Where have I Heard this Story Before?: A Case-study on Movie Summaries

Anonymous ACL submission

Abstract

People can identify similarities and correspondences between narratives in everyday life. For example, an analogy with the *Cinderella story* may be made in describing the unexpected success of an underdog. In this work, we present an approach for identifying retellings of the same narrative. Our method depends on finding correspondences between narratives, by considering similarities in terms of plot events, resemblances between characters and their attributes, as well as their social relationships in the narrative. We quantify these different facets and infer the best alignment between characters of any pair of narratives, to define a *story-kernel*, which characterizes the similarity between the two narratives, and can be efficiently computed. We empirically evaluate our method on a novel dataset of 577 movie remakes from Wikipedia. Our approach yields a 14% relative improvement in accuracy over a competitive baseline.

1 Introduction

The ability to understand narratives is a fundamental cognitive skill. Humans use narratives to share information, as well as learn social and moral norms (Gottschall, 2012; Miller and Mitchell, 1983). People also routinely invoke narratives to make sense of the world. They accept narratives that adhere to familiarity and our experiences, and reject or reinterpret those that appear unfamiliar (Herman, 2003). Thus, automatic understanding of narratives is a prerequisite towards developing a human-like understanding of the world, and has been a long-standing goal of AI.

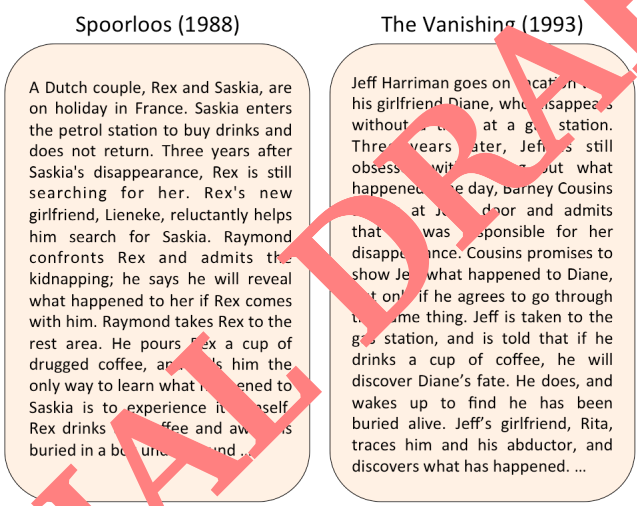

Figure 1: Example of (condensed) movie summaries with similar narratives. The plots correspond to movies *Spoorloos (1988)* and *The Vanishing (1993)*.

Advances in language technologies have traditionally focused on sentence-level processing of language and comparing sentences in terms of their syntax and semantics. This is most evident in their success at tasks such as semantic role labeling (Palmer et al., 2010), paraphrase detection (Madnani and Dorr, 2010), recognizing textual entailment (Dagan et al., 2013), etc. On the other hand, document level understanding of text has been studied only from the narrow perspective of tasks such as entity linking and event coreference resolution (Lee et al., 2013). In particular, the question of identifying when two documents are similar in terms of narrative (cognate with the question of when two sentences are semantically similar) remains largely unexplored despite relevance to multiple domains. As an example, given a news story about a current political event, an analyst working in the digital archives might identify similar stories in the past. Similarly, by analyzing stories from different folk cultures or mythologies, researchers in the humanities can identify interesting similarities and correspondences between stories. This suggests natural value in developing methods and evaluation tasks for narrative level understanding of language.

In this paper, we address the task of identifying similar narratives by taking into account two kinds of likenesses: (1) plot similarity (2) correspondences between characters in the narratives (based on attributes such as name, gender, prominence, and social relationships with other characters). Our approach is based on finding optimal alignments between pairs of narratives, and defining a *story-kernel* between them. An important note is that the approach is unsupervised, and hence does not require labeled data or training[1].

A significant issue in computationally exploring a subjective issue such as narrative similarity is the availability of annotated data for exploration and evaluation. For this work, we created a dataset of plot summaries of movies, which include pairs of movies that have been identified as remakes (see Section 4 for details). The underlying idea is that remakes of the same story would retain prominent elements in terms of narrative theme, even while they look superficially different. Figure 1 shows an example of two such narratives, condensed here for brevity. We frame the ability to identify such retellings of the same story as an objective measure for evaluating narrative similarity.

The rest of this paper is structured as follows. In Section 2, we briefly review related literature in computational narratives and story understanding. In Section 3, we describe our approach and define our *story-kernel* for narrative similarity. In Section 4, we describe our dataset and its basic pre-processing. In Section 5, we present results from empirical evaluation of our approach including quantitative as well as qualitative assessment. Finally, in Section 6, we conclude with a discussion and some directions of future work.

Our main contributions in this work are:

1. We introduce the novel problem of characterizing similarity between narratives, and formulate this as a ranking task.
2. We present a story-kernel that quantifies narrative similarity by considering correspondences between pairs of narratives using a character-centric approach.
3. We present a dataset of 577 narratives for this task, mined from plot summaries of movie remakes from Wikipedia, and hope that this motivates further research in this task.
4. We evaluate the utility of our *story-kernel* and its various components on this dataset and empirically demonstrate that it outperforms competitive baselines.

[1] We do tune parameters on a small development set.

2 Related Work

The field of computational narratology has explored algorithmic understanding and generation of narratives and narrative structures (Mani, 2012; Richards et al., 2009). Most previous work on modeling narrative plots has directly attempted to interpret them in terms of *sequences of events* in the story. These include seminal work on semantic scripts (Schank and Abelson, 2013; Mooney and DeJong, 1985; Regneri et al., 2010), which focuses on representing text in terms of sequences of events and causal relationships between them. More recent approaches have explored statistical induction of temporal series of event schemas or scripts from large volumes of unstructured text (Chambers and Jurafsky, 2009; Cheung et al., 2013; Pichotta and Mooney, 2014), etc. Narrative specific approaches that have taken an event-centric view for representation of plot units include Lehnert (1981); McIntyre and Lapata (2010); Goyal et al. (2010) and Finlayson (2012). As a consequence of viewing narratives solely in terms of events described in them, these approaches do not have any models of characters that persist through a narrative, or their relationships to each other.

On the other hand, other research has taken a *character centric* view of narratives from the perspectives of the principal characters and entities that occur in them (Wilensky, 1978). These have focused on models for identifying Proppian roles or character personas such as the protagonist, the foil, the villain, etc. (Propp, 1968; Bamman et al., 2013, 2014), or assigning such roles to characters in simple folk stories (Valls-Vargas et al., 2014). A notable approach is that of Elsner (2012), who also explore the plot structure of novels to distinguish original texts from novels from artificially permuted versions of the same. Some recent approaches have also explicitly focused on modeling relationships between literary characters (Chaturvedi, 2016; Chaturvedi et al., 2017;

Iyyer et al., 2016), and extracting social networks of characters in a text (Elson et al., 2010; Agarwal et al., 2013; Krishnan and Eisenstein, 2015; Srivastava et al., 2016). However, while the above methods model prototypical patterns that characterize narratives, they do not address the issue of *comparing* narratives. To the best of our knowledge, this is the first work to infer similarities between narratives in context of an objective task.

In terms of technical approach, our method can be seen as close to structured kernels, which define compositional kernels over discrete objects such as graphs (Haussler, 1999; Smalter, 2008; Bai et al., 2015). While we describe our story-kernel as defining a similarity metric between pairs of narratives, it can be seen as a proper Mercer Kernel over the domain of character graph pairs.

3 Identifying Narrative Similarity

In this section we describe our approach in detail. Our goal is to develop an unsupervised method that can *read* a story (such as a movie or a novel summary) and retrieve its re-tellings (from a collection of stories). Our approach's core consists of a *story-kernel*, $\mathbb{S}(s_i, s_j)$ that characterizes the similarity between the two narratives, s_i and s_j. The story-kernel consists of two major components. The first component, described in Section 3.1, incorporates the similarity between the textual content of the two stories using the description of the overall plot. We capture the notion of a plot using the events and entities described in it. The second component, described in Section 3.2, additionally considers the various characters and their attributes mentioned in the story. In the rest of this section we clearly define this story-kernel and its two components.

3.1 Plot Kernel

A natural choice for our story-kernel is to incorporate lexical similarities between the textual descriptions of the two narratives. However, our goal is to identify narratives that have similar plot structures but might not have verbatim correspondences in the summaries. Therefore, in designing the textual representation of a narrative, we only consider events, entities and properties of the entities as described by the narrator. We capture the events mentioned in a story using all verbs occurring in the text of the narrative. We capture entities and their properties by identifying nouns and the adjectives that modify them. However, as described later, we emphasize that characters play a central role in the development of a story and so model them specifically as a separate components in the story-kernel. Hence, at this stage, we only consider entities that do not represent a character.

Finally, we represent the plot of a narrative using a bag-of-words representation of its events and entities (and their characteristics) as described above. We then define $\mathbb{S}_{plot}(s_i, s_j)$ as the cosine similarity between these representations for narratives s_i and s_j.

In our experiments, we evaluate this hypothesis regarding the utility of representing stories using their plots (events, non-character entities and their properties) in Section 5.2.

3.2 Character Alignment Kernel

Section 3.1 compares two given narratives using their overall plot (events and entities). However, the primary goal of fictional (and sometimes also real) stories is to describe certain actors (characters), events concerning them and their social relationships with each other. Taking this character-centric approach to narrative understanding, we decided to design a component in our *story-kernel* which is dedicated to the characters mentioned in the story.

This component compares two narratives by *aligning* each character of one story with a 'similar' character of the other story, and considering the overall alignment score/similarity. In this section, we first describe the alignment process and then describe how we determine similarity between characters.

Character alignment: We begin by aligning characters from the two stories. Specifically, we align each character, c_i, of a story, s_i, to a character, c_j, of the other story, s_j. This alignment is done using a similarity score, $\mathcal{S}(c_i, c_j)$, between the two characters being aligned. The goal of this process is to output an alignment such that it maximizes the combined average alignment score of aligned characters:

$$\mathbb{S}_{character}(s_i, s_j) = max \frac{\sum_{c_i \in s_i, c_j \in s_j} \mathcal{S}(c_i, c_j)}{N}$$

where, N is the total number of aligned characters from the two narratives s_i and s_j

The above combinatorial optimization problem is non-trivial. However, it can be solved in polynomial time by modifying the Hungarian assignment algorithm (Kuhn, 1955), which assigns each character from a story to a character in the other story. In cases when the two stories have different number of characters, we modify the formulation so that extra unaligned characters are aligned to a special *null* character from the other story. In terms of graph theory, the problem can be seen as finding the minimum cost perfect matching in a bipartite graph with non-negative edge weights. In our formulation, an edge's weight is defined as the similarity between the two characters being aligned. If a character is being aligned to a *null* character the similarity or the corresponding edge's weight is 0 (minimum possible value) so as to discourage alignments to *null* characters whenever possible. In the rest of this section we describe the inter-character similarity, $\mathcal{S}(c_i, c_j) \in [0, 1]$ when a character is aligned to another non-null character (from a different story).

Inter-character Similarity: In the above-mentioned procedure for aligning characters, we must evaluate the similarity between any given pair of characters, $\mathcal{S}(c_i, c_j)$. In this work, we assume that the similarity between two characters is a function of their (i) name, (ii) gender, (iii) prominence in the story, and (iv) their social relationships with other characters mentioned in the same story. In particular, we define it as a convex combination of these factors:

$$\begin{aligned} \mathcal{S}(c_i, c_j) &= \lambda_1 \cdot \mathcal{S}_{name}(c_i, c_j) \\ &\quad +\lambda_2 \cdot \mathcal{S}_{gender}(c_i, c_j) \\ &\quad +\lambda_3 \cdot \mathcal{S}_{prominence}(c_i, c_j) \\ +(1 - \lambda_1 - \lambda_2 - \lambda_3) &\cdot \mathcal{S}_{relationship}(c_i, c_j) \end{aligned}$$

(1) In the above equation, $\mathcal{S}_{name}(c_i, c_j)$, tries to identify if the two characters have matching names. It does so by defining $\mathcal{S}_{name}(c_i, c_j) = 1$ if name of c_i is the same as name of c_j, and 0 otherwise. $\mathcal{S}_{name}(c_i, c_j)$ tries to align characters with identical names, and could provide a very strong signal in many cases. For examples, two versions of the folk tale *Beauty and the Beast*, will have the same characters: Belle, her father Maurice, the Beast etc. However, a more interesting case would be when the same story is told with a different set of character-names (for example the two movie-summaries shown in Figure 1), or with fewer or more characters. In such cases, it is essential to not blindly align characters based on their names, but to analyze other attributes such as their gender, prominence in the narrative, and social relationships to other characters. These compose the other factors that define character similarity.

(2) $\mathcal{S}_{gender}(c_i, c_j)$ prefers alignments of characters that have the same gender; i.e. $\mathcal{S}_{gender}(c_i, c_j) = 1$ if gender of c_i is the same as gender of c_j, and 0 otherwise.

(3) $\mathcal{S}_{prominence}(c_i, c_j)$ prefers alignments where the prominence of aligned characters is similar, i.e. it prefers to match prominent characters in a story to prominent characters in the other. Consider a case where two different stories share some character names. For example, they both have a male character named 'Harry', who is the protagonist in one story and a side-character in the story. In such a case, even though the two characters have the same name and gender, we would not want to align them. Therefore, we compute the *prominence* of a character, $prom(c)$, as simply the fraction of all character mentions when (s)he is mentioned in the story, i.e.

$$prom(c) = \frac{\text{number of tokens that refer to c}}{\sum_{c'} \text{number of tokens that refer to c'}}$$

And we define $\mathcal{S}_{prominence}(c_i, c_j)$ as:
$\mathcal{S}_{prominence}(c_i, c_j) = 1 - |prom(c_i) - prom(c_j)|$

(4) The fourth factor, $\mathcal{S}_{relationship}(c_i, c_j)$, considers how the two characters are related to other characters from their respective stories in determining their similarity. This is motivated by the observation that a narrator usually describes some characters, like the protagonist, in a general *positive* light. Such characters, for example, may have a cordial relationship with everybody else in the story. On the other hand, certain characters, such as the villain, are described in a general negative light, and so they may be portrayed as having unpleasant relationships with most other in the story. This factor attempts to discourage such mismatched alignments (like that between a protagonist of a story with the villain of the other story).

There have been previous works that model inter-character relationship in narratives (Srivastava et al., 2016; Chaturvedi, 2016). Most of these methods quantify the relationship between two characters (from the same narrative) using a set of specific features. These features are primarily a set of words extracted from sentences in which the two characters of interest appear together. For example, consider the following sentence depicting relationship between John and Tony: 'John brutally stabs Tony with the knife he had hidden under his shirt'. In this sentence, we extract the following feature-words:

1. the actions that the two characters do to each other in the narrative. For example, 'stabs' in the example sentence.
2. the narrators bias in describing those actions. For example, 'brutally' in the example sentence above.
3. the semantic frames that are evoked with respect to the two characters of interest. For example, a semantic frame called 'Cause_harm' is evoked in the sample sentence above with 'John' and 'Tony' as its frame elements.

For extracting actions and the narrators bias, we consider the dependency parse of such sentences and use various dependency relations. For example for actions, we consider verbs that have the concerned characters as their agents (identified using 'nsubj' and 'agent' dependency relations), and patients (using 'dobj' and 'nsubjpass' relations). For the semantic frame based words we exploit the frame-semantic parse of the sentence.

In this work, we represent a character's relationship with all other characters in the narrative using the features described above. We then compute the relationship-based similarity, $\mathcal{S}_{relationship}(c_i, c_j)$ between two characters, c_i and c_j, using the cosine similarity in this feature space.

3.3 Story-kernel

Sections 3.1 and 3.2 above define the two components of our *story-kernel*: the plot-based component, $\mathbb{S}_{plot}(s_i, s_j)$, and the character-based component, $\mathbb{S}_{character}(s_i, s_j)$. We combine these components by defining the story-kernel, $\mathbb{S}(s_i, s_j)$, as a convex combination of the two:

$$\begin{aligned}\mathbb{S}(s_i, s_j) &= \alpha \cdot \mathbb{S}_{plot}(s_i, s_j) \\ &\quad + (1 - \alpha) \cdot \mathbb{S}_{character}(s_i, s_j)\end{aligned}$$

This allows us to define a prediction rule for our task. Given a narrative, s, as input and a database of other narratives, we output the narrative that is most similar to s. In other words, we output: $argmax_{s'} \mathbb{S}(s, s')$.

4 Movie Remakes Dataset

There are no existing datasets that evaluate document-level similarity of narratives. Hence, one of the contributions of this work is a dataset for evaluating narrative similarity. While any annotations of narrative similarity would be inherently subjective, we chose to use human-provided labels from an external knowledge resource (Wikipedia) as a proxy for narrative similarity. As stated earlier, we assume that movie remakes are retellings of the same story, which retain prominent narrative elements. Hence, a good measure of narrative similarity should evaluate remakes as being 'similar/close' to each other.

Original	Remake(s)
My Name Is Julia Ross (1945)	Dead of Winter (1987)
Diversion (1980)	Fatal Attraction (1987)
Gojira (1954)	Godzilla (1998), Godzilla (2014)
It's a Wonderful Life (1946)	It Happened One Christmas (1977)

Table 1: Examples of movies and their remakes in the dataset.

4.1 Dataset Creation

Our data consists of movie summaries scraped from a December 15, 2016 dump of Wikipedia. In particular, we scraped lists of movies from the 'Lists of film remakes' page on Wikipedia, which consist of entries of movies that are considered remakes of previous movies. Since some movies have been remade multiple times, we obtain clusters of movie plots, each of which share the same narrative theme. In some cases, the remakes are close to the originals at a surface level, whereas in other cases, they diverge greatly at a surface level, and may also differ in the narrative. The movie clusters so obtained were manually pruned to remove scraping errors, resulting in 577 plot summaries in the final dataset.

Table 1 lists names of some movies (and their remakes) in the dataset. We extracted the corresponding plot summaries each of these movies for the evaluation of our narrative similarity task. We note that names of movies shown in the table are representational, and our approach does not use them for adjudging. Table 2 shows the final statistics of the curated dataset. In particular, we observe that the average movie summary is reasonably long, which would make user annotations of similarity for such narratives very difficult.

4.2 Data-Split

Since our unsupervised approach requires parameter selection, we randomly divided the dataset into two parts. We kept 20% of the movies as devel-

Number of movies	577
Number of clusters	266
Avg number of movies per cluster	2.17
Max number of movies in a cluster	7
Avg number of tokens in a summary	564
Max number of tokens in a summary	2778
Min number of tokens in a summary	26

Table 2: Summary statistics for narrative similarity Movie Remakes dataset.

opment set, for tuning parameters. The remaining 80% dataset consisting of 466 movies was treated as the held-out test set. Our final performances are reported on this test set.

4.3 Pre-processing

We pre-processed texts of movie summaries to be usable by our approach. For the component of the kernel that studies plots, we removed stopwords and punctuations. We used a POS-tagger to identify nouns, verbs and adjectives, and a Lemmatizer to lemmatize these words. We used a standard NLP pipeline for these annotations [2].

The component of the kernel studying characters required considerably more pre-processing. We obtained dependency parses of the summary sentences, identified major characters using the BookNLP pipeline (Bamman et al., 2014). This pipeline also clusters various character mentions (apart from coreference resolution). For example, it identifies that 'Elizabeth Bennet', 'Ms. Bennet' and 'Elizabeth' refer to the same character. However, the pipeline is designed for very long documents involving multiple characters, such as novel texts, and we found it to be conservative in resolving co-references. We augmented its output using coreferences obtained from the Stanford Core NLP system (Manning et al., 2014). We obtained the gender information about character mentions using the Stanford Core NLP system. However, this was sometimes noisy. For example, it is possible that various mentions of the same character get assigned to more than one gender (like 'male', and 'neutral'). So, for each character we assign the gender that is most frequently assigned to that character's mentions across the story. Finally, we obtained frame-semantic parses of the text using the Semafor parser (Das et al., 2014).

[2] http://cogcomp.cs.illinois.edu/page/software/

5 Empirical Evaluation

In this section we describe our quantitative and qualitative experiments and results.

5.1 Evaluation Measure

While there are several ways of evaluating document-level similarity, we employ a strict evaluation measure. Given a story we output the most similar story from the database. The output is deemed correct if the input movie and the output movie are a remake of each other (belong to the same remake cluster), and incorrect otherwise. In our experiments we report this as *Accuracy*. From the perspective of information-retrieval, this is equivalent to reporting Precision at 1.

5.2 Evaluating Plot-kernel

Section 3.1 described the component of our kernel that postulates that the task of identifying similar stories can benefit from leveraging their plots. It captures the *overall plot* using only events and non-character entities. In our first experiment, we evaluate this hypothesis. To this end, we evaluate the accuracy of the plot-based component of the kernel $\mathbb{S}_{plot}(s_i, s_j)$ (this is equivalent to setting $\alpha = 1$ in the equation in Section 3.3). Table 3a shows the performance of this kernel on the held-out test set. The first row of the table corresponds to the case when we use all words in the movie summary (after removing stopwords and punctuations and lemmatizing words) as features. This is our primary baseline. The next row corresponds to the case when we represent the plot using only events and entities. In Section 3.1 we included only non-character entities in our plot-kernel because we incorporate characters separately in our character-based kernel. Here, since we do not have a separate component for the characters, we retain those words in the plot definition. We see that using this plot definition ($\mathbb{S}^{++}_{plot}$), the accuracy improves from 55.79% to 57.94% validating our thesis that for this task, it helps to represent a plot using only events and entities.

However, if we remove the words referring to a character from plot definition ($\mathbb{S}_{plot}$, represented by the last row of Table 3a), the accuracy drops considerably to 54.93%. This is expected since matching character names are frequently marker of remakes in our data. In the next experiment, we evaluate how separately modeling information about the characters helps us in this task.

Setting	Accuracy
All words	55.79%
$\mathbb{S}^{++}_{plot}$ (incl. character-mentions)	57.94%
$\mathbb{S}_{plot}$	54.93 %

(a) Evaluating plot based kernel. We can see that the plot based method ($\mathbb{S}^{++}_{plot}$), which considers only events and entities, performs better than one that considers all words. Also, dropping character-mentions ($\mathbb{S}_{plot}$) hurts performance.

Setting	Accuracy
$\mathbb{S}_{plot} + \mathbb{S}^{simple}_{character}$	60.08%
$\mathbb{S}^{++}_{plot} + \mathbb{S}^{simple}_{character}$	57.94%

(b) Evaluations using a simpler character-based kernel, which only considers character-name overlap. Combining this simple character-based kernel with the plot kernel helps in improving performance.

Setting	Accuracy
$\mathbb{S}_{plot} + \mathbb{S}_{character}$	**63.73%**
random	$< 1\%$

(c) Evaluating our character-alignment based kernel. Combining the information about character's name with their gender, prominence and social relationships helps in improving performance over a simpler kernel that considers only character names.

Table 3: Performance of various kernel combinations on the held-out test-set.

5.3 Importance of Character-centric Approach

The previous experiment indicates that characters are important in modeling narrative similarity. In Section 3.2, we designed a special character-alignment based kernel that analyzes not just character name, but also their gender, prominence and social relationships. In this experiment we evaluate if modeling facets of character similarity assists narrative similarity. We also attempt to gauge the value of a simpler character kernel based only on character names, which we refer to as $\mathbb{S}^{simple}_{character}(s_i, s_j)$. This kernel also serves as an alternative baseline to the character-alignment based kernel, $\mathbb{S}_{character}(s_i, s_j)$, described in Section 3.2. We define this alternative kernel, $\mathbb{S}^{simple}_{character}(s_i, s_j)$, using the set of character-names from the two movies C_{s_i} and C_{s_j}:

$$\mathbb{S}^{simple}_{character}(s_i, s_j) = \frac{\text{Intersection of } C_{s_i} \text{ and } C_{s_j}}{\text{Union of } C_{s_i} \text{ and } C_{s_j}}$$

We combine this alternative character-kernel with the plot-based kernels (last two rows of Table 3a) in the same manner described in Section 3.3 (using a parameter α). The parameter is tuned on the development set. Table 3b summarizes our results. When we combine this alternative character-kernel with the plot-based kernel, $\mathbb{S}_{plot}$ (first row of the table), the accuracy improves from 54.93 to 60.08 (with $\alpha = 0.7$). This indicates that it helps to have a special component dedicated to characters while solving this task.

For completeness, we also combine the alternative character-kernel with the plot-based kernel that included character mentions, $\mathbb{S}^{++}_{plot}$, since it was performing better than $\mathbb{S}_{plot}$ when considered in isolation in Table 3a. Interestingly, the accuracy remains the same at 57.94 (second row of Table 3b). In this case, we saw that while tuning the parameter, α, on the development set, the model relied only on plot-based component ($\alpha = 1.0$).

However, we saw that combining the alternative character-kernel with $\mathbb{S}_{plot}$ yields better performance (60.08) than that obtained (57.94) when combining it with $\mathbb{S}^{++}_{plot}$. We had made similar observations on the development set as well. Combining it with $\mathbb{S}_{plot}$ and $\mathbb{S}^{++}_{plot}$ yielded accuracies of about 6[illegible]% and 63% on the development set (not reported in the paper). Therefore, for the rest of the experiments we use $\mathbb{S}_{plot}$ only.

5.4 Evaluating Character-Kernel

In the previous experiment we demonstrated the need to model characters by dedicating a separate, though simplified version of our character-kernel. In this experiment we evaluate the potential of the character-alignment based kernel described in Section 3.2 by comparing it to this simpler alternative. Table 3c describes our results. Comparing the first row of this table (63.73) with the first row of Table 3b (60.08), we can see that our character-alignment based kernel, $\mathbb{S}_{character}$, which considers not only character names, but also their gender, prominence in the story, and relationship with other characters performs better than the simpler baseline character-kernel, $\mathbb{S}^{simple}_{character}$, that considers only character names.

The weights given to individual components of our kernel(s) [3] are shown in Table 4. We observe that the model considers the plot struc-

[3] These weights were obtained during parameter tuning on the development set for the model corresponding to the first row of Table 3c.

Component	Weights
$\mathbb{S}_{plot}$	0.7
$\mathbb{S}_{character}$	0.3
$\mathbb{S}_{character}$ – name	0.4
$\mathbb{S}_{character}$ – gender	0.1
$\mathbb{S}_{character}$ – prominence	0.1
$\mathbb{S}_{character}$ – relationship	0.4

Table 4: Parameters for various components of the story-kernel. The model relies both on plot-structure and characters. For aligning across narrative characters, it primarily uses characters' names and social relationships (and to some extent their genders and prominences).

ture to be most important in determining narrative similarity, gives it a weight of 0.7. The character-alignment based component of the kernel has a weight of 0.3. Among the various sub-components of the character-alignment based component, it relies primarily on name and social relationships (weights of 0.4 each) in aligning characters of a movie with another movie. It yields smaller weightage (0.1 each) to the characters' gender and prominence.

These results validate our assumption that both plot and character similarity are distinct and important facets in evaluating narrative similarity. Further, our character alignment approach yields significantly improved results for the task.

5.5 Qualitative Results and Error Analysis

We next present an illustrative example of character alignment (Figure 2) using our *story-kernel* for the movie-summaries shown in Figure 1. As stated earlier, the story on the right is a remake of the story on the left. However, they do not share any character names. Our method successfully aligns the protagonists of the two narratives: Rex and Jeff. It also aligns Rex's kidnapped girlfriend, Saskia, with Jeff's kidnapped girlfriend, Diane. Rex's new girlfriend, Lieneke, is also successfully aligned with Jeff's new girlfriend, Rita. However, it aligns Saskia's kidnapper, Raymond, with a *null* character, even though the movie's summary mentions Diane's kidnapper, Barney Cousins, and he should have been aligned with Raymond. A cursory analysis reveals that this error occurred because the NLP pipeline could not identify Barney Cousins as an animate character, possibly due to his unusual name. As a result of which the method received as input a summary in which only three characters were identified for the story on the right. Nevertheless, even with this pre-processing error, the method correctly identifies the story on the right as most similar to the story on the left.

Further error analysis reveals that apart from missed character-identification like the one above, other NLP pipeline errors such as missed coreference, are a significant source of other errors.

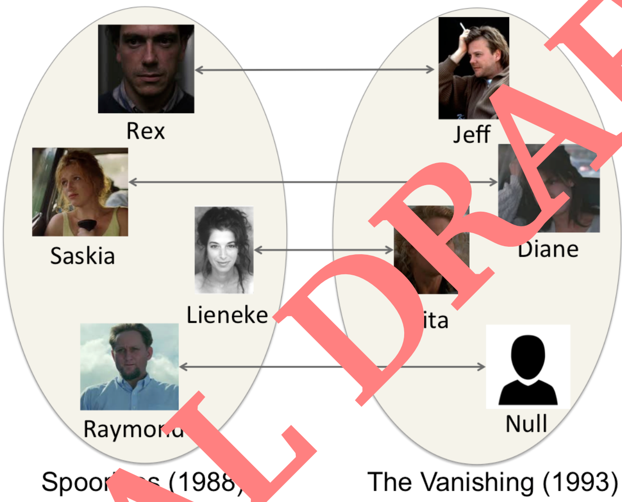

Figure 2: Example of aligned characters from the two movies in Figure 1. All characters, except Raymond, were correctly aligned. Raymond is aligned to a *null* character because the NLP pipeline could not identify the corresponding character in the story on the right.

6 Conclusion

We have presented a method for characterizing correspondences between narratives, which incorporates multiple facets of narrative similarity. We also introduce an objective task and benchmark dataset for quantitative evaluation of metrics of narrative similarity. An interesting consequence of our alignment-based method is that it can suggest across narrative correspondences between characters that bear different names, but serve similar functions in stories. While our test bed in this work was movie summaries, our approach is domain-agnostic and scalable, and can be extended for narratives in other domains such as newswire stories, folk tales and literary fiction. Future work can also sharpen the task by also evaluating character and event alignments between narratives based on established ground truths.

References

Apoorv Agarwal, Anup Kotalwar, and Owen Rambow. 2013. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*. pages 1202–1208.

Lu Bai, Luca Rossi, Zhihong Zhang, and Edwin Hancock. 2015. An aligned subtree kernel for weighted graphs. In *International Conference on Machine Learning*. pages 30–39.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 352–361. http://www.aclweb.org/anthology/P13-1035.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 370–379. http://www.aclweb.org/anthology/P/P14/P14-1035.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, pages 602–610.

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

Snigdha Chaturvedi, Mohit Iyyer, and Hal Daumé III. 2017. Unsupervised Learning of Evolving Relationships Between Literary Characters. In *Proceedings of the Thirty First AAAI Conference on Artificial Intelligence*. AAAI Press.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. pages 837–846. http://aclweb.org/anthology/N/N13/N13-1104.pdf.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers. https://doi.org/10.2200/S00509ED1V01Y201305HLT023.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics* 40(1):9–56. https://doi.org/10.1162/COLI_a_00163.

Micha Elsner. 2012. Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 634–644.

David K. Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting Social Networks from Literary Fiction. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*. pages 138–147.

Mark Alan Finlayson. 2012. *Learning Narrative Structure from Annotated Folktales*. Ph.D. thesis, Massachusetts Institute of Technology.

Jonathan Gottschall. 2012. *The storytelling animal: How stories make us human*. Houghton Mifflin Harcourt.

Amit Goyal, Ellen Riloff, and Hal Daumé III. 2010. Automatically Producing Plot Unit Representations for Narrative Text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 77–86.

David Haussler. 1999. Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, University of California at Santa Cruz, Santa Cruz, CA, USA. http://citeseer.ist.psu.edu/haussler99convolution.html.

David Herman. 2003. *Narrative theory and the cognitive sciences*. 158. Stanford Univ Center for the Study.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan L. Boyd-Graber, and Hal Daumé III. 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, San Diego California, USA, June 12-17, 2016*. pages 1534–1544.

Vinodh Krishnan and Jacob Eisenstein. 2015. "You're Mr. Lebowski, I'm the Dude": Inducing Address Term Formality in Signed Social Networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. pages 1616–1626.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2:83–97.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics* 39(4):885–916.

Wendy G. Lehnert. 1981. Plot Units and Narrative Summarization. *Cognitive Science* 5(4):293–331.

Nitin Madnani and Bonnie J Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics* 36(3):341–387.

Inderjeet Mani. 2012. *Computational Modeling of Narrative*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*. pages 55–60. http://aclweb.org/anthology/P/P14/P14-5010.pdf.

Neil McIntyre and Mirella Lapata. 2010. Plot Induction and Evolutionary Search for Story Generation. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*. pages 1562–1572.

Owen Miller and WJT Mitchell. 1983. On narrative.

Raymond J. Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, USA, August 1985*. pages 681–687.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies* 3(1):1–103.

Karl Pichotta and Raymond J. Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*. pages 220–229.

Vladimir Iakovlevich Propp. 1968. *Morphology of the Folktale*. University of Texas.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. 2010. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 979–988.

Whitman Richards, Mark Alan Finlayson, and Patrick Henry Winston. 2009. Advancing computational models of narrative. *MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, Tech. Rep* 63:2009.

Roger C Schank and Robert P Abelson. 2013. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Aaron Matthew Smalter. 2008. *Kernel Functions for Graph Classification*. Ph.D. thesis, University of Kansas.

Shashank Srivastava, Snigdha Chaturvedi, and Tom M. Mitchell. 2016. Inferring interpersonal relations in narrative summaries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*. pages 2807–2813. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12173.

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. 2014. Toward automatic role identification in unannotated folk tales. In *Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. AAAI Press, pages 188–194.

Robert Wilensky. 1978. *Understanding Goal-Based Stories*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York.