were disregarded.

[3] This setup was based upon the results from assignment 1

[4] Based on our results from assignment 3

having a clear positive or negative sentiment associated with them.

For the second approach, we took the averaged positive, negative and objective scores for each phrase as the features (using most-frequent sense for disambiguation). These were then used to train the same classifiers from the n-gram experiment using 3-fold cross validation to obtain an average weighted F1 score.

3.4 Authorship

The third analysis involved extracting different groups of novels from the dataset in order to test the effects of author and style on character relations. This experiment responds to the open question of whether authorship significantly affects descriptions of character relations in novels. (Bamman et al., 2014) By excluding each unique authors work from the training set, this experiment also tests whether the classifiers are learning author and genre specific character relations or universal indicators of relationships.

We extracted respectively the texts of each of ten authors with highest number of novels in the dataset. We only used these authors because we wanted to make sure we where categorizing on the level of author rather than the level of an individual book. We then trained classifiers according to the setup from the n-gram test in order to be able to compare their results and test whether a classifier trained on all other authors could predict each individual author's character relations equally well.

4 Results

The results for the n-gram classification experiment can be seen in Table 1. While we take the support vector machine results as our main results, we left several other classifiers in for comparison. It was found in general that classification outperformed a random baseline, but would rarely rise above 70% average accuracy. This might be in part because inter-annotator agreement in our dataset, corrected for chance, ranges between 20% and 80% for the four relation categories, suggesting the dataset may contain mistakes and that the task is very difficult even for humans. (Massey et al., 2015) It was further noted that even testing more advanced classifiers did not significantly improve the results. Similar results have been noted in stance detection research. In Mohammad et. al.'s summary of the SemEval16 stance detection competition, they note "that the SVM-ngrams baseline also performed very well, using only word and character n-grams in its classifiers [emphasizing that] the community is still a long way from an established set of best practices." (Mohammad et al., 2016). This led to some, like Aker et. al. (Aker et al., 2017) to "[call] into question the value of using complex sophisticated models for stance classification without first doing informed feature extraction." Thus it is interesting to note that the results and concerns from the field of stance detection are very similar to our own.

For the SentiWordNet tests, the results of which may be seen in Table 3, we noted similar results to the n-gram tests. However in this case most classifiers obtained near uniform scores, with less deviation across folds. Note that the The "SentiWordNet Average" score was simply the results based on our first experiment, where we classified based on the average sentiment of each phrase. This result is unique in that it obtained slightly above random (though by no means exceptional) scores without any learning, and only making use of the SentiWordNet corpus. It was also notable that the only features used for this classification were the three feature scores per phrase. The results obtained do seem to suggest that, in the context of this experiment, sentiment scores are at least as reliable as n-grams as a measure of character affinity.

The results for the authorship tests, seen in Table2 show that in all four categories, while the SVM classifier is worse at predicting relationship categories on an author whose work was not in the training set, it still predicts better than average results. This shows that authors do write about relationships differently, but that there are also recognizable words and bigrams that indicate affinity, relationship category and even change in a relationship over time. Moreover, while all the unique-author tests show better than average results in all categories, the extent to which they where better than average varies widely, which shows that some authors write about relationships in a more standard or predictable way than others. Additionally, because some authors are much more represented then others in the dataset,

	Affinity	Coarse-Grained	Fine-Grained	Change
Random Baseline	0.38	0.50	0.09	0.62
Naïve Bayes	0.59	0.50	0.20	0.87
Logistic Regression	0.59	0.57	0.20	0.87
Linear SVM	0.61	0.58	0.22	0.86

Table 1: Results for classification based upon n-grams, in terms of weighted F1 Score.

	Affinity	Coarse-Grained	Fine-Grained	Change
Random Baseline	0.38	0.41	0.06	0.57
Naïve Bayes	0.49	0.42	0.14	0.74
Logistic Regression	0.49	0.48	0.16	0.73
Linear SVM	0.50	0.50	0.17	0.71

Table 2: Results for unique-author classification based upon n-grams, in terms of weighted F1 Score.

	Affinity
Random Baseline	0.38
SentiWordNet Average	0.49
Naïve Bayes	0.59
LR	0.59
Linear SVM	0.59

Table 3: Classification results using SentiWordNet in terms of weighted F1 Score.

this finding suggests that the ngram and sentiment classifiers may be learning relationship classifications that are biased towards the most represented authors.

5 Discussion and Conclusion

In all results, there are several open problems which were not considered. The first authorial bias, discussed by Liu (Liu, 2012), where both language and sentiment may actually be part of the author's bias towards a character, as opposed to the character's personal bias towards another character. As noted by Liu, this is still an open problem. Similarly, in an informal conversation with Dr. Andrew Piper, it was noted that language in stories generally tends to be far more objective when dealing with characters, making valuable sentiment analysis difficult. Lastly, this work did not factor in temporality, a key feature in many of the related works and an important element of narratives. Future work would aim to integrate temporality as a feature of analysis, as well as explore further methods for adapting to author writing styles and biases. There were also several areas where data may have been incorrect, such as in incorrect annotations, incorrect character assignment, and incorrect data from the Book-NLP pipeline.

Classification of character relations is an important challenge in the field of computational narrative, and shares properties with NLP fields such as sentiment analysis, summarization and understanding. In this paper, we explored a supervised learning approach to classifying relations, using an available hand-annotated dataset. We processed the existing novels from the dataset and extracted sections of the novel where two characters were referenced. From this, we performed both an n-gram and sentiment based classification test, noting that we were able to outperform a random baseline, but generally obtained similar results across all classifiers. We also noted that, for affinity, both n-gram and sentiment analysis yielded similar results. And that authorship has a significant affect on the words used in all forms of character relationships, but that there are still universal indicators. Similar to the results from stance detection, we propose that future work should focus more deeply on feature extraction, such as the use of temporality, or methods to account for authorial writing style and bias.

6 Contributions

Related work was researched by both Talia and Ben. Ben acquired the data set and related books from Project Gutenberg. Talia handled the processing of the books using Book-NLP. Ben handled the extraction of text and characters, as well as the creation of the final dataset. Ben worked on the main classification tests, as well as the SentiWordNet classification tests. Talia handled the extraction of author specific test sets. Talia handled the author specific classification tests. Talia and Ben both worked on testing, and writing of the paper was similarly shared.

Acknowledgments

We thank Dr. Andrew Piper for his assistance and conversations surrounding co-reference resolution in literary works, as well as his comments on the expected difficulties of our proposal.

References

Ahmet Aker, Leon Derczynski, and Kalina Bontcheva. 2017. Simple open stance classification for rumour analysis. *CoRR*, abs/1708.05286.

David Bamman, Ted Underwood, and Noah A Smith. 2014. A bayesian mixed effects model of literary character. In *ACL*, pages 370–379.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daume III, and Chris Dyer. 2016. Modeling evolving relationships between characters in literary novels.

Snigdha Chaturvedi, Mohit Iyyer, and Hal Daumé III. 2017. Unsupervised learning of evolving relationships between literary characters. In *Association for the Advancement of Artificial Intelligence*.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 138–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *North American Association for Computational Linguistics*.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Philip Massey, Patrick Xia, David Bamman, and Noah A. Smith. 2015. Annotating character relationships in literary texts. *CoRR*, abs/1512.00728.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California, June. Association for Computational Linguistics.

Saif Mohammad. 2013. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. *CoRR*, abs/1309.5909.

Bruno Ohana and Brendan Tierney. 2009. Sentiment classification of reviews using sentiwordnet.

Shashank Srivastava, Snigdha Chaturvedi, and Tom M. Mitchell. 2015. Inferring interpersonal relations in narrative summaries. *CoRR*, abs/1512.00112.