Unsupervised Learning of Evolving Relationships Between Literary Characters

Snigdha Chaturvedi
Department of Computer Science
University of Illinois
Urbana-Champaign
snigdha@illinois.edu

Mohit Iyyer
Department of Computer Science
University of Maryland
College Park
miyyer@umiacs.umd.edu

Hal Daumé III
Department of Computer Science
University of Maryland
College Park
hal@umiacs.umd.edu

Abstract

Understanding inter-character relationships is fundamental for understanding character intentions and goals in a narrative. This paper addresses unsupervised modeling of relationships between characters. We model relationships as dynamic phenomenon, represented as evolving sequences of latent states empirically learned from data. Unlike most previous work our approach is completely unsupervised. This enables data-driven inference of inter-character relationship types beyond simple sentiment polarities, by incorporating lexical and semantic representations, and leveraging large quantities of raw text. We present three models based on rich sets of linguistic features that capture various cues about relationships. We compare these models with existing techniques and also demonstrate that relationship categories learned by our model are semantically coherent.

1 Introduction

Understanding characters in a narrative is essential for Natural Language Understanding. To this end, the field of computational narratives studies narratives or stories from the perspective of characters mentioned in them (Elsner 2012).

Recent attempts at character-centric story understanding model inter-character relationships (Krishnan and Eisenstein 2015; Chaturvedi 2016). Understanding relationships between characters assists in interpreting and justifying their actions in a narrative. It is also a step towards human-like natural language understanding by modeling capabilities of 'filling-in-the-gaps' about what is not explicitly stated in the text, and building expectations of future events in a narrative. Modeling relationships in unstructured texts can also be used to analyze large collections of texts in journalism, political science, digital humanities, etc. (Elson, Dames, and McKeown 2010; Agarwal, Kotalwar, and Rambow 2013). However, most existing methods for analyzing relationships are inadequate in three ways: (i) They characterize relationships by coarse sentiment polarities, e.g., friendly vs. adversarial, which conflates distinct semantic categories; (ii) They require expensive and resource-intensive manual annotation; (iii) They assume a static relationship between characters within a narrative, represented by a single variable.

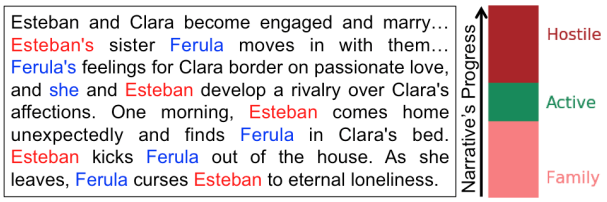


Figure 1: Left: Excerpts from summary of 'The House of the Spirits'. Right: Relationship sequence learned by our model for Esteban and Ferula. Each colored block represents their relationship for a short while, with its height proportional to the duration for which they were in that relationship (relationship definitions in Table 1).

Shortcomings of such methods are apparent. Fig. 1 shows excerpts from a novel-summary focusing on Esteban and Ferula. Their relationship transitions from familial to that of hostility because of Ferula's developing intimacy with Esteban's wife. Clearly, their relationship is complex and cannot be expressed as simply friendly vs. adversarial. Real-world relationships are nuanced with facets such as family, romance, formality/informality, etc. and an ideal model would be able to express these. Also, describing their relationship by a single category will not explain all of their actions. For instance, if we assume that they are rivals then we cannot explain why Ferula initially moves in with Esteban. Thus, there is a need to model relationships as dynamic variables that can express these real-world aspects of human relationships and evolve with the progress of the narrative.

In this paper we present a framework for **unsupervised modeling of inter-character relationships** from unstructured text. The input to our models is a narrative summary and a pair of characters appearing in the narrative. We model relationships by examining the sequence of sentences that mention both characters, arranged in order of their appearance in the narrative. Each sentence is represented by a feature-vector, which captures cues about their relationship in the sentence. Finally, each vector is associated with a latent state, representing the relationship between the two characters at that point in the narrative. The sequence of latent states hence represents their (dynamic) relationship over the course of the narrative. Treating this as an unsupervised

structured prediction task, we learn the latent states by making Markovian assumptions to capture the 'flow of information' between individual sentences.

2 Related Work

Srivastava, Chaturvedi, and Mitchell (2016) model inter-character relationships from narratives. However, they do not incorporate the dynamic nature of these relationships. Also, they make a limiting assumption that relationships are only of two types *cooperative* or *non-cooperative*. Chaturvedi et al. (2016) model the evolution of interpersonal relationships in novels in a supervised setting, requiring manually annotated data, and like the previous work, model relationships as binary polarities.

Our setting is most closely related to that of Iyyer et al. (2016) who also model evolving inter-character relationships. Given a narrative text and a character-pair, their proposed approach, RMN, learns a sequence of discrete states depicting the relationship between the two characters. RMN, which is an variation of a deep recurrent autoencoder, is trained on the raw text of a large collection of novels. Despite the differences in problem setting (plot summaries instead of raw text, and fewer relationship types), we compare our model's performance to that of the RMN on our dataset (described later).

Apart from this, previous works construct social networks of characters depicting limited types of relationships such as formality (He, Barbosa, and Kondrak 2013; Krishnan and Eisenstein 2015), volume of interaction (Elson, Dames, and McKeown 2010), participation in social events (Agarwal, Kotalwar, and Rambow 2013; Agarwal et al. 2014; 2013). These methods are different from ours because they do not necessarily model varied aspects of inter-character relationships, and if they do, they do not model their dynamic nature.

Other character-centric methods have focused on modeling character *personas* (Bamman, O'Connor, and Smith 2013; Bamman, Underwood, and Smith 2014). While this approach might be useful in understanding simple folktales (Valls-Vargas, Zhu, and Ontañón 2014; 2015), a clear mapping from characters to roles may not be feasible in complex or real-world narratives. On the other hand, approaches towards interpersonal relationship modeling circumvent this mapping process, while still assisting comprehension of character behavior.

Previous methods have also analyzed narratives from the perspective of events occurring in them and include scripts (Schank and Abelson 1977; Regneri, Koller, and Pinkal 2010; Orr et al. 2014; Pichotta and Mooney 2016), plot units (Lehnert 1981; Goyal, Riloff, and Daumé III 2010; Finlayson 2012), temporal event chains (Chambers and Jurafsky 2008; 2009; Chambers 2013), bags of related events (Cheung, Poon, and Vanderwende 2013), etc.

3 Feature-vectors Extraction

Given a narrative text and two characters appearing in it, our goal is to represent their relationship as a sequence of latent variables. We consider sentences in which the two characters appear together. These sentences have a natural order of appearance in the narrative, yielding a sequence, $\mathbf{s} = \langle s_1, s_2 \dots s_T \rangle$. We represent each sentence with a D-dimensional feature-vector, $\vec{f}_t \in \mathbb{R}^D$, producing the sequence: $\mathbf{f} = \langle \vec{f}_1, \vec{f}_2 \dots \vec{f}_T \rangle$. We provide this sequence of feature-vectors as input to our models. The models assign each feature-vector, $\vec{f}_t$, to a discrete latent relationship state, $r_t \in \{1, 2, \dots R\}$, thus outputting a sequence of latent relationship states, $\mathbf{r} = \langle r_1, r_2 \dots r_T \rangle$. In this section, we describe the process of obtaining the sequence of feature-vectors from textual sentences.

Sentence: After confronting Maria, Jim furiously asked her to end her friendship.
Surrogate Actions: confronting **Lexical:** furiously asked
Actions: asked **Semantic:** friendship, 'personal relationships'

Figure 2: Features for Jim and Maria's relationship.

We preprocessed the narratives using the BookNLP pipeline (Bamman, Underwood, and Smith 2014) to obtain POS tags, dependency parses, and co-referent mentions, and to identify major characters. We also obtained the frame-semantic parses of sentences (Das et al. 2014). Finally, we extract the following sets of words from each sentence (our feature-vector is an averaged embedding of these words):

Actions: Following Propps Structuralist narrative theory (Propp 1968), we represent inter-character relationship using their actions, especially those done to each other. For this, we identify verbs, and their agents (using 'nsubj' and 'agent' dependency relations) and patients (using 'dobj' and 'nsubjpass' relations). We also consider verbs conjoined with each other with a 'conj' relation. Finally, we extract the set of verbs, which have one character as an agent, and the other as a patient. For example in the sample sentence in Fig. 2, the action word is 'asked'.

Surrogate Actions: The high-precision action words can get affected by NLP pipeline's limitation. E.g. in Fig. 2, 'Jim' is the implicit agent of 'confronting' 'Maria'. To include such cases, we extract another set of verbs, which have either of the two characters as the agent or patient, provided the sentence did not contain a mention of another character.

Lexical: This is a bag-of-words set of all words (except stopwords) that appear between pair of mentions of the two characters (see Fig. 2 for an example).

Frame-semantic: This set makes use of the frame-sematic parse of the sentence and the frame-polarity lexicon (Chaturvedi et al. 2016) which contains a list of frames indicative of relationships. This set includes all frames (and the tokens at which they were evoked), appearing in the above mentioned lexicon, evoked for at least one of the characters as a *frame-element*. E.g. in Fig. 2, a 'personal relationships' frame is evoked at the token 'friendship' for Maria.

After extracting these sets of words from individual sentences, we obtain a feature-vector representation, $\vec{f}_t \in \mathbb{R}^D$, for each sentence, s_t, by averaging the vector-space embeddings of the individual words in the union of these sets (motivated by the additive model of vector compositional-

ity (Mitchell and Lapata 2008)). Sec. 5.1 contains more details of the word embeddings.

4 Learning Relationship Sequences

Given the feature vectors as input, $\mathbf{f} = \langle \vec{f_1}, \vec{f_2} \dots \vec{f_T} \rangle$, we now describe models that learn the relationship sequence, $\mathbf{r} = \langle r_1, r_2 \dots r_T \rangle$. The first model is *GHMM*, a non-Bayesian Hidden Markov Model with Gaussian Emissions. The hidden states comprise of relationship states and vector representation of sentences form the observations. We then describe *Penalized HMM*, which extends GHMM by *smoothing* the relationship sequences and discouraging frequent changes in relationship states within a sequence. Finally, our last approach, *Globally Aware GHMM*, attempts to simulate the intuition of a *global belief* about the relationship between the characters, while analyzing the individual sentences of the sequence.

4.1 GHMM

Our first approach is a Hidden Markov Model with Gaussian Emissions, which generates the feature-vector sequence as:

For every vector, $\vec{f_t} \forall t \in \{1, 2, 3 \dots T\}$:
1. If $t = 1$, choose $r_1 \sim Multinomial(\boldsymbol{\pi})$
2. If $t > 1$, choose $r_t \sim Categorical(\boldsymbol{\phi}_{\boldsymbol{r_{t-1}}})$
3. Emit vector $\vec{f_t} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{r_t}}, \boldsymbol{\Sigma}_{\boldsymbol{r_t}})$

where, $\boldsymbol{\pi}$ is an R-dimensional probability distribution indicating start state probabilities. Also, $\boldsymbol{\phi}_{\boldsymbol{r_{t-1}}}$ represents the transition probabilities, i.e. ϕ_{ij} is the probability of transitioning from state i to state j, and $\sum_{j=1}^{R} \phi_{ij} = 1$. Finally, it is assumed that vectors belonging to a state, r, are normally distributed with mean, $\boldsymbol{\mu}_{\boldsymbol{r}}$, and covariance, $\boldsymbol{\Sigma}_{\boldsymbol{r}}$.

This model thus defines the joint distribution over a sequence of feature-vectors as:

$$p(\mathbf{f}, \mathbf{r}) = \prod_{t=1}^{T} p(r_t | r_{t-1}) p(\vec{f_t} | r_t) \tag{1}$$

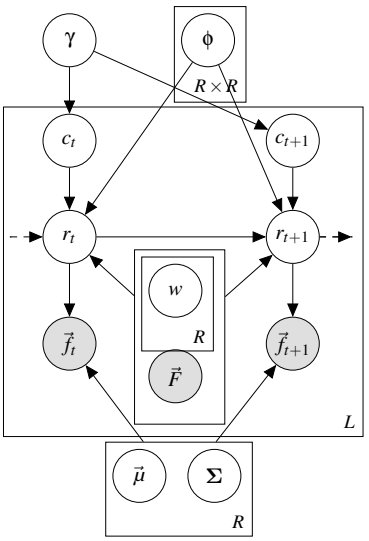
where, $p(r_t | r_{t-1})$ is obtained from $\phi_{r_{t-1} r_t}$, and $p(\vec{f_t} | r_t) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{r_t}}, \boldsymbol{\Sigma}_{\boldsymbol{r_t}})$. We use Baum-Welch algorithm to fit the various parameters of this model.

4.2 Penalized GHMM

In practice GHMM resulted in highly fluctuating relationship sequences. While this might be a good feature for traditional sequence modeling tasks like POS tagging, real-world relationship sequences tend to remain consistent over long parts of a narrative. We, therefore, propose a more domain-specific model, Penalized GHMM, which is similar to GHMM, except that in the generative process, every time the model makes a transition from state i to state j, it incurs a penalty, ρ_{ij}, which takes the value of 1 whenever $i = j$ and ϵ otherwise. This model defines the joint distribution over a sequence of feature-vectors as:

$$p(\mathbf{f}, \mathbf{r}) = \prod_{t=1}^{T} p(r_t | r_{t-1}) \rho_{r_{t-1}, r_t} p(\vec{f_t} | r_t) \tag{2}$$

The parameter estimation process for this model is similar to that of GHMM.



$\vec{F}$ = Global feature-vector of the sequence
$\vec{f_t}$ = Feature-vector for individual sentences
r_t = relationship states for sentences
c_t = global versus local choice
γ = model's preference towards the local component
$\vec{w}$ = weights of the global component
ϕ = transition probabilities for the local component
$\vec{\mu}$, Σ = parameters of Normal distribution
L = number of sequences in the dataset
R = number of relationship states

Figure 3: Diagram for the Globally Aware GHMM

4.3 Globally Aware GHMM

The above models are *local* in nature, as at any point in the sequence, t, the relationship state, r_t, depends only on the previous state, r_{t-1}, and the emitted features, $\vec{f_t}$. However, it may be argued that judging the relationship at any juncture needs consideration of not only the current sentence, but also a *global perspective* of the *overall* nature of the relationship between the characters. E.g., in 'Harry Potter and the Half Blood Price', Harry, the protagonist, 'learns more' about the villain, Voldemort. While 'learning more' does not tell us much about their relationship, knowing that they are enemies in general, indicates that the relationship is that of animosity and Harry is learning more about Voldemort to fight him better.

To incorporate this behavior, we propose another model, Globally Aware GHMM. This model makes a decision about the current relationship state, r_t, after weighing in information from a local component and a global component using a *choice* variable, $c_t \in \{0, 1\}$. The local component uses the Penalized GHMM style transitions to determine the current relationship state. Whereas, the global component (represented by θ) uses a logistic regression model based on a global feature-set, $\vec{F}$, extracted from the whole sequence, $\mathbf{s} = \langle s_1, s_2 \dots s_T \rangle$. This model defines the joint distribution over a sequence as:

$$p(\mathbf{f}, \mathbf{r}) = \prod_{t=1}^{T} [\gamma \cdot p(r_t | r_{t-1}) \cdot \rho_{r_{t-1}, r_t} + (1 - \gamma) \cdot \theta(r_t | \vec{F})] \cdot p(\vec{f_t} | r_t) \tag{3}$$

Here γ is the model's preference towards the local model ($\gamma = p(c = 1)$) and the global component, θ, is modeled as:

$$\theta(r | \vec{F}) = \frac{\exp(\vec{w_r} \cdot \vec{F})}{\sum_{r'=1}^{R} \exp(\vec{w_{r'}} \cdot \vec{F})} \tag{4}$$

where, $\vec{w_r}$ are the weights corresponding to the relationship state r that are learned during training.

Figure 3 pictorially describes our model and its generative story can be described as follows:

For every vector, $\vec{f_t} \forall t \in \{1, 2, 3 \dots T\}$:
1. Toss a *choice* variable, $c_t \sim Bernoulli(\gamma)$.
2. If $c_t = 0$, choose $r_t \sim \theta(r|\vec{F})$
3. If $c_t = 1$ & $t = 1$, then $r_1 \sim Categorical(\boldsymbol{\pi})$
4. If $c_t = 1$ & $t > 1$, then $r_t \sim Categorical(\boldsymbol{\phi_{r_{t-1}}}) \cdot \rho_{r_{t-1} r_t}$
5. Emit vector $\vec{f_t} \sim \mathcal{N}(\boldsymbol{\mu_{r_t}}, \boldsymbol{\Sigma_{r_t}})$

Training: The model parameters, $\lambda = (\boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{w}, \boldsymbol{\gamma})$ are learned using EM. In each EM iteration, let λ and λ' represent the current and candidate models respectively. We want $p_{\lambda'}(\mathbf{f}) > p_{\lambda}(\mathbf{f})$. It can be shown that this is equivalent to maximizing the following:

$$Q(\lambda, \lambda') = \sum_l \sum_r \sum_c p_\lambda(\mathbf{r^l}, \mathbf{c^l} | \mathbf{f^l}) \log p_{\lambda'}(\mathbf{r^l}, \mathbf{c^l}, \mathbf{f^l})$$

where, l is the index over various sequences in the dataset (L in number) and x^l represents the variable, x, for the l^{th} sequence. In the above equation, $p(\mathbf{r}, \mathbf{c}, \mathbf{f})$ for a sequence is modeled as:

$$p(\mathbf{c}, \mathbf{f}, \mathbf{r}) = \prod_{t=1}^{T} [\{\gamma \cdot p(r_t|r_{t-1}) \cdot \rho_{r_{t-1}, r_t}\}^{\delta_1(c_t)} + \{(1-\gamma) \cdot \theta(r_t|\vec{F})\}^{\delta_0(c_t)}] \cdot p(\vec{f_t}|r_t) \quad (5)$$

where, $\delta_a(x)$ is the Kronecker delta function which takes the value of 1 whenever $x = a$ and 0 otherwise. In the E-step, we use scaled forward-backward algorithm to compute forward and backward probabilities for a sequence. In the M-step we update all the parameters. In this step, we also learn weights, $\vec{w_r}$, of the global component (Eqn. 4) by maximizing the following objective function using a subspace trust-region method based on the interior-reflective Newton method (Coleman and Li 1994; 1996):

$$\sum_l^L \sum_r^R \frac{\exp(\vec{w_r} \cdot \vec{F^l})}{\sum_{r'=1}^R \exp(\vec{w_{r'}} \cdot \vec{F^l})} \sum_{t=2}^{T^l} \beta_r^l(t) \cdot \alpha_{r0}^l(t) \quad (6)$$

5 Empirical Evaluation

Evaluating these models is difficult for several reasons. Not only is manually designing a taxonomy of relationship types challenging, judging the quality of a learned relationship sequence is also subjective. Therefore, we first use a manually annotated dataset (assuming binary relationship types) to compare the performance of the various models (Sec.5.2). We then evaluate how our model's performance compares with human judgment in characterizing relationships (Sec.5.3). We also evaluate if the learned relationship categories are semantically coherent (Sec.5.4). Lastly, we compare our model with a previously proposed approach (Iyyer et al. 2016) (Sec. 5.5) [1].

[1] Supplementary material is available on the first author's webpage.

5.1 Dataset and Implementation Details

We use a dataset of 300 English novel-summaries [2], released by Chaturvedi et al. (2016). We identified major characters in these summaries, and pairs of characters that appeared together in more that 5 sentences were considered for analysis. This threshold was used to obtain character-pairs that interacted long enough to demonstrate the dynamic nature of their relationships but also resulted in a sizeable dataset. The final dataset contained 634 such sequences, with an average length of 8.2 sentences per sequence. The vocabulary size of the input sentences was 10K, and that of the feature-sets extracted from them was 4.2K.

To obtain word-embeddings (Sec. 3), we used the skip-gram model (Mikolov et al. 2013) trained with $D = 200$ on a collection of novels [3] from Project Gutenberg [4].

Globally Aware GHMM uses the average of feature-vectors of all sentences in a sequence as its global feature vector (i.e. $\vec{F} = mean(\vec{f_1}, \vec{f_2} \dots \vec{f_T})$). We used $\epsilon = 0.8$ (selected using cross-validation). Estimating the covariance matrix $\boldsymbol{\Sigma}$ degraded performance, which might be due to overfitting (Shinozaki and Kawahara 2007). Hence, we only show results for estimating $\vec{\mu_r}$, and we use a fixed diagonal matrix as $\boldsymbol{\Sigma}$ (with each diagonal entry being 0.01), following previous approaches (Lin et al. 2015).

5.2 Supervised Evaluation

We begin with indirectly evaluating the models on a supervised task by heuristically aligning learnt latent states against label categories. For this purpose, we use the manually annotated sequences of the data provided by Chaturvedi et al. (2016). It consists of about 50 sequences in which each sentence is labeled with a binary relationship state, *cooperative* or *non-cooperative*, which we refer to as the gold-classes. Relationship states changed in around 30% of the sequences. However, our unsupervised models assign each sentence to a relationship state/cluster but do not provide a label to the states. For this evaluation we heuristically assign each of the learned states, j, a cooperative/non-cooperative label as $\arg\max_{i \in \{coop, non-coop\}} \{\frac{m_i^j}{N_i}\}$. Where, m_i^j is the number of sentences belonging to the learned state j with gold-class i, and N_i is the total number of sentences in the gold-class i. We did not simply label each state with the most-frequent gold-class because of the class skew ($= 0.78$) in the annotated data. Like Chaturvedi et al. (2016), we report averaged F-measures of the two gold-classes. They obtained an F-measure of 76.76. It should be noted that a direct comparison is to them is unfair because our method solves a related but different problem and loses valuable information in mapping states to binary labels. Also, their supervised task requires manual annotations and so has access to a clearer definition of label-space.

Fig. 4 compares the performance of various models for different values of the user-provided input R – the number of relationship states. Since training depends on initializations,

[2] SparkNotes: `http://www.sparknotes.com/`

[3] pre-processed to remove punctuation and capitalization

[4] `https://www.gutenberg.org/`

State	Most Frequent words	MP
Familial	kinship, relationship, personal, father, son, mother, family, marry, wife, brother, friend, love, daughter	1.00
Desire	love, want, realize, fear, declare, desiring, hope, family, kinship, vow, believe, confess, desire, feel, depend	0.33
Active	meet, go, come, take, leave, find, together, tell, return, kill, attack, get, try, run, protect, back, kinship	1.00
Communicative	tell, ask, say, want, leave, find, go, see, know, come, decide, tells, desiring, make, learn, marry, meet, try	0.83
Hostile	kill, killing, die, try, cause harm, destroy, revenge, stab, hurt, decide, fight, leave, murder, kinship, killed	1.00

Table 1: Representative words for relationship states learned by the Globally Aware GHMM, and their Model Precisions (MP)

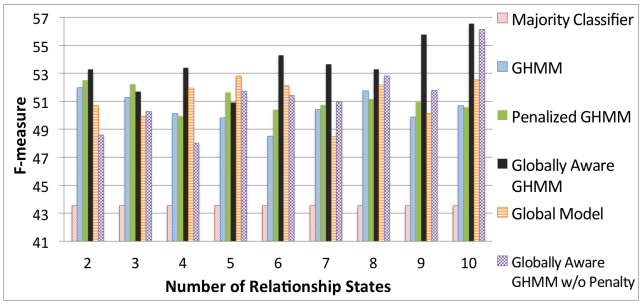


Figure 4: Performance comparison of various models.

Figure 5: Global preference learned by our model with and without penalty. Without penalty, the model has weaker 'global' preference, resulting in lower accuracy due to frequent shifts in relationship states within a sequence.

we report average values for 50 runs for each model. We can see that, all the models significantly outperform the baseline, which always predicts the majority (cooperative) class. The figure shows that Globally Aware GHMM, in general, performs better than the Penalized GHMM and the baseline GHMM. It also outperforms the Global Model, which is an unstructured baseline that clusters the sentences independently (corresponds to the global only component of the Globally Aware GHMM). This indicates that for this task, it is important to have a global as well as local perspective of characters' interactions.

The performance of Penalized GHMM is comparable to that of GHMM, which hints that the penalty term might not be contributing significantly to the model's performance. To investigate this further, we introduce 'Globally Aware GHMM w/o penalty'– a modified Globally Aware GHMM without the penalty term. We can see that its performance is much worse than that of Globally Aware GHMM. This suggests that while the penalty term is not very useful for a local model like GHMM, it is indeed valuable for models like Globally Aware GHMM, which have an ability to switch between local and global components. This is because frequent switching between the two components can result in frequent shifting between relationship states within a sequence. This frequent shifting is unnatural for the given task because the states represent inter-personal relationships, which are usually stable and evolve smoothly with the narrative. Therefore, such a model benefits from a penalty term, which smoothens the relationship trajectories and makes them more consistent with human judgment. This can also be observed in Fig. 5. The Globally Aware GHMM has much stronger preference for one of the components (global) than Globally Aware GHMM w/o penalty. The preference learned for the latter model is closer to 0.5 which would result in frequent shifts between the two components.

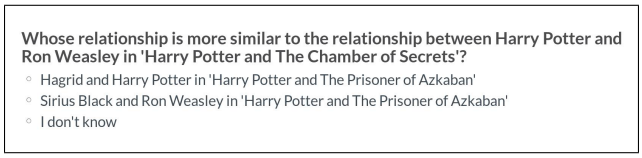


Figure 6: Screenshot from the relationship analogy task

5.3 Relation Analogy Task

We now evaluate our model against human judgment without any restrictive assumptions on the types of relationships. We evaluate the model on an objective task involving human subjects to answer questions based on semantics of inter-character relationships. Specifically, given a pair (P) of characters in a text, we asked subjects to pick a pair that reflects a similar relationship from two choices of pairs of characters, O_1 and O_2. Subjects were asked to consider not just the nature but also the trajectories of relationships while judging similarity. Fig. 6 shows an example question. The pairs were chosen from different books of the 'Harry Potter' series. This made the task easier for humans but not for the model (since it can't access character names). The two options were selected from pool of pairs similar (or dissimilar) to the given pair, P, as determined by our model. We use an edit-distance based measure to compute similarity between (normalized) relationships sequences of any two given pairs.

Subjects, who self-reported that they had previously read the books, were required to read their summaries before answering questions, and had access to these during the task. An illustrative sample question and answer was initially provided to explain the task. For each question, the subjects could choose one of the two candidates character-pairs, or a

don't know option. To reduce annotator fatigue, each annotator was asked questions about characters mentioned in only three books and could not answer more than 10 questions per session. Overall, we collected answers for about 100 such questions, each answered by at least 3 annotators. The raw inter-annotator agreement was 0.73 (Fleiss' $\kappa = 0.46$).

Treating the human provided answers (except the 4.8% 'don't know' options) as gold standard, our model's accuracy on this task was 66.0% (accuracy of a random baseline=50.0%). Considering that this is a difficult task even for humans, we can conclude that the model learns sequences that correlate significantly with human judgment ($p < 10^{-3}$), and can be used to address semantically complicated questions.

5.4 Coherence of Relationship States

Table 1 presents a visualization of the relationship states learned by the Globally Aware GHMM (with $R = 5$). For each state, we report the most frequent words from the union of the feature-sets extracted for all the sentences assigned to that state. We can see that the first state corresponds to familial relationships. The second state corresponds to a desire to initiate romantic relationships (indicated by words like *love*, *desiring*, *vow*, *confess*, etc.). The third state consists of sentences in which the characters participate in physical action like *go*, *meet*, etc. The last two states represent communicative and hostile relationship respectively.

Word-intrusion Detection Task: This task further investigates the semantic coherence of the states. In this task (Chang et al. 2009), a human subject is presented with 6 randomly ordered words. 5 of them are high frequency words from one of our learned relationship states, and one, the 'intruder', belongs to a different (randomly chosen) state. Humans subjects are then asked to identify the intruder word. The subjects were graduate students from varying disciplines and were comfortable with English. Each subject was shown at-least 5 sets of words, and no subject was shown more than 10 sets. We collected judgments for at least 8 sets per state and used them to calculate the 'Model Precision (MP)' for each state, which is the fraction of times a subject accurately identified the intruder. The last column of Table 1 shows the results of this experiment. We can see that the subjects successfully identified the intruder with high precision in all cases ($p < 10^{-3}$) except the 'desire' state ($p \sim 0.3$), indicating their semantic coherence.

5.5 Comparison to RMN

We now compare our model with another unsupervised approach, RMN (Iyyer et al. 2016) (described in Sec.2).

Which model produces better relationship sequences? We first compare the relationship sequences learned by our Globally Aware GHMM with those learned by RMN on our data. For this experiment, human subjects (on CrowdFlower) were presented with a novel summary and a pair of characters appearing in it. They were also shown the outputs of the two models for the given character-pair (represented using the visualization shown in Fig. 1). The subjects then chose which model's output best represented the characters' relationship (binary judgment). Both models were run on our dataset with $R = 5$, and the states learned by them were manually named by the first authors of the respective papers. However, an attempt was made to assign same names for states that looked very similar for the two models.

In order to avoid subject fatigue, we filtered out summaries with more than 1000 tokens. We also required the subjects to demonstrate proficiency on a set of 5 test-questions. We collected judgments on 133 character-pairs, each of which was annotated by at least 3 subjects. The subjects chose our model over RMN for 62.2% of the character-pairs ($p < 10^{-4}$ and inter-annotator agreement = 0.65).

Do the states represent relationships? The above experiment compares the relationship sequences learned by Globally Aware GHMM with those learned by RMN. We now evaluate if the individual states learned by the two models are indeed representing an aspect of inter-personal relationship. For this experiment, we ran the two models on our dataset using varying values of $R \in \{2, 3...10\}$, resulting in a total of 54 states for each model. We then presented human-subjects with a word-cloud based visualization of individual states (generated using Wordle(`wordle.net`)) represented by their 20 most frequent words. We then asked them to judge if the words in the state represent a human relationship (one binary judgment per state). To keep the evaluation fair, the subjects were unaware of the underlying models or their goals. Each state was judged by 3 subjects, resulting in 324 judgments. We employed 9 graduate students as subjects who self-reported that they were proficient in English.

The subjects judged 66.0% of the states learned by Globally Aware GHMM to be representing an inter-personal relationship, while only 50.0% of RMN's states were judged to be representing relationships (inter-annotator agreement of 0.72). Also, out of those states judged as representing relationships, there was a unanimous agreement between all the subjects for 62.7% of Globally Aware GHMM's states but only 33.3% of RMN's states. This indicates that Globally Aware GHMM learns states that are more representative of inter-personal relationships as compared to RMN.

6 Conclusion

This paper addressed the problem of learning evolving inter-character relationships from novel summaries. Treating this as an unsupervised structured prediction problem we present three models that incorporate linguistic as well as contextual information. We empirically demonstrate that for solving this problem, it is not sufficient to simply look at local cues about the relationships between the two characters of interest from a small part of text. Instead, it is important to maintain a global perspective of the *overall* nature of relationships. Future work could focus on making the relationship states non-overlapping and more diverse. Other directions may study usefulness of varying text modes (genre, number of characters, time-period of novels, etc.); or mining 'relationship patterns' from such texts.

References

Agarwal, A.; Kotalwar, A.; Zheng, J.; and Rambow, O. 2013. Sinnet: Social interaction network extractor from text. In *The Compan-*

ion Volume of the Proceedings of IJCNLP 2013: System Demonstrations, 33–36.

Agarwal, A.; Balasubramanian, S.; Kotalwar, A.; Zheng, J.; and Rambow, O. 2014. Frame semantic tree kernels for social network extraction from text. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 211–219.

Agarwal, A.; Kotalwar, A.; and Rambow, O. 2013. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 1202–1208.

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*.

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*, 370–379.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08*, 789–797.

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

Chang, J.; Boyd-Graber, J.; Wang, C.; Gerrish, S.; and Blei, D. M. 2009. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*.

Chaturvedi, S.; Srivastava, S.; Daumé III, H.; and Dyer, C. 2016. Modeling evolving relationships between characters in literary novels. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2704–2710.

Chaturvedi, S. 2016. *Structured Approaches for Exploring Interpersonal Relationships in Natural Language Text*. Ph.D. Dissertation, University of Maryland, College Park.

Cheung, J. C. K.; Poon, H.; and Vanderwende, L. 2013. Probabilistic frame induction. In *In Proceedings of the North American Chapter of the Association of Computational Linguistics*.

Coleman, T. F., and Li, Y. 1994. On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds. *Mathematical Programming* 67(2):189–224.

Coleman, T. F., and Li, Y. 1996. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* 6(2):418–445.

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 *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

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

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*, 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*, 1312–1320.

Iyyer, M.; Guha, A.; Chaturvedi, S.; Boyd-Graber, J. L.; and Daumé III, H. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 12-17, 2016*, 1534–1544.

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 North American Chapter of the Association for Computational Linguistics*, 1616–1626.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 5(4):293–331.

Lin, C.-C.; Ammar, W.; Dyer, C.; and Levin, L. 2015. Unsupervised pos induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, 1311–1316.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc. 3111–3119.

Mitchell, J., and Lapata, M. 2008. Vector-based models of semantic composition. In *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 236–244.

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 28th AAAI Conference on Artificial Intelligence*, 1565–1571.

Pichotta, K., and Mooney, R. J. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*.

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 *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 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.

Shinozaki, T., and Kawahara, T. 2007. Hmm training based on cv-em and cv gaussian mixture optimization. In *IEEE Workshop on Automatic Speech Recognition Understanding*, 318–322.

Srivastava, S.; Chaturvedi, S.; and Mitchell, T. M. 2016. Inferring interpersonal relations in narrative summaries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2807–2813.

Valls-Vargas, J.; Zhu, J.; and Ontañón, S. 2014. Toward automatic role identification in unannotated folk tales. In *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Valls-Vargas, J.; Zhu, J.; and Ontañón, S. 2015. Narrative hermeneutic circle: Improving character role identification from natural language text via feedback loops. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2517–2523.