Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Modeling Evolving Relationships Between Characters in Literary Novels

Snigdha Chaturvedi[1], Shashank Srivastava[2], Hal Daumé III[1], Chris Dyer[2]
[1]University of Maryland, College Park [2]Carnegie Mellon University
snigdhac@cs.umd.edu, ssrivastava@cmu.edu, hal@cs.umd.edu, cdyer@cs.cmu.edu

S1: Tom falls in love with Becky Thatcher, a new girl in town, and persuades her to get "engaged" to him.
S2: Their romance collapses when she learns that Tom has been "engaged" before—to a girl named Amy Lawrence.
S3: Shortly after being shunned by Becky, Tom …
S4: …Tom gets himself back in Becky's favor after he nobly accepts the blame for a book that she has ripped.
S5: Meanwhile, Tom goes on a picnic to McDougal's Cave with Becky and their classmates.

Figure 1: Sample sentences from a narrative depicting evolving relationship between characters: Tom and Becky. The relationship changes from cooperative (+) to non-cooperative (-) and then back to cooperative (+). '…' represent text omitted due to space constraints.

Abstract

Studying characters plays a vital role in computationally representing and interpreting narratives. Unlike previous work, which has focused on inferring character roles, we focus on the problem of modeling their relationships. Rather than assuming a fixed relationship for a character pair, we hypothesize that relationships temporally evolve with the progress of the narrative, and formulate the problem of relationship modeling as a structured prediction problem. We propose a semi-supervised framework to learn *relationship sequences* from fully as well as partially labeled data. We present a Markovian model capable of accumulating historical beliefs about the relationship and status changes. We use a set of rich linguistic and semantically motivated features that incorporate world knowledge to investigate the textual content of narrative. We empirically demonstrate that such a framework outperforms competitive baselines.

1 Introduction

The field of computational narrative studies focuses on algorithmically understanding, representing and generating stories. Most research in this field focuses on modeling the narrative from the perspective of (1) events or (2) characters.

Popular events-based approaches include scripts (Schank and Abelson 1977; Regneri, Koller, and Pinkal 2010), plot units (Goyal, Riloff, and Daumé III 2010; McIntyre and Lapata 2010; Finlayson 2012; Elsner 2012), temporal event chains or schemas (Chambers and Jurafsky 2008; 2009), and the more recent bags of related events (Orr et al. 2014; Chambers 2013; Cheung, Poon, and Vanderwende 2013).

The alternate perspective attempts to understand stories from the viewpoint of characters and relationships between them. This perspective explains the set of observed actions using characters' *personas* or *roles* and the expected behavior of the character in that role (Valls-Vargas, Zhu, and Ontañón 2014; Bamman, O'Connor, and Smith 2013; Bamman, Underwood, and Smith 2014; Elson 2012). Recent work has also focused on constructing social networks, sometimes signed networks, to model the relationships between characters (Agarwal et al. 2013; 2014; Elson, Dames, and McKeown 2010; Krishnan and Eisenstein 2015).

The work presented in this paper aligns with the second perspective. We address the problem of modeling relationships between characters in literary fiction, specifically novels. Existing work mentioned above, models each character as assuming a single narrative role, and these roles define the relationships between characters and also govern their actions. While such a simplified assumption provides a good general overview of the narrative, it is not sufficient to explain *all* events in the narrative. We believe that in most narratives, relationships between characters are not static but *evolve* as the novel progresses. For example, consider the relationship between *Tom* and *Becky* depicted in Fig. 1 which shows an excerpt from the summary [1] of *The Adventures of Tom Sawyer*. For most of the narrative (and its summary), the characters are participants in a romantic relationship, which explains most, but not all, of their mutual behavior. However, we can observe that their relationship was not static but evolving, driving nature of the characters' actions. In this particular case, the characters presumably start as lovers (sentence S1 in the figure), which is hinted at by (and explains) becoming engaged. The relationship sours when Tom reveals his previous love interest (S2 and S3). However, later in the narrative they reconcile (S4 and S5). A model that assumes a fixed romantic relationship between characters would fail to explain their behaviors during the phase when their relationship was under stress.

Therefore, we assume that the relationship between characters evolves with the progress of the novel and model it as a sequence of latent variables denoting relation state. In this work, we take a coarse-grained view, and model rela-

[1]SparkNotes Editors. SparkNote on The Adventures of Tom Sawyer. SparkNotes LLC. 2003. http://www.sparknotes.com/lit/tomsawyer/

tion states as binary variables (roughly indicating *cooperative/non-cooperative* relation). For instance in Fig. 1, the relationship between Tom and Becky can be represented by the sequence ⟨cooperative, non-cooperative, cooperative⟩. Given a narrative and a pair of characters appearing in it, we address the task of **learning relationship sequences.** The narrative fragment of interest for us is represented by the set of sentences in which the two characters of interest appeared together, arranged in the order of occurrence in the narrative.

To address this problem we propose a semi-supervised segmentation framework for training on a collection of fully labeled and partially labeled sequences of sentences from narrative stories. The structured prediction model in the proposed framework attempts to model the 'narrative flow' in the sequence of sentences. Following previous work (Propp 1968; Bamman, Underwood, and Smith 2014), it incorporates the linguistic and semantic information present in the sentences by tracking events and states associated with the characters of interest and enhances them with world knowledge (Feng et al. 2013; Liu, Hu, and Cheng 2005; Wilson, Wiebe, and Hoffmann 2005). We demonstrate the strength of our structured model by comparing it against an unstructured baseline that treats individual sentences independently. Our main contributions are as follows:

- We formulate the novel problem of relationship modeling in narrative text as a structured prediction task instead of a categorical binary or multi-class classification problem.
- We propose rich linguistic features that incorporate semantic and world knowledge.
- We present a semi-supervised framework to incorporate the narrative structure of the text and empirically demonstrate that it outperforms competitive baselines on two different but related tasks.

2 Relationship Prediction Model

In this section we describe our relationship-modeling framework. Given the narrative text in form of a sequence of sentences (in which the two characters of interest appear together), $\mathbf{x} = \langle x_1, x_2, \ldots, x_l \rangle$, we address the problem of segmenting it into non-overlapping and semantically meaningful segments that represent continuities in relationship status. Each segment is labeled with a single relationship status $r_j \in \{-1, +1\}$ hence yielding a relationship sequence $\mathbf{r} = \langle r_1, r_2, \ldots, r_k \rangle k \leq l$. We use a second order Markovian latent variable model for segmentation that is embedded in semi-supervised framework to utilize varying levels of labeling in the data. We now describe our segmentation model and the semi-supervised framework in detail.

2.1 Segmentation Model

This model forms the core of our framework. It assumes that each sentence in the sequence is associated with a latent state that represents its relationship status. While making this assignment, it analyzes the content of individual sentences using a rich feature set and simultaneously models the flow of information between the states by treating the prediction task as a structured problem. We utilize a second-order Markov model that can remember a long history of the relationship between the two characters and collectively maximizes the following linear scores for individual sequences:

$$\text{score} = \sum_i \mathbf{w}\boldsymbol{\phi}(\mathbf{x}, y_i, y_{i-1}, y_{i-2})] \tag{1}$$

where $\mathbf{x}$ is the input sequence and y_i denotes the latent state assignment of its i^{th} sentence to a relationship segment. Individual y_is collectively yield the relationship sequence, $\mathbf{r}$ (by collapsing consecutive occurrences of identical states). $\boldsymbol{\phi}$ represents features at the i^{th} sentence that depend on the current state, y_i, and the previous two states, y_{i-1} and y_{i-2}, and $\mathbf{w}$ represents their weights. Our second order Markov assumption ensures continuity and coherence of characters' behavior within individual relationship segments.

The linear segmentation model proposed here is trained using an averaged structured perceptron (Collins 2002). For inference, it uses a Viterbi based dynamic programming algorithm. The extension of Viterbi to incorporate second order constraints is straightforward. We replace the reference to a state (in the state space $|Y|$) by a reference to a state pair (in the two fold product space $|Y| \times |Y|$). Note that this precludes certain transitions while computing the Viterbi matrix, viz.: if the state pair at any point in narrative, t, is of the form (s_i, s_j), then the set of state pair candidates at $t{+}1$ only consists of pairs of the form (s_j, s_k). Incorporating these constraints, we compute the Viterbi matrix and obtain the highest scoring state sequence by backtracking as usual.

2.2 Semi-supervised Framework

The above segmentation model requires labeled $(\mathbf{x}, \mathbf{y})$ for training. However, given the nature of the task, obtaining a huge dataset of labeled sequences can be time consuming and expensive. On the other hand, it might be more convenient to obtain partially labeled data especially in cases in which only a subset of the sentences of a sequence have an obvious relationship state membership. We, therefore, propose a semi-supervised framework, which can leverage partial supervision for training the segmentation model. This framework assumes that the training dataset consists of two types of labeled sequences: fully labeled, in which the complete state sequence is observed $y_i \forall i \in \{1 \ldots l\}$ and partially labeled, in which some of the sentences of the sequence are annotated with y_i such that $i \subset \{1 \ldots l\}$.

Algorithm 1 Training the semi-supervised framework

1: **Input:** Fully F and partially P labeled sequences; and T: number of iterations
2: **Output:** Weights $\mathbf{w}$
3: **Initialization:** Initialize $\mathbf{w}$ randomly
4: **for** t : 1 to T **do**
5: $\quad \hat{\mathbf{y}}_j = \arg\max_{\mathbf{y}_j} [\mathbf{w_t} \cdot \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})_j] \; \forall j \in P$ such that $\hat{\mathbf{y}}_j$ agrees with the partial annotated states (ground truth).
6: $\quad \mathbf{w_{t+1}}$ = StructuredPerceptron($\{(\mathbf{x}, \hat{\mathbf{y}})_j\} \in \{P, F\}$)
7: **end for**

This framework uses a two step algorithm (Algorithm 1) to iteratively refine feature weights, $\mathbf{w}$, of the segmentation model. In the first step, it uses existing weights, $\mathbf{w_t}$, to assign state sequences to the partially labeled instances. For state assignment we use a constrained version of the Viterbi algorithm that obtains the best possible state sequence that agrees with the partial ground truth. In other words, for the annotated sentences of a partially annotated sequence, it precludes all state assignments except the given ground truth, but segments the rest of the sequence optimally under these constraints. In the second step, we train the averaged structured perceptron based segmentation model, using the ground truth and the state assignments obtained in the previous step, to obtain the refined weights, $\mathbf{w_{t+1}}$. Similar approaches have been used in the past (Srivastava and Hovy 2014).

3 Feature Engineering

This section describes the features used by our model.

3.1 Pre-processing

We first pre-processed the text of various novel summaries to obtain part-of-speech tags and dependency parses, identify major characters and perform character names clustering (assemble 'Tom', 'Tom Sawyer' etc.) using the Book-nlp pipeline (Bamman, Underwood, and Smith 2014). However, the pipeline, designed for long text documents involving multiple characters, was slightly conservative while resolving co-references. We augmented its output using coreferences obtained from the Stanford Core NLP system (Manning et al. 2014). We then obtained a frame-semantic parse of the text using Semafor (Das et al. 2014). We also obtained connotation (Feng et al. 2013), sentiment (Liu, Hu, and Cheng 2005) and prior-polarity (Wilson, Wiebe, and Hoffmann 2005) of words when needed during feature extraction. Finally, given two characters and a sequence of pre-processed sentences, in which the two appeared together, we extracted the following features for individual sentences.

3.2 Content features

These features help the model in characterizing the textual content of the sentences. They are based on the following general template which depends on the sentence, x_j, and its state, y_j: $\phi(x_j, y_j) = \alpha$ if the current state is y_j ; 0 otherwise where, $\alpha \in$ F1 to F33, where F1 to F33 are defined below.

1. Actions based: These features are motivated by Vladimir Propp's Structuralist narrative theory (Propp 1968) based insight that characters have a 'sphere of actions'. We model the actions affecting the two characters by identifying all verbs in the sentence, their agents (using 'nsubj' and 'agent' dependency relations) and their patients (using 'dobj' and 'nsubjpass' relations). This information was extended using verbs conjunct to each other using 'conj'. We also used the 'neg' relation to determine the negation status of each verb. We then extracted the following features:

- *Are Team [F1]:* This binary feature models whether the two characters acted as a team by indicating if the two characters were agents (or patients) of a verb together.
- *Acts Together [F2-F7]:* These features explicitly model the behavior of the two characters towards each other using verbs for which one of the characters was the agent and the other was patient. These six numeric features look at positive/negative connotation, sentiment and prior-polarity of the verbs while considering their negation statuses (see Pre-processing for details).
- *Surrogate Acts Together [F8-F13]:* These are high-recall features that analyze actions for which a character was an implicit/subtle agent or patient. For example, Tom is not the direct patient of *shunned* in S3 in Fig. 1. We define a set of six surrogate features that, like before, consider connotations, sentiments and prior-polarities of verbs (considering negation). However, only those verbs are considered which have one of the characters as either the agent or the patient, and occur in sentences that did not contain any other character apart from the two of interest.

Type	Frame[Frame-elements]
Negative	*killing* [*killer, victim*]
	attack [*assailant, victim*]
Positive	*forgiveness* [*judge, evaluee*]
	supporting [*supporter, supported*]
Ambi-guous	*cause bodily experience* [*agent, experiencer*]
	friendly or hostile [*side_1, side_2, sides*]
Rela-tionship	*kinship* [*alter, ego, relatives*]
	subordinates_superiors [*superior,subordinate*]

Table 1: Frame samples used by 'Semantic Parse' features.

2. Adverb based: These features model narrator's bias in describing characters' actions by analyzing the adverbs modifying the verbs identified in 'Action based' features (using 'advmod' dependency relations). For example, in S4 in Fig. 1 the fact that Tom *nobly* accepts the blame provides an evidence of a positive relationship.

- *Adverbs Together [F14-F19]* and *Surrogate Adverbs Together [F20-F25]:* Six numeric features measuring polarity of adverbs modifying the verbs considered in 'Acts Together' and 'Surrogate Acts Together' respectively.

3. Lexical [F26-27]: These bag-of-words style features analyze the connotations of all words (excluding stop-words) occurring between pairs of mentions of the two characters in the sentence. E.g. in S5 in Fig. 1 the words occurring between the pair of mentions the characters, *Tom* and *Becky*, are "goes on a picnic to McDougal's cave with" (stopwords included for readability).

4. Semantic Parse based: These features incorporate information from a framenet-style semantic parse of the sentence. To design these features, we manually compiled lists of positive (or negative) *frames* (and relevant *frame-elements*) depending on whether they are indicative of positive (or negative) relationship between participants (identified in the corresponding frame-elements). We also compiled a list of ambiguous frames like 'cause_bodily_experience' for which the connotation was determined on-the-fly depending on the lexical unit at which that frame fired. Lastly, we had a list of 'Relationship' frames that indicated familial or professional relationships. Table 1 shows examples of various frame-types and relevant frame-elements. The complete lists are available on the first author's webpage. Based on these lists, we extracted the following two types of features:

- *Frames Fired [F28-F30]:* Three numeric features counting number of positive, negative and 'relationship' frames fired such that at least one of the characters belonged to the relevant frame-element.

- *Frames Fired [F31-F33]:* Three features counting number of positive, negative and 'relationship' frames fired.

3.3 Transition features

While content features assist the model in analyzing the text of individual sentences, these features enable it to remember relationship histories, thus discouraging it from changing relationship states too frequently within a sequence.

- $\phi(y_j, y_{j-1}, y_{j-2}) = 1$ if current state is y_j and the previous two states were y_{j-1}, y_{j-2}; 0 otherwise
- $\phi(y_j, y_{j-1}) = 1$ if current state is y_j and the previous state was y_{j-1}; 0 otherwise
- $\phi(y_0) = 1$ if state of the first sentence in the sequence is y_0; 0 otherwise

4 Empirical Evaluation

We now describe our data, baselines and evaluation results.

4.1 Datasets

We have used two datasets in our experiments both of which are based on novel summaries. We considered summaries instead of complete text of the novels because we found summaries to be more precise and informative. Due to the inherent third-person narration style of summaries, they contain more explicit evidences about relationships. On the other hand, while processing novel texts directly one would have to infer these evidences from dialogues and subtle cues. While this is an interesting task in itself, we leave this exploration for future.

SparkNotes: This dataset consists of a collection of summaries ('Plot Overviews') of 300 English novels extracted from the 'Literature Study Guides' section of SparkNotes [2]. We pre-processed these summaries as described in Sec. 3. Thereafter, we considered all pairs of characters that appeared together in at least five sentences in the respective summaries and arranged these sentences in order of appearance in the original summary. We refer to these sequences of sentences as simply a sequence. We considered the threshold of 5 sentences to harvest sequences long enough to manifest 'evolving relationships, but also sufficiently many to allow learning. This yielded a collection of 634 sequences consisting of a total of 5542 sentences.

For our experiments, we obtained annotations for a set of 100 sequences. Annotators were asked to read the complete summary of a novel and then annotate character-pair sequences associated with it. Sentences in a character-pair sequence were labeled as cooperative (when the two characters supported each others actions/intentions or liked each other) or non-cooperative (otherwise). Annotators had access to the complete summary throughout the annotation process. They were required to fully annotate the sequences whenever possible and partially annotate them when they couldn't decide a relationship status for some of the sentences in the sequence. It was permissible to annotate a sequence with all cooperative or all non-cooperative states. In fact, that happened in 70% of the sequences. The dataset thus obtained contained 50 fully annotated sequences (402 sentences) and 50 partially annotated sequences (containing 390 sentences, of which 201 were annotated). Of all annotated sentences, 472 were labeled with a cooperative state. (The dataset is available at the first author's webpage [3].)

Model	P	R	F	ED
J48	67.89	75.60	69.86	0.98
LR	72.33	77.51	72.94	1.06
Order 1 Model	74.32	73.16	73.70	0.9
Order 2 Model	76.58	76.97	**76.76**	**0.66**

Table 2: Cross validation performances on SparkNotes data. The second order model based framework outperforms the one that uses a first order model and the unstructured baselines LR and J48.

AMT: We considered another dataset only for evaluating our model. This dataset (Massey et al. 2015) was collected independently by another set of authors using Amazon Mechanical Turk. The annotators were shown novel-summaries and a list of characters appearing in the novel. Given a pair of characters, they annotated if the relationship between them changed during the novel (binary annotations). They were also asked other questions, such as the overall nature of their relationship etc., which were not relevant for our problem. There was some overlap between the novel summaries used by the two datasets described here, due to which, 62 pairs of characters from this dataset could be found in the SparkNotes dataset. This dataset of 62 pairs can be viewed as providing additional binary ground truth information and was used for evaluation only after training on SparkNotes data. The relationship was annotated as 'changed' (positive class) for 20% of these pairs.

4.2 Baselines and Evaluation Measures

Our primary baseline is an unstructured model that trains flat classifiers using the same content features as used by our framework but treats individual sentences of the sequences independently. We compare our model with this baseline to test our hypothesis that relationship sequence prediction is a structured problem, which benefits from remembering intra-novel history of relationship between characters.

We also compare our framework, which employs a second order Markovian segmentation model, with an identical framework, with a first order Markovian segmentation model. This baseline is included to understand the importance of remembering a longer history of relationship between characters. Also, since a higher order model can look further back, it will discourage frequent changes in relationship status within the sequence more strongly.

For evaluation, we use two different measures comparing model-performances at both sentence and sequence levels. Our first measure accesses the goodness of the binary relationship state assignments for every sentence in the sequence using averaged Precisions (P), Recalls (R) and F1-measures (F) of the two states. The second evaluation measure, mimics

[2] http://www.sparknotes.com/lit/

[3] https://sites.google.com/site/snigdhac/academics

Model	P	R	F
J48	68.18	43.55	48.54
LR	71.93	46.77	51.48
Order 1 Model	72.36	50.64	52.52
Order 2 Model	71.62	56.45	**60.76**

Table 3: Performance comparison on the AMT dataset. The second order model based framework outperforms the one that uses a first order model and the unstructured models LR and J48.

a more practical scenario by evaluating from the perspective of the predicted relationship sequence, $\mathbf{r}$, instead of looking at individual sentences of the sequence. It compares the 'proximity' of the predicted sequence to the ground truth sequence using Edit Distance and reports its mean value (ED) over all test sequences. A better model will be expected to have a smaller value for this measure.

4.3 Evaluation on the SparkNotes dataset

Table 2 compares 10-fold cross validation performances of our second order Semi-supervised Framework (Order 2 Model) with its first order counterpart (Order 1 Model) and two unstructured baselines: Decision Tree (J48) and Logistic Regression (LR). Since the performance of the semi-supervised frameworks depends on random initialization of the weights, we report are mean values over 50 random restarts in the table. The number of relationship states, $|Y|$, was set to be 2 to correspond to the gold standard annotations. The table shows that the framework with the first order Markov model yields slightly better performance (higher averaged F-measure and lower mean Edit Distance) than the unstructured models (LR and J48). This hints at a need for modeling the information flow between sentences of the sequences. The further performance improvement with the second order model emphasizes this hypothesis and also demonstrates the benefit of remembering longer history of characters while making relationship judgments.

4.4 Evaluation on the AMT dataset

Table 3 compares performances of the various models on the AMT dataset using averaged Precision, Recall and F measures on the binary classification task of change prediction. The problem setting, input sequences format and the training procedure for these models is same as above. However, the models produce structured output (relationship sequences) that need to be converted to the binary output of change prediction task. We do this simply by predicting the positive class (change occurred) if the outputted relationship sequence contained at least one change. We can see that while the performance of the framework using the first order model is similar to that of the baseline LR, the second order model shows a considerable improvement in performance. A closer look at the F measures of the two classes (not reported due to space constraints) revealed that while the performance on the positive class was similar for all the models (except J48 which was lower), the performance on the negative class (no change) was much higher for the structured models (56.0 for LR and 57.4 and 67.8 for the First and

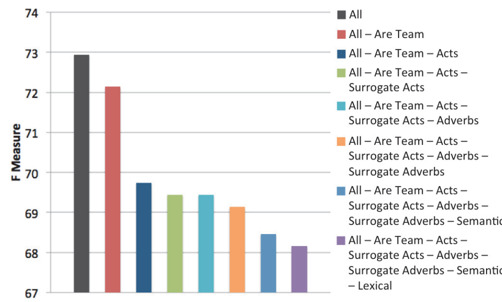

Figure 2: Ablation results on SparkNotes dataset. All represents performace with full feature-set and rest of the bars indicate performance with incrementally removing various feature-families.

Second order models respectively). This might have happened because the unstructured model looks at independent sentences and cannot incorporate historical evidence so it is least conservative in predicting a change, which might have resulted in low recall. The structured models, on the other hand, look at previous states and can better learn to make coherent state predictions.

4.5 Ablation Study

Fig. 2 plots 10-fold cross validation F-measure to study the predictive importance of various feature-families using LR on the SparkNotes data. The black bar (labeled 'All') represents the performance using the complete feature set and the rest of the bars represent the scenario when various features-families are incrementally omitted. We can note that the 'Are Team' and 'Acts Together' features seem to be very informative as removing them degrades the performance remarkably. On the other hand, the 'Adverbs Together' feature seems to be least informative, possibly because it was sparsely populated in our dataset. Nevertheless we can conclude that, in general, removing any feature-family degrades model's performance indicating their predictive utility.

5 Case Study

We now use our framework to gain additional insights into our data. To this end, we use the framework to make predictions about various character pairs from the seven *Harry Potter* novels by *J. K. Rowling*. As before, only those pairs were considered for which the two characters appeared together in at least five sentences and none of these pairs were manually annotated. We then clustered the various pairs according to the Edit-distance based similarity between their relationship sequences. Table 4 shows sample pairs for three such clusters. Note that some of the character pairs appear more than once because several characters are shared across the seven books. While performing this clustering no information other than the relationship sequence itself (such as character identities) was used. Also, the pairs are unordered.

Cluster 1 consists of pairs whose relationship remained mostly cooperative throughout the novel. This includes relationships of Harry with his friends Ron and Hermione,

Cluster 1	Cluster 2	Cluster 3
Harry, Dobby [*Chamber of Secrets*]	Ron, Hermione [*Deathly Hallows*]	Harry, Snape [*Prisoner of Azkaban*]
Harry, Dumbledore [*Half-Blood Prince*]	Harry, Ron [*Deathly Hallows*]	Draco, Harry [*Half-Blood Prince*]
Hagrid, Harry [*Prisoner of Azkaban*]	Sirius, Ron [*Prisoner of Azkaban*]	Voldemort, Dumbledore [*Half-Blood Prince*]
Ron, Harry [*Order of the Phoenix*]	Sirius, Harry [*Prisoner of Azkaban*]	Voldemort, Dumbledore [*Deathly Hallows*]
Harry, Hermione [*Deathly Hallows*]	Hermione, Sirius [*Prisoner of Azkaban*]	Harry, Voldemort [*Half-Blood Prince*]

Table 4: Sample character pairs (and book titles) from the clusters obtained from the *Harry Potter* series. Pairs in clusters 1 and 3 had a cooperative and non-cooperative relationship throughout the novel (respectively). Cluster 2 contains pairs for which the relationship became non-cooperative once in the novel but then finally became cooperative.

benefactors Dumbledore, Hagrid and Dobby. Cluster 2 consists of pairs like ⟨ Ron, Hermione⟩ and ⟨ Harry, Ron⟩ from 'Harry Potter and the Deathly Hallows'. Their relationships were similar in the sense that the three characters started out as friends and go on a quest. However, because of their initial failures Ron abandons the other two, but reunites with them towards the later part of the novel. Hence Ron's relationship with each of the other two was both cooperative and non-cooperative during the course of the novel.

Similarly, Cluster 3 consists of character pairs, which had a non-cooperative relationship for most of the novel. However, a more careful examination revealed that in some of these pairs the model assigned a cooperative status to a few sentences in the beginning. For example, for Voldemort and Dumbledore in the book titled 'Harry Potter and the Half-Blood Price' when Dumbledore (along with Harry) tries to learn more about Voldemort. A human reader, who has access to the complete text of the summary as well as context from previous novels, can understand that they learn about him to fight him better and hence the reader can infer that the relationship is non-cooperative. However, in our setting, we ignore all sentences except those in which the two characters appear together, hence depriving the model of the valuable information present in between the input sentences. We believe that a more sophisticated approach that understands the complete narrative text (instead of sporadic sentences about the two characters of interest) will make better inferences.

6 Literature Survey

Our work is most closely related to the character-centric methods in computational narrative domain. Bamman, O'Connor, and Smith (2013) presented two latent variable models for learning personas in summaries of films by incorporating events that affect the characters. In their subsequent work (Bamman, Underwood, and Smith 2014), they automatically infer character personas in English Novels. Similarly Valls-Vargas, Zhu, and Ontañón (2014) extract character roles from folk tales based on their actions. There have been other attempts towards understanding narratives from view points of characters (Chaturvedi, Goldwasser, and Daume 2015). Unlike our work, these approaches do not model relationships.

Previous work has also focused on constructing social networks from text, though the interpretation of links between people varies considerably. Elson, Dames, and McKeown (2010) constructed social networks of characters of British novels and serials by analyzing their dialogue interactions. Their goals required them to model 'volume' of interactions rather than 'nature' of relationships. Agarwal et al. (2013) focused on social events to construct social network with unstructured text. They also do not model polarity of relationships. However, they emphasized the importance of using dynamic networks. He, Barbosa, and Kondrak (2013) presented a method to infer speaker identities and use them to construct a social network showing familial or social relationships. Most of these approaches used social networks to identify positive relationships between people. Leskovec, Huttenlocher, and Kleinberg (2010) proposed signed social networks to model both positive and negative relationships, though they operate in social media domain. More recently, Krishnan and Eisenstein (2015) analyze movie scripts to construct a signed social network depicting formality of relationships between characters. Srivastava, Chaturvedi, and Mitchell (2015) construct signed but static social networks from movie summaries. Apart from domain of application, our work differs from these because we model 'polarity' of relationships and do that in a dynamic fashion.

7 Conclusion and Discussion

In this paper we model dynamic relationships between pairs of characters in a narrative. We analyze summaries of novels to extract relationship trajectories that describe how the relationship evolved. Our semi-supervised framework uses a structured segmentation model that makes second-order Markov assumption to remember the 'history' of characters and analyzes textual contents of summaries using rich semantic features that incorporate world knowledge. We demonstrate the utility of our model by comparing it with an unstructured model that treats individual sentences independently and also with a lower order model that remembers shorter history. In future we would like to experiment with higher order and semi-Markov models.

Also, this work treats different character pairs from the same novel independently and does not attempt to understand the complete text of the narrative. In future, we would like to explore a more sophisticated model that exploits intra-novel dynamics while predicting relationships.

An important contribution this work is identifying the evolutionary nature of relationships. Further work in this direction could be used to answer questions like "What kind of novels have happy endings?", "Are there general narrative templates of relationship evolution between the protagonist and his/her lover?" etc. Acknowledging the dynamic nature of relationships can also find application in social media domain. For instance, social networking sites could use this phenomenon for customizing news feeds. A change in nature/strength of user's relationship can lead to change in interest in 'news' related to her 'friends'.

References

Agarwal, A.; Kotalwar, A.; Zheng, J.; and Rambow, O. 2013. SINNET: social interaction network extractor from text. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan*, 33–36.

Agarwal, A.; Balasubramanian, S.; Kotalwar, A.; Zheng, J.; and Rambow, O. 2014. Frame semantic tree kernels for social network extraction from text. *EACL 2014* 211.

Bamman, D.; O'Connor, B.; and Smith, N. A. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Sofia, Bulgaria, Volume 1: Long Papers*, 352–361.

Bamman, D.; Underwood, T.; and Smith, N. A. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Baltimore, MD, USA, Volume 1: Long Papers*, 370–379.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus, Ohio, USA*, 789–797.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics, Singapore*, 602–610.

Chambers, N. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, Seattle, Washington, USA*, 1797–1807.

Chaturvedi, S.; Goldwasser, D.; and Daume, III, H. 2015. Ask, and shall you receive?: Understanding Desire Fulfillment in Natural Language Text. *ArXiv e-prints*.

Cheung, J. C. K.; Poon, H.; and Vanderwende, L. 2013. Probabilistic frame induction. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Atlanta, Georgia, USA*, 837–846.

Collins, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, 1–8.

Das, D.; Chen, D.; Martins, A. F. T.; Schneider, N.; and Smith, N. A. 2014. Frame-semantic parsing. *Computational Linguistics* 40(1):9–56.

Elsner, M. 2012. Character-based kernels for novelistic plot structure. In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France*, 634–644.

Elson, D. K.; Dames, N.; and McKeown, K. 2010. Extracting social networks from literary fiction. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden*, 138–147.

Elson, D. K. 2012. *Modeling Narrative Discourse*. Ph.D. Dissertation, Columbia University.

Feng, S.; Kang, J. S.; Kuznetsova, P.; and Choi, Y. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Sofia, Bulgaria, Volume 1: Long Papers*, 1774–1784.

Finlayson, M. A. 2012. *Learning Narrative Structure from Annotated Folktales*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Goyal, A.; Riloff, E.; and Daumé III, H. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, Massachusetts, USA*, 77–86.

He, H.; Barbosa, D.; and Kondrak, G. 2013. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Sofia, Bulgaria, Volume 1: Long Papers*, 1312–1320.

Krishnan, V., and Eisenstein, J. 2015. "You're Mr. Lebowski, I'm the Dude": Inducing address term formality in signed social networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1616–1626.

Leskovec, J.; Huttenlocher, D. P.; and Kleinberg, J. M. 2010. Signed networks in social media. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA*, 1361–1370.

Liu, B.; Hu, M.; and Cheng, J. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan*, 342–351.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Baltimore, MD, USA*, 55–60.

Massey, P.; Xia, P.; Bamman, D.; and Smith, N. 2015. Annotating character relationships in literary texts. *ArXiv e-prints*.

McIntyre, N., and Lapata, M. 2010. Plot induction and evolutionary search for story generation. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden*, 1562–1572.

Orr, J. W.; Tadepalli, P.; Doppa, J. R.; Fern, X.; and Dietterich, T. G. 2014. Learning scripts as hidden markov models. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada.*, 1565–1571.

Propp, V. I. 1968. *Morphology of the folktale*. University of Texas.

Regneri, M.; Koller, A.; and Pinkal, M. 2010. Learning script knowledge with web experiments. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Sweden*, 979–988.

Schank, R. C., and Abelson, R. P. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures (Artificial Intelligence Series)*. Psychology Press, 1 edition.

Srivastava, S., and Hovy, E. 2014. Vector space semantics with frequency-driven motifs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 634–643. Baltimore, Maryland: Association for Computational Linguistics.

Srivastava, S.; Chaturvedi, S.; and Mitchell, T. 2015. Inferring interpersonal relations in narrative summaries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona*.

Valls-Vargas, J.; Zhu, J.; and Ontañón, S. 2014. Toward automatic role identification in unannotated folk tales. In *Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2014, Raleigh, NC, USA*.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT 2005, Human Language Technology Conference, Vancouver, British Columbia, Canada*.