AligNarr: Aligning Narratives on Movies

Paramita Mirza, **Mostafa Abouhamra**, and **Gerhard Weikum**
Max Planck Institute for Informatics
Saarbrücken, Germany
{paramita, mostafa, weikum}@mpi-inf.mpg.de

Abstract

High-quality alignment between movie scripts and plot summaries is an asset for learning to summarize stories and to generate dialogues. The alignment task is challenging as scripts and summaries substantially differ in details and abstraction levels as well as in linguistic register. This paper addresses the alignment problem by devising a fully unsupervised approach based on a global optimization model. Experimental results on ten movies show the viability of our method with 76% F1-score and its superiority over a previous baseline. We publish alignments for 914 movies to foster research in this new topic.

1 Introduction

Motivation and Problem. An important aspect of language understanding is the ability to produce a concise and fluent summary of stories, dialogues and other textual contents. Automatic text summarization is a long-standing topic in natural language processing (Nenkova and McKeown, 2012; Dey and Das, 2020), with numerous approaches for a variety of inputs, largely focusing on news articles and scholarly publications (e.g., See et al. (2017); Hardy et al. (2019); Lev et al. (2019)).

In this paper, our focus is on less explored *narrative* texts such as books and movie scripts. Our goal is to automatically align scenes from movie scripts with sentences from plot summaries. Such alignments support story browsing and explorative search over screenplays (e.g., find all love scenes), and can also be an asset towards improving summarization and text-generation models for dialogues and other narratives.

Figure 1 shows an example: a scene snippet from the movie script of *Shrek*, and its corresponding sentence from the plot description of the movie's Wikipedia article. Establishing this alignment is challenging for three reasons:

- *Input Length:* Movies have many scenes (often more than a hundred), with longer dialogues or multi-person conversations. Plot summaries, on the other hand, are much shorter (e.g., 700 words for *Shrek* on Wikipedia).
- *Disparate Registers:* Scripts and summaries have fundamentally different registers (i.e., language styles, vocabulary and structure). Scripts are dominated by direct speech in dialogues, whereas plot summaries consist of, often complex, descriptive sentences and may introduce abstractions (e.g. *"fell in love ..."* instead of giving details on dating, kissing etc.).
- *Disparate Granularities:* Scripts contain every detail of the screenplay, whereas summaries focus on salient points and can leave out less important sub-stories. Thus, the units in scripts–*scenes*–and the units in plot summaries–*sentences*–are difficult to match.

Narrative Alignment Task: Given a *script* S consisting of a sequence of m scenes $\{s_1, s_2, ..., s_m\}$ and a *summary* U of n sentences $\{u_1, u_2, ..., u_n\}$, the narrative alignment task is to find a mapping between S and U, where both sides can be partial (i.e., some scenes and some sentences are not mapped) and certain constraints are satisfied.

Prior Work and its Limitations. The task of aligning narratives across different registers, like script dialogues and plot summaries, has not received much attention before. Gorinski and Lapata (2015) proposed a graph-based summarization method for movie scripts, exploiting given alignments between script scenes and plot sentences, to select a chain of scenes representing a film's story. Their focus was on the generation of the textual summary, and the alignment itself was addressed merely by simple best-match heuristics based on Nelken and Shieber (2006). Nevertheless, as this work is the relatively closest to ours, it is treated as

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 427–433
August 1–6, 2021.

Script:

Suddenly the magic of the spell pulls Fiona away. She's lifted up into the air and she hovers there while the magic works around her. Suddenly Fiona's eyes open wide. She's consumed by the spell and then is slowly lowered to the ground.

SHREK: (going over to her) Fiona? Fiona. Are you all right?
FIONA: (standing up, she's still an ogre) Well, yes. But I don't understand. I'm supposed to be beautiful.
SHREK: But you ARE beautiful.

Summary:

Fiona is bathed in light as her curse is broken but is surprised that she is still an ogre, as she thought she would become beautiful, to which Shrek replies that she is beautiful

Figure 1: Snippet from *Shrek*'s script, and its summary sentence from *Shrek*'s Wikipedia article.

the baseline against which we evaluate our method. Tapaswi et al. (2015) used a graph-based method to compute an alignment between book chapters and video scenes using matching dialogues and characters as cues. As far as we know, our work is the first in-depth investigation of the narrative alignment task between movie scripts and plot summaries.

Approach and Contributions. We model the narrative alignment task as a global optimization over the possible pairs of scene-sentence mappings. To cope with disparate language registers, we devise embedding-based similarity measures. To cope with the length issue and different granularities, we design this for partial mappings where not all scenes and not all sentences need to be mapped. Typically, a notable subset of scenes is left out, but most sentences are aligned. To keep the alignments concise, we constrain the number of scenes that a sentence can be mapped to, and vice versa. Furthermore, we assume that script and summary both follow the chronology of events in the movie. This is modeled as a constraint for approximate order-preservation. All these considerations are cast into an Integer Linear Program (ILP).

The salient contributions of our work are:

- a fully unsupervised methodology using ILP for aligning two narratives, and
- an aligned corpus of movie scripts and plot summaries for 914 movies, which can serve as training data for text summarization and story generation tasks.

2 Approach

Our alignment method, AligNarr, has three steps: *(i) pre-processing*, which includes linking names found in both inputs, *(ii) building a similarity matrix* between the text units of the two narratives, and *(iii) constructing the alignment mapping* given the similarity matrix as input.

2.1 Pre-Processing

Given a movie script S and its summary U, we first segment them into corresponding units s_i and u_j, which are *scenes* and *sentences* respectively. An interior or exterior indicator 'INT.' or 'EXT.' is commonly used to mark a *scene heading*–separating different scenes–followed by a location or setting. A scene usually contains narrative descriptions as well as dialogue lines, as shown in Figure 1.

Linking Story Entities. We retrieve all phrases that are capitalized, as well as speaker names that start the dialogue lines in a given script, as *candidate names*, excluding the beginning of sentences. However, in movie scripts it is often the case that words are in all-capitals for emphasis, e.g., 'ARE' in Figure 1. Therefore, we first ran *Truecaser*[1] (Lita et al., 2003) to avoid having such words identified as candidate names.

For each pair of collected candidate names, we compute string similarity based on Levenshtein distance using *FuzzyWuzzy*[2]. Given the distance matrix between pairs of names, we then cluster the names using the DBSCAN algorithm (Ester et al., 1996) in order to have a cluster of names representing one *story entity*, e.g., E_{40}: {'Fiona', 'FIONA', 'Princess Fiona'}.

To resolve pronouns, we run *AllenNLP coreference resolution*[3], an end-to-end neural model (Lee et al., 2017) leveraging SpanBERT embeddings (Joshi et al., 2020). All occurrences of clustered names in the script and summary are then replaced with the corresponding entity identifier (e.g., E_{40}). Note that we only consider linking story entities appearing in the summary, since they represent a subset of story entities that are central to the story.

[1] github.com/nreimers/truecaser
[2] github.com/seatgeek/fuzzywuzzy
[3] demo.allennlp.org/coreference-resolution

2.2 Similarity Matrix

We investigate three methods to measure similarity between units of script S and summary U:

Document Relevance Score. After removing stop words and punctuation, we compute the relevance scores of script units $\{s_1, ..., s_m\}$ (as the document collection D), for a given summary unit u_j (as the query q), using a ranking function. In this work, we use BM25 (Robertson and Zaragoza, 2009), a TF-IDF-based ranking function.

Word Overlap Score. We consider the sum of intersecting story entities and words (excluding stop words) that are similar (e.g., 'married' in s_i and 'wedding' in u_j), weighted by their similarity scores. As the similarity score between two words, we take the cosine similarity of word2vec embeddings (Mikolov et al., 2013); words are considered to be similar if their cosine similarity is above 0.5.

Sentence Similarity Score. We first compute sentence embeddings for a given summary unit u_j and all sentences in a script unit s_i, using RoBERTa (Liu et al., 2019) in *Sentence-Transformers*[4] (Reimers and Gurevych, 2019) optimized for the task of Semantic Textual Similarity (stsb-roberta-large). Taken as the similarity score is the highest cosine similarity between u_j's embeddings and embeddings of sentences in s_i. For practical reasons, we only compute sentence similarity scores for pairs of script and summary units with non-zero word overlap scores.

2.3 Alignment Mapping

Given a similarity matrix between units of script $S = \{s_1, ..., s_m\}$ and summary $U = \{u_1, ..., u_n\}$, we devise an Integer Linear Programming (ILP) model to optimize the overall alignment mapping as follows:

Objective Function. We want to maximize the story coherence between S and U in terms of textual similarity between the units: $\max \sum_i \sum_j sim(s_i, u_j) \cdot X_{ij}$, where $sim(s_i, u_j)$ is a numeric feature indicating the similarity or relatedness of s_i and u_j resulting from the previous step, and X_{ij} is a decision variable: $X_{ij} = 1$ if s_i and u_j are aligned, 0 otherwise.

Constraints. We define the following constraints to make sure that the alignment mapping follows the linear constraint of both narratives:

- Each summary sentence can only be aligned with at most r scenes: $\sum_j X_{*j} \leq r$.
- Each summary sentence can only be aligned with a block of r consecutive scenes: $\sum_i \sum_j \sum_k X_{ij} + X_{kj} \leq 1$ if $k \geq i + r$ and $\sum_i \sum_j \sum_k X_{ij} + X_{kj} \leq 1$ if $k \leq i - r, i \geq r$.
- The next summary sentence can only be about the same or the next scenes: $\sum_i \sum_j \sum_k X_{ij} + X_{kj+1} \leq 1$ if $k < i, j < n - 1$.
- The previous summary sentence can only be about the same or the previous scenes: $\sum_i \sum_j \sum_k X_{ij} + X_{kj-1} \leq 1$ if $k > i, j > 0$.

Candidate Space Pruning. To speed up the ILP inference, we exclude pairs of script and summary units, s_i and u_j, which are unlikely to be aligned. We employ the following pruning conditions:

- Given a summary unit u_j, we only consider scenes that yield similarity scores above θ in the ranked list of scenes.
- Given the most similar scene s_{top} to a summary unit u_j, we only consider scenes s_i in which $sim(s_{top}, u_j) - sim(s_i, u_j) < \sigma$.
- The candidate pairs (s_i, u_j) are within the diagonal line boundaries as depicted in Figure 2, by considering only (i, j) pairs that satisfy $j < ni/m + \tau n$ and $i < mj/n + \tau m$ with hyper-parameter τ.

3 Experiments

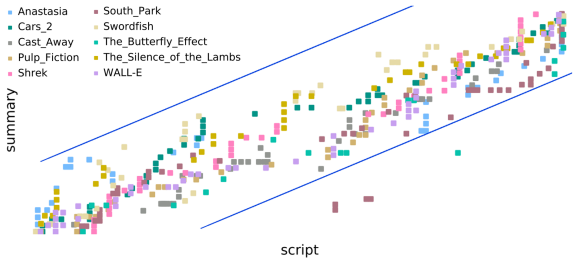

Figure 2: Ground truth alignment for ten movies.

Dataset. We used the ScriptBase corpus[5] (Gorinski and Lapata, 2015, 2018) that contains preprocessed scripts (with various automatic annotations and scene segmentation), along with the corresponding plot summaries taken from Wikipedia. Data statistics are given in Table 3. Two annotators manually created the alignment mappings for ten movies with varying script lengths, yielding inter-annotator agreement of 0.79 Fleiss' κ. The

[4] sbert.net/

[5] github.com/EdinburghNLP/scriptbase

	P	R	F1
Gorinski and Lapata (2015)	.520	.739	.482
AligNarr $_{\text{bm25}}$	.757	.719	.737
AligNarr $_{\text{bm25·w2v}}$	.789	.716	.746
AligNarr $_{\text{bm25·sts}}$	.789	**.734**	**.756**
AligNarr $_{\text{bm25·sts·w2v}}$	**.808**	.720	.754
AligNarr $_{\text{bm25·sts non-ILP}}$	.690	.717	.702

Table 1: AligNarr's performance against baseline.

movie	bm25·sts·w2v			bm25·sts·bert		
	P	R	F1	P	R	F1
Shrek	.85	.80	.82	.92↑	.90↑	.91↑
Pulp Fiction	.92	.86	.89	.89	.85	.87
Cars 2	.84	.72	.77	.87↑	.74↑	.79↑
The Silence of the Lambs	.86	.78	.81	.82	.78	.80
Anastasia	.87	.78	.82	.89↑	.79↑	.83↑
South Park: Bigger, Lo...	.83	.72	.76	.82	.71	.75
Wall-E	.92	.72	.79	.85	.74	.78
Swordfish	.75	.65	.68	.70	.61	.64
The Butterfly Effect	.63	.61	.62	.66↑	.61	.63↑
Cast Away	.61	.56	.58	.59	.56	.57
average	**.81**	**.72**	**.75**	**.80**	**.73↑**	**.76↑**

Table 2: AligNarr $_{\text{bm25·sts·w2v}}$ vs AligNarr $_{\text{bm25·sts·bert}}$.

ground-truth alignments (for which both annotators agree) are shown in Figure 2. These mappings confirm our intuition that a summary normally follows the corresponding script narration in a linear manner, with very few exceptions.

Hyper-Parameters. We defined r (in Section 2.3) as the average ratio of scenes to summary sentences $\lceil m/n \rceil$ based on ten movies, setting it to $r = 5$. θ was set to the 50^{th} percentile (i.e., the median). σ was set to the standard deviation of similarity scores for all scenes given the summary sentence u_j. Hyper-parameter τ, for pruning elements outside the diagonal line boundaries, was set to 0.3.

Baseline. Gorinski and Lapata (2015) used a classifier with sentence-level features (lemma overlap and word stem similarity) to compute sentence-to-sentence alignments. These aligned sentences were then used to identify aligned scene-sentence pairs forming the "gold chain" of scenes in this work (which focused more on the subsequent summarization task), in which a scene contains at least one sentence aligned with a summary sentence. They reported a precision of .53 at a recall rate of .82 for four movies. We re-ran their aligner (provided by the authors) on ten movies in our dataset.

4 Results and Discussion

We report macro-averaged precision (P), recall (R) and F1 results in Table 1. The best performing AligNarr variant, which runs ILP on the combination of document relevance and sentence similarity scores (AligNarr $_{\text{bm25·sts}}$) outperforms the baseline by a large margin on precision and F1-score.

Ablation Study. Document relevance scores alone (AligNarr $_{\text{bm25}}$) already yield very good performance with .737 F1-score averaged over ten movies. When combined with word overlap scores (AligNarr $_{\text{bm25·w2v}}$), the overall performance is further improved to .746 F1-score. Word overlap scoring using word embeddings is particularly useful when the summary uses different vocabulary, for example, using "...a *growing seedling*" for describing a scene with "...a small *plant* in its early stage of *growth*." Combining document relevance scores with sentence similarity scores (AligNarr $_{\text{bm25·sts}}$) results in the best performance with .756 F1 score. Adding word overlap scores on top of that (AligNarr $_{\text{bm25·sts·w2v}}$) yields higher precision of .808 but unfortunately at a lower recall rate of .720. Detailed comparisons and runtime are available in Appendix A and B.

We explored different strategies to combine the similarity matrices, and found element-wise matrix multiplication to perform the best.

Global vs. Local Alignments. To assess the benefit of using ILP, we devised an alignment algorithm focusing on finding the best scene alignment per summary sentence, that is, locally without using the ILP. Given a ranked list of scenes for a given summary sentence, we greedily pick scene-sentence pairs while observing the constraints on at most r consecutive scenes and the diagonal boundary for order-preservation. This local alignment algorithm results in .702 F1-score (AligNarr $_{\text{bm25·sts non-ILP}}$), showing the advantage of computing alignment mappings via global optimization.

Principal Limitation. The ILP constraints and diagonal line boundaries for candidate space pruning (presented in Section 2.3) are too restrictive to allow for 100% F1-score. Considering only candidate pairs that are within the diagonal line boundaries yields in reduced recall of .993, leading to F1-score of .997. If we also take into account all constraints employed by the ILP, recall is further reduced to .944, leading to F1-score of .969.

Contextual Embeddings. We also investigate the utility of contextual embeddings for computing word overlap scores. Specifically, we utilized a pretrained BERT model (bert-large-uncased) from Huggingface (https://huggingface.co/

movie	#scenes	#summary sentences	ratio	P	R	F1
Shrek	35	38	0.9	.89	.89	.89
Pulp Fiction	85	30	2.8	.89	.88	.88
Cars 2	113	36	3.1	.85	.75	.80
The Silence of the Lambs	136	28	4.9	.80	.79	.79
Anastasia	114	31	3.7	.80	.75	.77
South Park: Bigger, Lo...	120	41	2.9	.81	.72	.75
Wall-E	71	35	2.0	.84	.73	.77
Swordfish	*193*	*29*	*6.7*	*.76*	*.66*	*.70*
The Butterfly Effect	*182*	*17*	*10.7*	*.65*	*.61*	*.63*
Cast Away	*300*	*32*	*9.4*	*.60*	*.56*	*.58*
average	**135**	**32**	**4.7**	**.79**	**.73**	**.76**

Table 3: AligNarr $_{\text{bm25·sts}}$'s performance on ten movies.

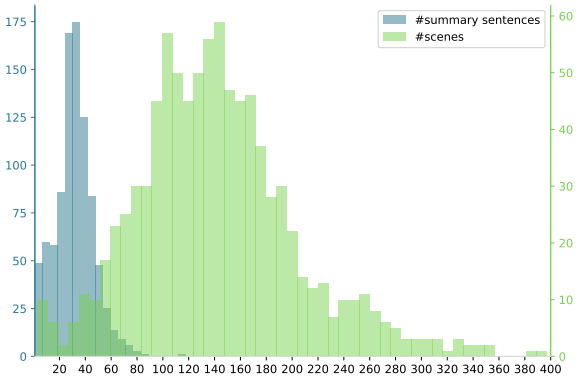

Figure 3: Histograms of number of scenes and summary sentences in ScriptBase movies.

transformers/model_doc/bert.html) to embed sentences from a given pair of script and summary units. We then retrieved individual vectors for each token (i.e., wordpiece) by summing together the outputs of BERT's last four layers.

For each token in the summary unit u_j, we look for similar tokens in the script unit s_i by computing cosine similarity of their embeddings, and take the highest one from each sentence as our intersecting tokens (only if their cosine similarity is above 0.7). Finally, BERT-based word overlap scores are the sum of overlapping tokens (excluding stop words) weighted by their cosine similarity.

Replacing word2vec embeddings with BERT embeddings (AligNarr $_{\text{bm25·sts·bert}}$ in Table 2) yields better performance for some movies like *Shrek*, *Cars 2* and *Anastasia*), which interestingly belong to the same genre (animation). The better performance may be attributed to the ability of BERT to better represent less common words (e.g., *ogre*) using contextual information. However, the overall performance is comparable with the performance of AligNarr $_{\text{bm25·sts·w2v}}$, which requires much less computing time (see Appendix B).

Movie Comparison. AligNarr's performance per movie is shown in Table 3. We observed a trend that the higher the ratio of scenes to summary sentences, the worse the alignment performance, particularly for three movies with ratio above r (average ratio, $r = 5$). This is potentially useful for estimating AligNarr's performance on other movies, which is negatively correlated to the compression rate of a given summary. The most difficult movie to align is *Cast Away*, where *(i)* there was only one active *story entity* throughout the narration, *(ii)* the summary is highly abstract (e.g., *"He also has regular conversations and arguments with Wilson."*), and *(iii)* the story plots and entity names do not fully match, possibly due to the outdated script version.

Data and Code. We provide alignments by AligNarr for ten movies at d5demos.mpi-inf.mpg.de/alignarr/experiments; the same platform was used to manually annotate the alignment mappings. The code for producing the alignments is published at github.com/paramitamirza/AligNarr.

We applied the best performing AligNarr $_{\text{bm25·sts}}$ on the ScriptBase corpus[6] (Gorinski and Lapata, 2015, 2018), leveraging the XML version of movie scripts in *ScriptBase-J* and Wikipedia plot summaries from *ScriptBase-alpha*, totaling to 914 movies. Figure 3 shows the histograms of number of scenes and summary sentences in the corpus, with most summaries containing 20-40 sentences and most scripts consisting of around 100-180 scenes. The alignment mappings for those movies are made available for viewing and downloading at d5demos.mpi-inf.mpg.de/alignarr/scriptbase.

References

Monalisa Dey and Dipankar Das. 2020. A deep dive into supervised extractive and abstractive summarization from text. In *Data Visualization and Knowledge Engineering*, pages 109–132. Springer.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of KDD'96*, pages 226–231.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of NAACL-HLT'15*, pages 1066–1076.

[6] github.com/EdinburghNLP/scriptbase

Philip John Gorinski and Mirella Lapata. 2018. What's this movie about? a joint neural network architecture for movie content analysis. In *Proceedings of NAACL-HLT'18*, pages 1770–1781.

Hardy Hardy, Shashi Narayan, and Andreas Vlachos. 2019. HighRES: Highlight-based reference-less evaluation of summarization. In *Proceedings ACL'19*, pages 3381–3392.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of EMNLP'17*, pages 188–197.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. TalkSumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *Proceedings of ACL'19*, pages 2125–2131.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. tRuEcasIng. In *Proceedings of ACL'03*, pages 152–159.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13*.

Rani Nelken and Stuart M. Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *In Proceedings of EACL'06*.

Ani Nenkova and Kathleen R. McKeown. 2012. A survey of text summarization techniques. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 43–76. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP'19*, pages 3982–3992.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL'17*, pages 1073–1083.

Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. 2015. Book2movie: Aligning video scenes with book chapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1827–1835.

movie	AligNarr $_{bm25}$			AligNarr $_{bm25\cdot w2v}$			AligNarr $_{bm25\cdot sts}$			AligNarr $_{bm25\cdot sts\cdot w2v}$		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Shrek	.85	.84	.85	.85	.81	.83	.89↑	.89↑	.89↑	.85	.80	.82
Pulp Fiction	.85	.86	.86	.88↑	.85	.86	.89↑	.88↑	.88↑	.92↑	.86	.89↑
Cars 2	.86	.76	.80	.81	.71	.75	.85	.75	.80	.84	.72	.77
The Silence of the Lambs	.76	.73	.75	.85↑	.76↑	.80↑	.80↑	.79↑	.79↑	.86↑	.78	.81↑
Anastasia	.79	.75	.77	.87↑	.78↑	.82↑	.80↑	.75	.77	.87↑	.78↑	.82↑
South Park: Bigger, Lo...	.80	.74	.77	.79	.71	.74	.81↑	.72	.75	.83↑	.72	.76↑
Wall-E	.78	.71	.74	.86↑	.75↑	.79↑	.84↑	.73↑	.77↑	.92↑	.72	.79↑
Swordfish	.65	.62	.63	.71↑	.63↑	.66↑	.76↑	.66↑	.70↑	.75	.65	.68
The Butterfly Effect	.62	.61	.61	.62	.58	.60	.65↑	.61	.63↑	.63	.61	.62
Cast Away	.61	.57	.59	.65↑	.58↑	.61↑	.60	.56	.58	.61↑	.56	.58
average	**.76**	**.72**	**.74**	**.79↑**	**.72**	**.75↑**	**.79↑**	**.73↑**	**.76↑**	**.81↑**	**.72**	**.75**

Table 4: AligNarr's performance on ten movies (ablation study).

movie	text pre-processing	ILP (bm25·sts)	computing similarity matrix bm25	w2v	sts	bert
Shrek	5.5	4.1	0.02	69.7	131.3	1972.5
Pulp Fiction	20.8	12.3	0.03	222.0	172.0	2576.2
Cars 2	56.5	36.1	0.04	168.9	210.2	3169.8
The Silence of the Lambs	29.5	29.4	0.04	329.1	199.9	2439.2
Anastasia	21.6	28.6	0.03	147.6	147.8	1919.6
South Park: Bigger, Lo...	62.2	1350.4	0.04	165.4	251.9	3603.4
Wall-E	6.2	11.6	0.02	172.5	169.3	2771.6
Swordfish	17.0	457.4	0.04	143.4	152.6	1893.6
The Butterfly Effect	8.1	84.2	0.05	233.9	130.0	1297.2
Cast Away	8.3	329.4	0.04	237.4	294.8	3069.5
average	**23.6**	**234.4**	**0.04**	**189.0**	**186.0**	**2471.2**
computing infrastructure	4x Intel(R) Xeon(R) Gold 6136 # of cores: 48 # of threads: 96 Memory: 1.5TB		1x AMD EPYC 7502P # of cores: 32 # of threads: 64 Memory: 1TB GPU: 4x NVIDIA Quadro RTX 8000, 48 GB GDDR6			

Table 5: AligNarr's runtime (in seconds).

A Detailed Ablation Study

We report in Table 4 the ablation study on AligNarr's performance using different similarity matrices on ten movies. In general, leveraging word-based (w2v) and sentence-based (sts) semantic similarity scores via embeddings, in addition to document relevance scores (bm25), results in significantly higher precision for some movies, while recall remains more or less stable.

B AligNarr's Runtime

In Table 5 we detail the average runtime of the best performing AligNarr $_{bm25\cdot sts}$, along with the computing infrastructure used.

Note that to compute sentence similarity scores (sts) we need word overlap scores via word2vec embeddings (w2v) to filter out scene-sentence pairs that are unlikely to be similar, in order to speed up the runtime. Computing word overlap scores using BERT embeddings (bert) requires almost 13 times the time of computing the scores with word2vec embeddings.