STPar: A Structure-Aware Triaffine Parser for Screenplay Character Coreference Resolution

Li Zheng[1], **Hao Fei**[2], **Lei Chen**[1], **Bobo Li**[1],
Fei Li[1*], **Chong Teng**[1], **Liang Zhao**[3], **Donghong Ji**[1]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University, Wuhan, China
[2]National University of Singapore, Singapore, Singapore
[3]University of Sao Paulo, Sao Paulo, Brazil
{zhengli,chhenl,boboli,lifei_csnlp,tengchong,dhji}@whu.edu.cn
haofei37@nus.edu.sg, zhao@usp.br

Abstract

Character Coreference Resolution in Movie Screenplays (MovieCoref) is a newly emerging task for understanding complex movie plots and character relationships. This task poses greater challenges than traditional coreference resolution, due to the intricate narrative structures and character interactions unique to screenplays. In light of these challenges, we introduce a novel approach: a Structure-aware Triaffine Parser (STPar) for the MovieCoref task. STPar combines discourse and syntactic structures in the feature encoding process, enabling comprehensive analysis of ternary relationships and complex interactions. During the pairing process, STPar utilizes a triaffine scorer to consider high-order relations between candidate mention pairs, thus enhancing its ability to capture detailed narrative structures. In addition, STPar incorporates multi-task learning, encompassing singleton and span detection tasks, to further improve coreference resolution performance. Our evaluations on the MovieCoref dataset demonstrate that STPar significantly outperforms the best baseline by 7.4%, 21.5%, 7.1%, and 10.2% in F1 scores of B^3, CEAF$_e$, LEA, and CoNLL. Further analysis highlights the benefits of integrating structural discourse and syntactic information as well as the combined approaches of triaffine and multi-task learning.[1]

1 Introduction

Coreference resolution (Lee et al., 2017; Dai et al., 2019; Lu and Ng, 2021; Zheng et al., 2024a), a pivotal technique in natural language processing (NLP), has significantly advanced the accuracy of text comprehension and information extraction. This task focuses on identifying and associating entity mentions within a text, such as people, places, organizations, etc., to determine whether they refer to the same entity. It plays a vital role in many downstream tasks, including sentiment analysis (Zheng et al., 2023a,b), relation extraction (Feng et al., 2025; Yuan et al., 2024), and complex task reasoning (Zheng et al., 2024b, 2025). Character Coreference Resolution in Movie Screenplays (MovieCoref) (Baruah et al., 2021) is an emerging and practical task in media analysis, which aims to help audiences understand the movie plot and disentangle the intricate character relationships.

Traditional coreference resolution methods primarily focus on dialogue and news texts, modeling the context as sequences (Zhang et al., 2023b; Wu et al., 2020) or diverse graphs (Jiang and Cohn, 2022, 2021) to learn feature representations. In contrast, movie screenplay scenarios introduce additional and richer information (e.g., scene descriptions, dialogues), with intricate clues and character relationships. As a result, it is challenging to directly transfer existing coreference resolution approaches to solve the MovieCoref task. Recent efforts (Baruah et al., 2021; Baruah and Narayanan, 2023) establish coreference models by designing structural rules and scoring word pairs, yielding certain improvement. However, the lack of in-depth structural analysis and modeling of the movie screenplay greatly hampers their performance.

To solve the MovieCoref task, we conduct comprehensive analysis of movie screenplay scenarios from both global and local perspectives, and identify the following core challenges in constructing

[*] Corresponding author.
[1] Code available at https://github.com/ZhengL00/STPar.

Transactions of the Association for Computational Linguistics, vol. 13, pp. 923–937, 2025. https://doi.org/10.1162/tacl.a.28
Action Editor: Vincent Ng. Submission batch: 10/2024; Revision batch: 11/2024; Published 8/2025.

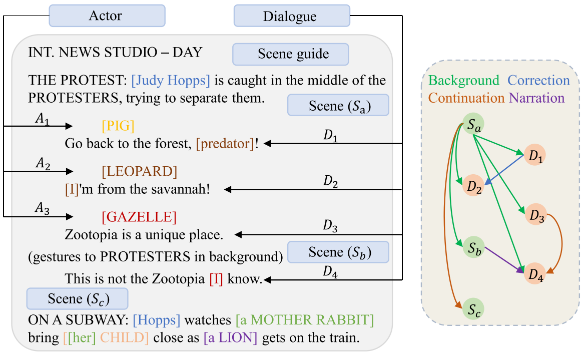

Actor
Dialogue
INT. NEWS STUDIO – DAY
Scene guide
THE PROTEST: [Judy Hopps] is caught in the middle of the PROTESTERS, trying to separate them.
Scene (S_a)
A_1
[PIG]
Go back to the forest, [predator]!
D_1
A_2
[LEOPARD]
[I]'m from the savannah!
D_2
A_3
[GAZELLE]
Zootopia is a unique place.
D_3
(gestures to PROTESTERS in background)
Scene (S_b)
This is not the Zootopia [I] know.
D_4
Scene (S_c)
ON A SUBWAY: [Hopps] watches [a MOTHER RABBIT] bring [[her] CHILD] close as [a LION] gets on the train.
Background
Correction
Continuation
Narration
S_a
D_1
D_2
D_3
S_b
D_4
S_c

Figure 1: **Left:** An example screenplay excerpt from the movie Zootopia. Like-colored mentions are co-referring. **Right:** Discourse dependency parsing structure. Different colored lines represent different types of discourse dependencies.

models for character coreference resolution: ❶ Labyrinth of character and plot clues caused by complex screenplay structures. ❷ Character coreference resolution difficulty caused by the shortage of high-order information.

For the clues labyrinth challenge, as depicted in Figure 1, it is evident that screenplays possess a rich and intricate composition, encompassing not only dialogues between characters but also additional elements such as actors, and scene descriptions. Moreover, the discourse dependencies between characters are often spread across multiple sentences. Simply concatenating the screenplay into a long flat sequence lacks the ability to differentiate and filter out different clue information, resulting in the disruption of the screenplay's structure and the interruption of clues between characters. For example, the mention of ''Judy Hopps'' in S_a and ''Hopps'' in S_c exhibit a coreferent relationship, but there are five sentences between them, making the clues easily interrupted. However, if we consider the discourse dependency relationships in the screenplay structure, such as ''Continuation'' between S_a and S_c, it facilitates predicting the coreferent relationship between ''Judy Hopps'' in S_a and ''Hopps'' in S_c. Therefore, explicitly modeling the multi-level structure of the screenplay enables effective navigation through the clues labyrinth.

For the coreference ambiguity challenge, there are numerous complex interactions between characters in screenplays. For instance, in Figure 1, there exist complex interactions among the mentions ''Hopps'', ''a MOTHER RABBIT'' and ''her'' due to their close-related context. If the model solely relies on pairwise scoring between two mentions ''her'' and ''Hopps'' for coreference resolution, there is a risk of incorrectly referring ''her'' to ''Hopps'' instead of the correct antecedent ''a MOTHER RABBIT''. However, if we already know that ''her'' and ''a MOTHER RABBIT'' are coreferent and introduce such high-order information for coreference resolution between ''her'' and ''Hopps'', the model is more likely to make correct prediction.

Furthermore, after surveying the data, we have found that movie screenplays often contain singleton mentions (about 15.6% of all characters), which refer to unique characters that exhibit distinct behaviors and traits compared to other entities. Failing to accurately identify and differentiate singleton mentions lead to confusion in identifying non-singleton mentions and result in the performance deterioration for coreference resolution. Therefore, distinguishing singleton mentions can effectively eliminate coreference ambiguities and provide a more precise solution to the MovieCoref task.

Based on the aforementioned observations, in this paper, we propose a ***S**tructure-aware **T**riaffine **Par**ser (STPar)* for the MovieCoref task. The primary goal of STPar is to interconnect various elements, effectively aggregate discourse information and high-order information in the screenplay. STPar has three main characteristics: ❶ It explicitly models the discourse and syntactic structures of screenplays utilizing dialogue discourse dependency relationships and syntactic dependency relationships, respectively. By combining such structural information with graph attention networks, movie screenplay structures can be well captured in both discourse and sentence levels. ❷ It is equipped with a triaffine scorer to take the third mention as input when calculating the pairwise score of the target pair of mentions. The intuition is that the third mention is considered as high-order information and a supplemental clue for disambiguating in coreference resolution. ❸ It performs multi-task learning and inference, including singleton detection and span detection tasks, to further boost the performance of MovieCoref. The former is to alleviate the negative impact of the coreference chains owning only one mention and the latter is to prune low quality mention candidates.

We perform experiments on the benchmark MovieCoref dataset. The experimental results show that our model significantly outperforms the best baseline Fusion-based model by 7.4%, 21.5%, 7.1%, and 10.2% in F1 scores of B^3, $CEAF_e$, LEA, and CoNLL. Further ablation experiments indicate that each component of our framework is essential. Specifically, applying the graph component in our model leads to an increase of 4.2% in LEA F1 score. Moreover, further analysis reveals that our framework better handles coreference resolution for long mentions and the ones that are located far beyond one sentence. To sum up, this paper contributes mainly in three ways.

- We propose a novel syntactic and discourse graph model that integrates syntactic and discourse clues to capture the complex relationships and rich structural information among different elements in the screenplay.
- We employ multiple mechanisms to model different key factors in screenplay character coreference resolution such as triaffine scoring, singleton detection, and span filtering to harness more valuable features and reduce noise.
- Our extensive experimental results on the MovieCoref dataset demonstrate that STPar achieves state-of-the-art (SOTA) performance and outperforms the best baseline with large margins.

2 Related Work

2.1 Coreference Resolution

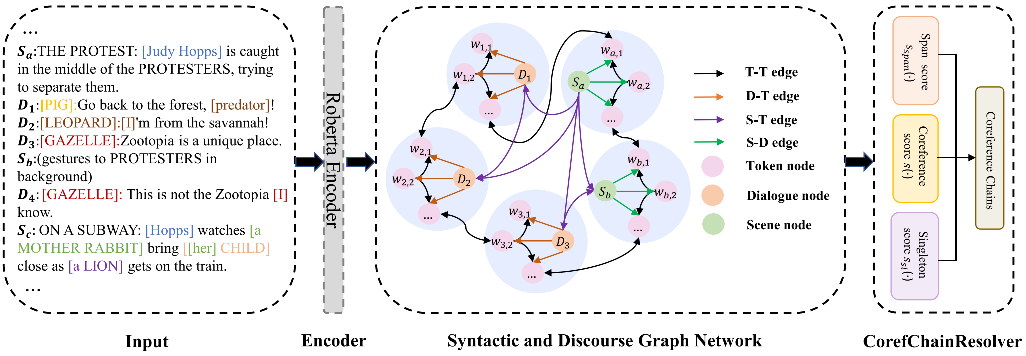
Coreference resolution has long been a fundamental NLP task (Lu and Ng, 2018; Kong and Fu, 2019), aiming at identifying mentions of the same entity. Early works primarily focused on syntactic features for coreference resolution. Ge et al. (1998) propose the Hobbs distance for ranking candidate antecedents of a given pronoun based on Hobbs' pronoun resolution algorithm utilizing syntactic parse trees (Hobbs, 1978). Kong and Zhou (2011) introduce various path-related features, such as the root path between the root node and the current reference. Durrett et al. (2013) integrate entity-level information for coreference resolution. Lee et al. (2018) introduce a fully differentiable approximation method for addressing higher-order inference in coreference problems.

With the advancements in deep learning, existing traditional coreference resolution tasks are primarily divided into sequence-based (Zhang et al., 2023b; Wu et al., 2020) and graph-based (Jiang and Cohn, 2022, 2021) approaches. In terms of sequence-based methods, Zhang et al. (2023b) employ a pre-trained seq2seq transformer model and fine-tune it to map an input document to a tagged sequence. Wu et al. (2020) apply a question-answering framework to the task. On the other hand, graph-based methods (Jiang and Cohn, 2022) investigate the use of constituent syntax in neural coreference models. Jiang and Cohn (2021) propose a heterogeneous graph to solve the coreference resolution task.

Recently, character coreference resolution in movie screenplays has been proposed to help audiences understand the movie plot and disentangle the intricate character relationships. Baruah et al. (2021) introduce a coreference annotation guideline for movie screenplays and devise rules based on the structure of the screenplay to enhance performance. Baruah and Narayanan (2023) propose a screenplay character coreference dataset and apply a word-level model to address this task. However, existing methods overlook in-depth analysis of critical screenplay structures, lack discrimination and filtering of different clue information.

2.2 Discourse Dependency Parsing

Discourse dependency parsing has demonstrated its effectiveness in various discourse understanding tasks, such as dialog sentiment recognition (Zhang et al., 2023a) and multi-party dialogues (Wang et al., 2021). Existing research (Jiang and Cohn, 2022; Meng et al., 2023) indicates that syntactic dependency is a commonly useful feature in several coreference systems, while the utility of discourse relations is more intricate and has not been thoroughly explored. Recently, various discourse parsers have been proposed. Liu and Chen (2021) design a transformer-based discourse parser, DDP, which is characterized by 16 types of discourse relations. Ko et al. (2023) treat each sentence as an answer to the questions triggered by the preceding context and adopt a linguistic framework for discourse analysis. Their parser, QUD, is based on 10 types of discourse relations, which are designed to capture certain semantic and syntactic connections within the text. These parsers provide explicit discourse structures for downstream applications. In this paper, we choose to utilize QUD to incorporate discourse



…
S_a:THE PROTEST: [Judy Hopps] is caught in the middle of the PROTESTERS, trying to separate them.
D_1:[PIG]:Go back to the forest, [predator]!
D_2:[LEOPARD]:[I]'m from the savannah!
D_3:[GAZELLE]:Zootopia is a unique place.
S_b:(gestures to PROTESTERS in background)
D_4:[GAZELLE]: This is not the Zootopia [I] know.
S_c: ON A SUBWAY: [Hopps] watches [a MOTHER RABBIT] bring [[her] CHILD] close as [a LION] gets on the train.
…

Figure 2: The overall architecture of our proposed model.

structure information into STPar model. The purpose is to enhance the model's ability to perceive the close interactions among various elements in the screenplay.

3 Methodology

In this paper, we propose a Structure-aware Triaffine Parser (STPar) to address the MovieCoref task, whose architecture is depicted in Figure 2.

Task Definition. Following Lee et al. (2017), we formulate the MovieCoref task as a set of antecedent assignments y_i for each word w_i. Our goal is to learn a probability distribution P over all possible antecedent words Y_{w_i} for each word w_i:

$$P(y_{w_i}) = \frac{e^{s(w_i, y_{w_i})}}{\sum_{y' \in Y_{w_i}} e^{s(w_i, y')}} \tag{1}$$

where $s(w_i, y_{w_i})$ is the coreference score between words w_i and y_{w_i}. The set $Y_{w_i} = \{w_1, \ldots, w_{i-1}\}$ represents the potential antecedents for w_i.

3.1 Encoder

In accordance with the approach described in Baruah and Narayanan (2023), we adopt RoBERTa (Liu et al., 2019) to encode each segment of the screenplay text S and then combine them to recover the screenplay representation that comprises a sequence of n words. The same approach as Baruah and Narayanan (2023) is adopted for screenplay segmentation to divide long screenplays into segments, which allows us to effectively handle lengthy screenplays. The encoding process can be formulated briefly as:

$$\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\} = RoBERTa\left(\{w_1, \ldots, w_n\}\right) \tag{2}$$

where $\boldsymbol{v}_n$ is the output representation for the word w_n.

3.2 Syntactic and Discourse Graph Network

Due to the rich elements and complex hierarchical structure of the screenplay, we design a syntactic and discourse graph network to represent different types of nodes and their relationships, aiming to capture the global multi-level structural information and dependencies.

Node Construction. The graph consists of three different types of nodes: token nodes (T), dialogue nodes (D), and scene nodes (S). The representation of token nodes is contextual embeddings from the RoBERTa encoder. Dialogue nodes and scene nodes are initialized by averaging the embeddings of the tokens they encompass.

Edge Construction. Based on the feature structures, we establish four distinct types of edges to represent the rich dependency relationships between nodes.

- **Token-Token** Edges are constructed based on the syntactic dependency trees that are produced by the Stanford CoreNLP parser.[2] Specifically, each syntactic dependency edge connects a head word with a dependent word that has a specific syntactic label. Self-loop edges with loop labels are also added to each node in the graph. Additionally, we connect the root nodes of two adjacent sentences to allow for inter-sentence interactions.
- **Scene-Dialogue** Edges are constructed based on the discourse dependency structure of screenplays. Specifically, we employ 16 discourse dependency types derived from a pre-trained discourse parser (Liu and Chen, 2021) to predict discourse dependencies in

[2] https://stanfordnlp.github.io/CoreNLP/.

the screenplay. This facilitates a better understanding of the relationships between sentences and the integration of screenplay structure information.

- **Dialogue-Token** Edges connect utterances with their corresponding text tokens, enabling the transfer and aggregation of features and leading to a more comprehensive understanding of contextual information.
- **Scene-Token** Edges link the scene descriptions in the screenplay with their corresponding text, incorporating scene information to enhance tokens' semantic representation.

Graph Attention Layer. After learning contextualized word representations from the screenplay, we propose a novel graph attention network to model the relationships between elements, aiming to capture the structural information and dependencies at different levels of the screenplay. Specifically, the graph attention network propagates information between different nodes by stacking multiple graph attention layers. In each layer, the network learns updated node representations by aggregating information from neighboring nodes using self-attention. The graph attention mechanism operates on each node through the following aggregation scheme:

$$\boldsymbol{h}_i^{(t)} = ReLU(\sum_{k\in\kappa}\sum_{j\in\mathcal{N}_k(i)} (\alpha_{ij}^{(t)}\boldsymbol{W}_k^{(t-1)}\boldsymbol{h}_j^{(t-1)} + \boldsymbol{b}_k^{(t-1)})) \quad (3)$$

where $\boldsymbol{h}_i^{(t)}$ is the hidden representation of the word w_i in the t-th layer,[3] $\boldsymbol{W}_k^{(t-1)}$ and $\boldsymbol{b}_k^{(t-1)}$ are learnable parameters, κ are different types of edges, and $\mathcal{N}_k(i)$ denotes the neighbors of the node i connected with the k^{th} type of edge. The attention weight $\alpha_{ij}^{(t)}$ reflects the strength of aggregation level between nodes and is learned through an MLP parameterized by $\boldsymbol{\omega}^{(t)}$:

$$e_{ij}^{(t)} = \boldsymbol{\omega}^{(t)\top} tanh([\boldsymbol{W}^{(t)}\boldsymbol{h}_i^{(t-1)}; \boldsymbol{W}^{(t)}\boldsymbol{h}_j^{(t-1)}]) \quad (4)$$

$$\alpha_{ij}^{(t)} = \frac{exp(LeakyReLU(e_{ij}^{(t)}))}{\sum_{k\in\mathcal{N}(i)} exp(LeakyReLU(e_{ik}^{(t)}))} \quad (5)$$

where $[\cdot;\cdot]$ is concatenation. By stacking T layers, we obtain enhanced node representations that capture both syntactic and discourse information. The output of the final layer is the updated word

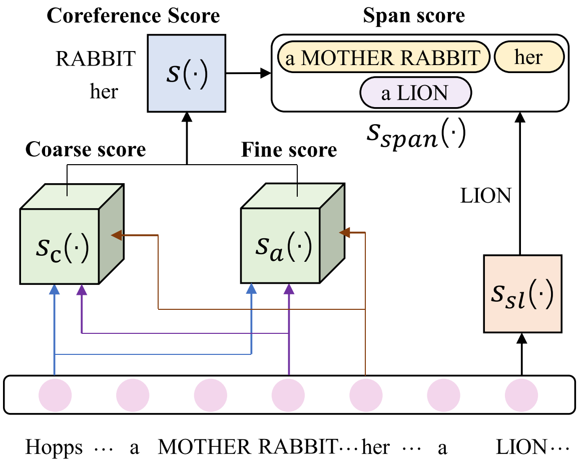


Figure 3: Detailed process of CorefChainResolver, where 'her' is the target word, 'a MOTHER RABBIT' is the candidate antecedent, 'Hopps' is the third part for triaffine computation.

representations $\boldsymbol{H} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n\}$. By utilizing a graph attention network composed of multiple graph attention layers, we effectively model the interactions between nodes in the screenplay. This network is capable of capturing the relationships between nodes by adaptively integrating information from other nodes. By doing so, we generate expressive node representations that allow for a thorough exploration of the inter-node relationships present in the screenplay.

3.3 CorefChainResolver: Multi-Task Learning for Coreference Prediction

We present a CorefChainResolver module for multitask learning and inference, as depicted in Figure 3. This module effectively handles singleton detection and span detection auxiliary tasks while incorporating the triaffine scorer for high-order scoring to enhance the performance of the MovieCoref task.

Singleton Detection. We design a singleton predictor to distinguish singleton mentions from non-singleton mentions. The singleton score $s_{sl}(w_i)$ of a word w_i denotes the likelihood of it being the head word of a singleton mention:

$$s_{sl}(w_i) = FFNN_{sl}(\boldsymbol{h}_i) \quad (6)$$

where $\boldsymbol{h}_i$ is the representation of the word w_i obtained from Eq. 3. By integrating this additional task into the character coreference resolution framework, our approach strengthens the ability to accurately identify and differentiate singleton

[3] The input $\boldsymbol{h}_i^{(0)}$ of the first layer is $\boldsymbol{v}_i$, obtained from the encoder (cf. Eq. 2).

mentions. In turn, this contributes to improving the overall performance of character coreference resolution by addressing the challenges posed by singleton characters in screenplays.

Coarse Coreference Scores. For the coreference relation determination between the target word w_i and its antecedent w_j, we follow previous work (Dobrovolskii, 2021) to compute both coarse and fine coreference scores. The intuition is to balance computational efficiency and accuracy. Coarse scores assist in narrowing down the range of candidate antecedents, reducing computational complexity. Meanwhile, fine scores introduce additional features to assess the relevance between tokens at a more granular level. Moreover, we design a triaffine scorer (Carreras, 2007) to consider the third part w_k as a supplementary clue for disambiguation in coreference resolution. Concretely, the head word w_k belongs to the mention k that is adjacent to the left side of the mention i or j, where mention i or j contains the word w_i or w_j. Therefore, the coarse coreference score (Dobrovolskii, 2021) calculated via Biaffine (Dozat and Manning, 2017) can be extended via triaffine as below:

$$\begin{aligned} s_c(w_i, w_j, w_k) = \boldsymbol{h}_k \cdot \boldsymbol{h}_i \cdot \boldsymbol{W}_c \cdot \boldsymbol{h}_j^\top \\ + s_m(w_i) + s_m(w_j) + s_m(w_k) \end{aligned} \tag{7}$$

where $\boldsymbol{h}_i$, $\boldsymbol{h}_j$, $\boldsymbol{h}_k$ are the representations of the target word, antecedent and third part, and $\boldsymbol{W}_c$ is a learnable parameter. Following Dobrovolskii (2021), we employ a feed-forward neural network to calculate the mention candidate score $s_m(w_i)$ to evaluate how likely w_i belongs to a mention.

Fine Coreference Scores. Following Dobrovolskii (2021), we calculate the fine-grained coreference scores as below:

$$s_a(w_i, w_j, w_k) = FFNN_a([\boldsymbol{h}_k, \boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k \odot \boldsymbol{h}_i \odot \boldsymbol{h}_j, \phi]) \tag{8}$$

where ϕ is the concatenation of features such as the distance between w_i and w_j and whether they are spoken by the same character. By incorporating distance and speaker information features into the ϕ, our model can gain additional contextual information. This enhances the model's ability to capture subtle relationships and improves the accuracy of MovieCoref. The word representations and their concatenation and element-wise products ($\odot$) are also used. Concatenation allows for a comprehensive representation of individual and combined information, while element-wise product emphasizes shared features and interactions. This combination approach provides a more holistic representation of token embeddings, enhancing the model's understanding of contextual relationships. All above features are fed into a feed-forward neural network.

Overall Coreference Scores. The overall coreference scores are defined as the sum of two scores:

$$s(w_i, w_j, w_k) = s_c(w_i, w_j, w_k) + s_a(w_i, w_j, w_k) \tag{9}$$

The candidate antecedent w_j with the highest score is considered as the true antecedent for the word w_i. If $s(w_i, w_j, w_k)$ is negative for all candidates, the word w_i is considered to have no antecedents.

Span Detection. For each word w_i determined to have coreferent relationships with other words, we reconstruct its span by expanding its start and end boundary words w_p and w_q. The range of the start boundary word can be w_1 to w_i, and the end boundary word can be w_i to w_n. The span detection scorer, SpanPredictor, consists of a feed-forward neural network and a convolutional block.

$$s_{span}(w_i) = SpanPredictor(\boldsymbol{h}_i, \boldsymbol{h}_p, \boldsymbol{h}_q) \tag{10}$$

where $\boldsymbol{h}_i$, $\boldsymbol{h}_p$, and $\boldsymbol{h}_q$ are the word representations. The span with the highest score is used as the mention for the word w_i.

3.4 Training

Following Dobrovolskii (2021), we train the coarse- and fine-grained coreference scorers by utilizing the negative log marginal likelihood ($\mathcal{L}_{MLL}$) as the primary loss function. Additionally, we incorporate a binary cross-entropy loss ($\mathcal{L}BCE$) as an additional regularization factor:

$$\mathcal{L}_{MLL} = -log \prod_{i=1}^{n} \sum_{\hat{y} \in Y_i \cap G_i} P(\hat{y}) \tag{11}$$

$$\mathcal{L}_{coref} = \mathcal{L}_{MLL} + \alpha \mathcal{L}_{BCE} \tag{12}$$

where G_i denotes the set of words in the gold cluster containing word w_i, Y_i denotes the set of words to the left of w_i, α is the hyper-parameter to control the weight of the BCE loss and we set it as 0.5 following prior work. We train the singleton detection module and span detection module using

	MUC			B^3			CEAF$_e$			LEA			CoNLL
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	F1
Rule-based	89.3	79.2	83.7	54.8	63.3	57.5	61.5	40.9	47.9	63.2	54.7	57.3	63.1
Hierarchical-based	94.2	92.4	93.2	70.6	78.8	73.8	39.5	60.4	46.2	70.3	78.6	73.5	71.1
Fusion-based	93.9	92.8	93.3	81.3	69.0	74.5	34.7	**62.9**	43.3	**81.0**	68.7	74.2	70.4
STPar (ours)	**96.1**	**93.9**	**95.0**	**81.6**	**82.2**	**81.9**	**75.7**	56.7	**64.8**	80.9	**81.9**	**81.3**	**80.6**
	(+1.9%)	(+1.1%)	(+1.7%)	(+0.3%)	(+3.4%)	(+7.4%)	(+14.2%)	(−6.2%)	(+16.9%)	(−0.1%)	(+3.3%)	(+7.1%)	(+9.5%)

Table 1: Results on the MovieCoref dataset. The numbers in parentheses are the improvements of our model over the best-performing baseline(s).

the cross-entropy loss, denoted as $\mathcal{L}_{sl}$ and $\mathcal{L}_{span}$, respectively. Finally, the total loss function is the summation of all the aforementioned losses:

$$\mathcal{L} = \mathcal{L}_{coref} + \mathcal{L}_{sl} + \mathcal{L}_{span} \tag{13}$$

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate our model on the standard MovieCoref dataset (Baruah and Narayanan, 2023). The dataset consists of six full-length movie screenplays and excerpts from three movie screenplays. The average document length of the full-length movie screenplays is approximately 30,000 words. MovieCoref contains 418 characters involved in 25,793 mentions across a total of 201,804 words.

Evaluation Metrics. In terms of evaluation metrics, we align with the MovieCoref (Baruah and Narayanan, 2023), and utilize CoNLL F1, which is the average F1 score of three metrics: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF$_e$ (Luo, 2005), as well as LEA F1 score (Moosavi and Strube, 2016).

Hyperparameter Settings. We train our model by setting the epoch, dropout, GCN layers and batch size to 50, 0.2, 2, and 64, respectively. The learning rate is set to 2e-5 for RoBERTa and 2e-4 for other modules. L2 regularization is applied with a decay rate of 1e-3. We retain the top 50 antecedent candidates. All scores are averaged over five runs with different random seeds.

4.2 Baseline Systems

To validate the effectiveness of our model, we compare it against the following SOTA baselines:

- **Rule-based:** Baruah et al. (2021) combined neural models with structural rules to adapt to the MovieCoref task.
- **Fusion-based:** Baruah and Narayanan (2023) partitioned the screenplay into overlapping sub-documents and performing inference separately on each sub-document.
- **Hierarchical-based:** Baruah and Narayanan (2023) divided the screenplay into non-overlapping sub-documents and merged them for coreference clustering.

Notably, all baselines do not explicitly model the multi-level structure of the screenplay and capture the high-order dependency relationships.

4.3 Overall Results

The experimental results for the MovieCoref task are presented in Table 1, revealing several key findings. First of all, compared to the best baseline Fusion-based model, our model achieves a substantial improvement of 7.1% in LEA F1 and 10.2% in CoNLL F1. Moreover, when compared to the Rule-based model, the performance gains are even more substantial, enhancing LEA F1 by 24.0% and CoNLL F1 by 17.5%. These results clearly demonstrate the substantial superiority of STPar over other strong baselines. Secondly, in terms of the fine-grained metrics, STPar surpasses the baselines across almost all categories, including MUC, B^3, and CEAF$_e$, thereby confirming the robustness of our model. Moreover, the substantial improvements in our model's performance are particularly evident in the CEAF$_e$ F1 and B^3 F1 scores, where it records increases of 21.5% and 7.4% respectively compared to the best baseline. This can be attributed to our model's accurate identification and grouping of coreferent mentions at the entity and boundary levels. Overall, the performance highlights the efficacy of STPar in leveraging syntactic and discourse features, as well as capturing high-order dependency relations.

	MUC	B^3	$CEAF_e$	LEA	CoNLL
STPar (ours)	95.0	81.9	64.8	81.3	80.6
w/o Graph	93.1(−1.9)	76.2(−5.7)	57.8(−7.0)	76.7(−4.6)	75.7(−4.9)
w/o T-T edge	93.9(−1.1)	78.4(−3.5)	60.8(−4.0)	78.7(−2.6)	77.7(−2.9)
w/o D-T edge	94.1(−0.9)	79.4(−2.5)	61.5(−3.3)	79.3(−2.0)	78.3(−2.3)
w/o S-T edge	94.0(−1.0)	79.1(−2.8)	61.2(−3.6)	79.0(−2.3)	78.1(−2.5)
w/o S-D edge	93.5(−1.5)	78.1(−3.8)	60.3(−4.5)	78.2(−3.1)	77.3(−3.3)
w/o Triaffine	93.8(−1.2)	78.2(−3.7)	60.5(−4.3)	78.4(−2.9)	77.5(−3.1)
w/o Singleton	94.1(−0.9)	78.9(−3.0)	61.1(−3.7)	79.2(−2.1)	78.0(−2.6)

Table 2: Ablation results on MovieCoref. The numbers in parentheses are the decreased values compared with our full model.

4.4 Ablation Study

We conduct ablation experiments to assess the contribution of each component in our model. Our ablation experiments involve removing a particular component from the complete model, retraining the model on the modified architecture without that component, and then comparing the performance of the retrained model with that of the original model. The results in Table 2 show that no variants can compete with the complete model, implying the indispensability of each component for MovieCoref. Specifically, the absence of the syntactic and discourse graphs led to the most significant performance degradation, with a decrease of 4.6% in LEA F1 and 4.9% in CoNLL F1. This indicates its significant impact on modeling the complex structural information of screenplays. To validate the necessity and effectiveness of each node and edge plays in the syntactic and discourse graphs, we individually removed them. The sharp decrease in results demonstrates the pivotal role that nodes and edges in capturing speaker, discourse, scene text, and their interactions. Additionally, the decrease in results when removing the triaffine scorer highlights the crucial impact of high-order scoring on the MovieCoref task. Furthermore, the removal of singleton detection leads to a performance drop, indicating that enhanced identification of singletons contributes to improving model's performance.

4.5 Analyses and Discussion

To further investigate the effectiveness of STPar, we conduct in-depth analyses to answer the questions (in bold) in the following sections, with the aim to reveal how our proposed methods advance.

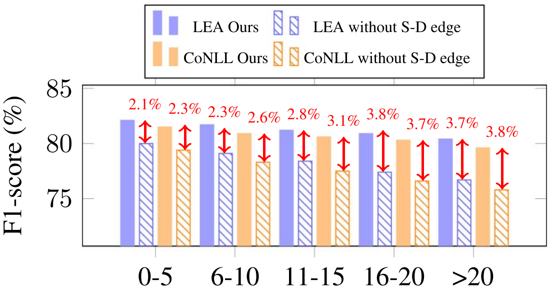


Figure 4: Performance comparisons with and without discourse dependencies in different mention distances.

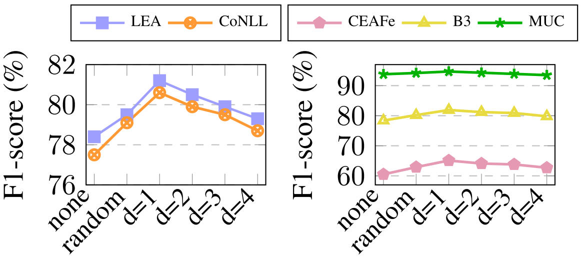


Figure 5: Results of different third word selection methods.

1) Does discourse information play an important role in coreference resolution if context and mention distance become long? We investigate the impact of discourse structure parsing on the performance of coreference at different mention distances in MovieCoref. In Figure 4, we observe that regardless of the distance between mentions, the inclusion of discourse structure parsing consistently outperforms the model without it, indicating the effectiveness of discourse structure parsing for the MovieCoref task. Especially when the distance between mentions increases, the performance gap between the models with and without discourse structure parsing becomes more significant. Specifically, when the distance exceeds 20, the performance decline without discourse parsing amounts to 3.7% in LEA F1 and 3.8% in CoNLL F1. This highlights the crucial role of discourse structure parsing in bridging larger gaps between mentions and facilitating accurate coreference resolution across longer distances.

2) Does the method of selecting the third word in triaffine influence the performance? We are curious about the impact of different triplet combinations on model performance in high-order scoring. In Figure 5, we employ various strategies for selecting the third word, including no third word, random selection, and selecting a word at

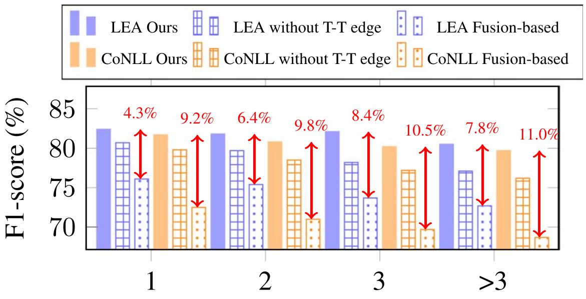


Figure 6: Performance comparisons with and without syntactic dependencies across different mention lengths.

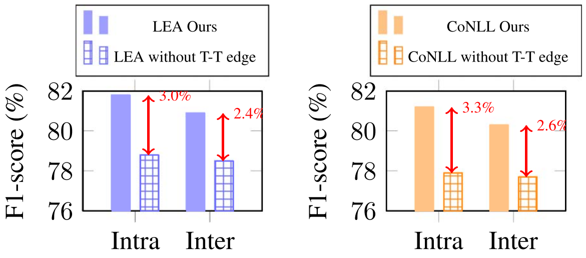
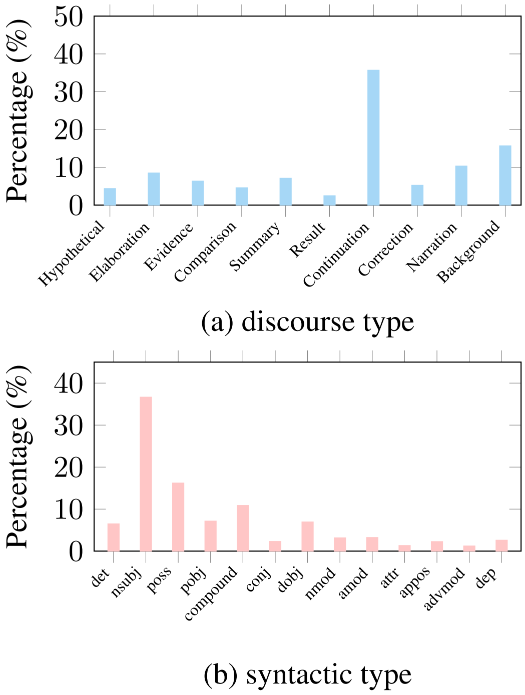


Figure 7: Performance comparisons with and without syntactic information for intra-sentence and inter-sentence coreference resolution.

different distances from the mention. We observe that the performance is poorest when using only the binary features, highlighting the necessity of utilizing the triaffine scorer for high-order scoring. Additionally, we find that the performance of randomly selecting the third word is significantly lower than that of specifying its position, with closer distances yielding better results. This suggests that the relationship between the mention and neighboring words is crucial for accurate coreference resolution.

3) What is the role of syntactic information in performance improvement? Figure 6 displays the performance comparison of our complete model, our model without syntactic information, and the best baseline model across different mention lengths. The results demonstrate the consistent superiority of our complete model over both the baseline and the model without syntactic information. Notably, our complete model exhibits a more substantial performance improvement for longer mentions. This may be because syntactic information can help singleton detection and span detection in long mentions that entail syntactic structures, thereby boosting the overall performance.

On the other hand, we investigate the role of syntactic information in context understanding and representation. Concretely, we compare the models' performances with and without syntactic dependencies for coreference resolution of both intra-sentence and inter-sentence mentions. As shown in Figure 7, incorporating syntactic information has a positive influence on the MovieCoref task in both cases. The effect of syntactic information on the MovieCoref task is particularly significant for intra-sentence mentions. This is because syntactic information directly capture the grammatical dependencies between mentions and their antecedents, providing valuable clues and contextual information. Additionally, syntactic information provides assistance in resolving coreference for inter-sentence mentions. When mentions and antecedents are located in different sentences, the utilization of syntactic information offers additional clues to aid the model in determining the correct coreferent relationships.



(a) discourse type

(b) syntactic type

Figure 8: Distribution of syntactic and discourse dependency types with regards to coreference mentions.

4) What dependency types of syntactic and discourse are more strongly associated with the MovieCoref task? We first analyze the discourse dependency types associated with coreferent mentions identified by our model. Concretely, we count the discourse dependency types occurred between two sentences with coreferent mentions in Figure 8(a). We observe a clear correlation between certain discourse dependency types and sentences. For instance, the presence of a ‘‘continuation’’ dependency type strongly suggests coreference between the mentions in those sentences. Conversely, ‘‘Result’’ or ‘‘Hypothetical’’ dependency types indicate a less prominent coreferent relationship between the mentions. This emphasizes the value of discourse dependency information in resolving character coreference and enhancing our understanding of the screenplay’s multi-level structure.

Next, we delve into the analysis of syntactic dependency types, exploring the most relevant syntactic dependency types for coreferent mentions within the same sentence. As illustrated in Figure 8(b), we observe that when the dependency types are ‘‘nsubj’’, ‘‘poss’’, and ‘‘compound’’, there is a higher likelihood of a coreferent relationship between the two words. STPar demonstrates a stronger ability to establish coreferent relationships for such cases. On the other hand, for words with dependency types such as ‘‘attr’’ and ‘‘advmod’’, the existence of coreferent relationships can be ignored. In summary, these analysis highlight the successful learning of structural correlations within and between sentences by STPar.

5) How does STPar generalize to traditional coreference datasets? To validate the generalization ability of our approach, we compare STPar with the state-of-the-art traditional coreference resolution model, wl-coref (Dobrovolskii, 2021), CorefQA (Wu et al., 2020), Link-Append (Bohnet et al., 2023) and Maverick-mes (Martinelli et al., 2024) on two traditional coreference resolution datasets, Litbank (Bamman, 2020) and OntoNotes (Pradhan et al., 2012). The experimental results, as shown in Table 3, clearly indicate that STPar outperforms the current SoTA models on both datasets.

6) How is the efficiency of our model? To evaluate the efficiency of our model, we conduct experiments with and without the Graph and Triaffine. The results are presented in Table 4. We observe significant improvements when utilizing Graph and Triaffine. Specifically, LEA F1 increased by 7.2%, ConLL F1 increased by 7.4%, and CEAF$_e$ F1 increased by 11.4%. Furthermore, the inference speed between the two methods remain relatively consistent, indicating that the use of Graph and Triaffine does not have a significant impact on the model’s inference speed. These comprehensive findings emphasize the advantage of incorporating Graph and Triaffine to enhance the accuracy of screenplay coreference resolution without compromising the efficiency of the inference process.

Method(dataset)	CoNLL	MUC	B^3	$CEAF_e$
wl-coref (litbank)	79.7	84.1	81.5	73.6
CorefQA (litbank)	79.9	84.4	81.7	73.7
Link-Append (litbank)	80.2	85.0	81.9	73.8
Maverick-mes (litbank)	79.5	84.0	81.2	73.3
STPar (litbank)	**81.0**	**85.9**	**82.6**	**74.5**
wl-coref (ontonotes)	82.9	87.2	82.0	79.4
CorefQA (ontonotes)	83.1	88.0	82.2	79.1
Link-Append (ontonotes)	83.3	87.8	82.6	79.5
Maverick-mes (ontonotes)	83.4	88.1	82.7	79.3
STPar (ontonotes)	**83.9**	**88.8**	**83.5**	**79.6**

Table 3: The results on Litbank and OntoNotes dataset.

	LEA	CoNLL	MUC	B^3	$CEAF_e$	speed(s)
w/o Graph and Triaffine	74.1	73.2	91.6	74.5	53.4	224
STPar	81.3	80.6	95.0	81.9	64.8	240

Table 4: Comparison of results and efficiency between using Graph and Triaffine and not using them.

	LEA	CoNLL	MUC	B^3	$CEAF_e$
• ***RoBERTa(125M)***					
Fusion-based	74.2	70.4	93.3	74.5	43.3
STPar	81.3	80.6	95.0	81.9	64.8
• ***Flan-T5(11B)***					
Fusion-based	80.5	79.7	94.3	81.0	63.9
STPar	87.8	86.9	97.5	89.7	73.5

Table 5: Experimental results of large generative language models in screenplay coreference resolution task.

7) How do large generative language models perform in screenplay coreference resolution task? To investigate the applicability of large language models in MovieCoref, we conduct experiments using the flan-t5-11B in combination with STPar. We compare the experimental results with the SOTA model, and the specific findings are presented in Table 5. It is evident that STPar outperforms fusion-based models by a significant margin, regardless of whether a large or small language model is utilized. This indicates the impressive performance of STPar in modeling complex screenplay structures and semantic understanding. Additionally, we discover

	LEA	CoNLL	MUC	B3	CEAF$_e$
DDP (16)	80.9	80.1	94.5	81.4	64.4
DDP (2)	79.6	79.1	93.9	80.3	63.1
QUD (10)	81.3	80.6	95.0	81.9	64.8
None	78.2	77.3	93.5	78.1	60.3

Table 6: Experimental results on different discourse parsers and types of discourse relations.

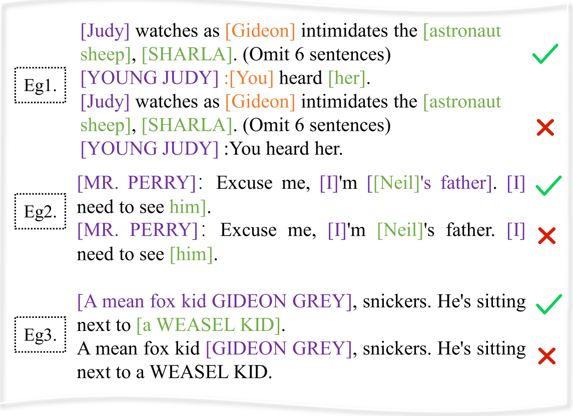
	MUC	B^3	CEAF$_e$	LEA	CoNLL
wl-coref	93.3	74.5	43.3	74.2	70.4
Maverick-mes	93.9	78.9	56.8	78.9	76.5
STPar (ours)	95.0	81.9	64.8	81.3	80.6

Table 7: The results on MovieCoref dataset.

that incorporating LLMs further improves performance, affirming the suitability and effectiveness of LLMs in MovieCoref.

8) How do different discourse parsers and types of discourse relations influence coreference resolution? To investigate the impact of discourse information on screenplay coreference resolution, we conduct experiments using two discourse parsers with differing discourse theories and relation types. DDP (Liu and Chen, 2021) comprises 16 discourse relation types, while QUD (Ko et al., 2023) encompasses 10 discourse relation types. As shown in Table 6, our experimental results consistently demonstrate the superiority of using discourse parsers over not using them, underscoring the necessity of analyzing the discourse structure of scripts. Figure 8(a) analyzes which discourse relations are most beneficial for coreference resolution. To discern the truly helpful information, we reduce the discourse relation types to only two categories: continuation and non-continuation. We observe a decrease in performance, indicating the need to analyze the fine-grained relations between script discourses to aid coreference resolution.

9) How does the performance of traditional coreference resolution methods fare in MovieCoref? To explore the performance of traditional coreference resolution methods in MovieCoref, we transfer the current SOTA models in traditional resolution tasks to our MovieCoref task. The results are shown in Table 7. We find that Maverick-mes (Martinelli et al., 2024), due to its design of probability-based calculation of mention start and end positions, which helps to more accurately identify character mentions in the screenplay, outperforms wl-coref (Dobrovolskii, 2021). However, it is slightly inferior compared to our method, which is specifically designed for the screenplay structure. This further indicates that screenplays, different from ordinary documents and dialogues, have a unique structure, and how to clarify the structural clues of the screenplay is of crucial importance.



Eg1.

[Judy] watches as [Gideon] intimidates the [astronaut sheep], [SHARLA]. (Omit 6 sentences) [YOUNG JUDY] :[You] heard [her]. ✓

[Judy] watches as [Gideon] intimidates the [astronaut sheep], [SHARLA]. (Omit 6 sentences) [YOUNG JUDY] :You heard her. ✗

Eg2.

[MR. PERRY]: Excuse me, [I]'m [[Neil]'s father]. [I] need to see [him]. ✓

[MR. PERRY]: Excuse me, [I]'m [Neil]'s father. [I] need to see [him]. ✗

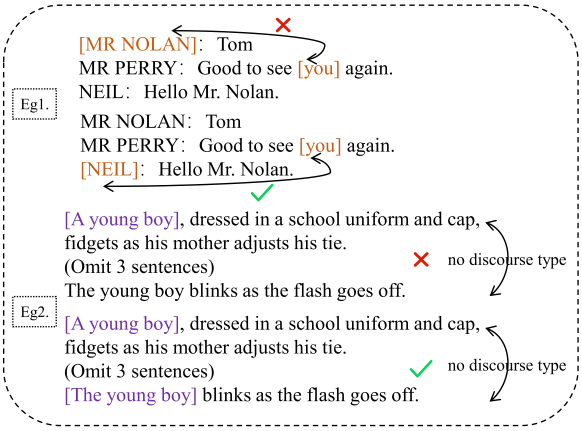
Eg3.

[A mean fox kid GIDEON GREY], snickers. He's sitting next to [a WEASEL KID]. ✓

A mean fox kid [GIDEON GREY], snickers. He's sitting next to a WEASEL KID. ✗

Figure 9: Three examples of case studies on STPar (Upper) and Fusion-based method (Lower). Like-colored mentions are co-referring in every example.

10) Case Study To gain a deeper understanding of our model's capabilities, we conduct a comprehensive case study on the MovieCoref task. As shown in Figure 9, our full model successfully performs coreference resolution. In contrast, the Fusion-based model struggles in cases involving singletons, long mentions, and large mention distances. For instance, in *Eg1* where mention distances are large, the Fusion-based model faces challenges in establishing correct links between ‘‘Gideon’’ and ‘‘you’’ and between ‘‘SHARLA’’ and ‘‘her’’. This highlights the limitations of the Fusion-based method in capturing long-range dependencies. Similarly, in *Eg2* and *Eg3*, both containing long mentions, the Fusion-based method fails to identify the correct coreferences, illustrating its struggle in handling complex and lengthy references. Furthermore, in *Eg3*, the Fusion-based method fails to recognize the singleton mention ‘‘a WEASEL KID’’, indicating its difficulty in handling isolated mentions. Overall, this analysis emphasizes the critical role of our model in effectively leveraging



[MR NOLAN]: Tom
MR PERRY: Good to see [you] again.
NEIL: Hello Mr. Nolan.

MR NOLAN: Tom
MR PERRY: Good to see [you] again.
[NEIL]: Hello Mr. Nolan.

[A young boy], dressed in a school uniform and cap,
fidgets as his mother adjusts his tie.
(Omit 3 sentences)
The young boy blinks as the flash goes off.

[A young boy], dressed in a school uniform and cap,
fidgets as his mother adjusts his tie.
(Omit 3 sentences)
[The young boy] blinks as the flash goes off.

Figure 10: Two examples of error analysis on STPar method (Upper) compared with gold (Lower). Like-colored mentions are co-referring.

the multi-level screenplay structure and capturing high-order dependencies.

11) Error Analysis In Figure 10, we conduct an error analysis on STPar to gain insights for future work. We analyze 100 examples and identify two types of errors, including pronoun referencing errors (58%) and discourse relation recognition errors (42%). The first error, as shown in *Eg1*, occurs in dialogues with multiple roles where pronoun references are ambiguous, leading to incorrect identification of coreference relationships. This ambiguity arises from the presence of multiple potential referents among the dialogue participants. Future research can design models that better understand context and semantic information. The second error, as illustrated in *Eg2*, involves the failure to recognize coreference relationships between mentions in sentences with unrecognized dependency relationships. STPar relies on the output of the discourse parser, and errors may occur when the parser fails to correctly parse sentence relationships or the relationships between sentences are ambiguous. Future work can focus on enhancing discourse parsing techniques to accurately capture the dependencies and relationships between sentences.

5 Conclusion

In this paper, we propose a novel model, STPar, for character coreference resolution in movie screenplays. Our model has mainly addressed three problems. First, it leverages discourse and syntactic information via graph attention network to model the complex screenplay structure. Second, it introduces the triaffine mechanism instead of biaffine to consider high-order information which is crucial for coreference resolution. Third, it simultaneously performs coreference scoring, singleton detection and span detection for mutual benefit of multiple tasks. Through experiment evaluation, we have found that all of our claimed innovative approaches and hypotheses have been demonstrated to be effective. Our work provides an insightful view to check the effect of discourse and syntactic information in MovieCoref and a promising model to boost the line of this task.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2022YFB3103602), the National Natural Science Foundation of China (No. 62176187, No. 62202210). This work is also supported by the fund of Laboratory for Advanced Computing and Intelligence Engineering. This work is also supported by the open project of Sichuan Provincial Key Laboratory of Philosophy and Social Science for Language Intelligence in Special Education (No. YYZN-2023-1).

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

David Bamman. 2020. Litbank: Born-literary natural language processing. *Computational Humanites, Debates in Digital Humanities (2020, preprint).*

Sabyasachee Baruah, Sandeep Nallan Chakravarthula, and Shrikanth Narayanan. 2021. Annotation and evaluation of coreference resolution in screenplays. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 2004–2010. https://doi.org/10.18653/v1/2021.findings-acl.176

Sabyasachee Baruah and Shrikanth Narayanan. 2023. Character coreference resolution in movie screenplays. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10300–10313. https://doi.org/10.18653/v1/2023.findings-acl.654

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226. https://doi.org/10.1162/tacl_a_00543

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 957–961.

Zeyu Dai, Hongliang Fei, and Ping Li. 2019. Coreference aware representation learning for neural named entity recognition. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 4946–4953. https://doi.org/10.24963/ijcai.2019/687

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 7670–7675. https://doi.org/10.18653/v1/2021.emnlp-main.605

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.

Greg Durrett, David Hall, and Dan Klein. 2013. Decentralized entity-level modeling for coreference resolution. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 114–124.

Yuxuan Feng, Qian Chen, Qianyou Wu, Xin Guo, and Suge Wang. 2025. Sure: Mutually visible objects and self-generated candidate labels for relation extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 459–468.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Sixth Workshop on Very Large Corpora*.

Jerry R. Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338. https://doi.org/10.1016/0024-3841(78)90006-2

Fan Jiang and Trevor Cohn. 2021. Incorporating syntax and semantics in coreference resolution with heterogeneous graph attention network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 1584–1591. https://doi.org/10.18653/v1/2021.naacl-main.125

Fan Jiang and Trevor Cohn. 2022. Incorporating constituent syntax for coreference resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10831–10839. https://doi.org/10.1609/aaai.v36i10.21329

Wei-Jen Ko, Yating Wu, Cutter Dalton, Dananjay Srinivas, Greg Durrett, and Junyi Jessy Li. 2023. Discourse analysis via questions and answers: Parsing dependency structures of questions under discussion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11181–11195. https://doi.org/10.18653/v1/2023.findings-acl.710

Fang Kong and Jian Fu. 2019. Incorporating structural information for better coreference resolution. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 5039–5045. https://doi.org/10.24963/ijcai.2019/700

Fang Kong and Guodong Zhou. 2011. Combining dependency and constituent-based syntactic information for anaphoricity determination in coreference resolution. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 410–419.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 188–197. https://doi.org/10.18653/v1/D17-1018

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692.

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. *CoRR*, abs/1907.11692.

Zhengyuan Liu and Nancy F. Chen. 2021. Improving multi-party dialogue discourse parsing via domain integration. *CoRR*, abs/2110.04526. `https://doi.org/10.18653/v1/2021.codi-main.11`

Jing Lu and Vincent Ng. 2018. Event coreference resolution: A survey of two decades of research. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 5479–5486. `https://doi.org/10.24963/ijcai.2018/773`

Jing Lu and Vincent Ng. 2021. Span-based event coreference resolution. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 13489–13497. `https://doi.org/10.1609/aaai.v35i15.17591`

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 25–32. `https://doi.org/10.3115/1220575.1220579`

Giuliano Martinelli, Edoardo Barba, and Roberto Navigli. 2024. Maverick: Efficient and accurate coreference resolution defying recent trends. *arXiv preprint arXiv: 2407.21489*. `https://doi.org/10.18653/v1/2024.acl-long.722`

Yuan Meng, Xuhao Pan, Jun Chang, and Yue Wang. 2023. Rgat: A deeper look into syntactic dependency information for coreference resolution. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE. `https://doi.org/10.1109/IJCNN54540.2023.10191577`

Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? A proposal for a link-based entity aware metric. In *Proceedings of the 54th annual meeting of the association for computational linguistics*, volume 1, pages 632–642. Association for Computational Linguistics. `https://doi.org/10.18653/v1/P16-1060`

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40. Association for Computational Linguistics.

Marc B. Vilain, John D. Burger, John S. Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding, MUC* 1995, pages 45–52. `https://doi.org/10.3115/1072399.1072405`

Ante Wang, Linfeng Song, Hui Jiang, Shaopeng Lai, Junfeng Yao, Min Zhang, and Jinsong Su. 2021. A structure self-aware model for discourse parsing on multi-party dialogues. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI* 2021, pages 3943–3949. `https://doi.org/10.24963/ijcai.2021/543`

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 6953–6963. `https://doi.org/10.18653/v1/2020.acl-main.622`

Li Yuan, Yi Cai, Jingyu Xu, Qing Li, and Tao Wang. 2024. A fine-grained network for joint multimodal entity-relation extraction. *IEEE Transactions on Knowledge and Data Engineering*. `https://doi.org/10.1109/TKDE.2024.3485107`

Duzhen Zhang, Feilong Chen, and Xiuyi Chen. 2023a. Dualgats: Dual graph attention networks for emotion recognition in conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pages 7395–7408. https://doi.org/10.18653/v1/2023.acl-long.408

Wenzheng Zhang, Sam Wiseman, and Karl Stratos. 2023b. Seq2seq is all you need for coreference resolution. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 11493–11504. https://doi.org/10.18653/v1/2023.emnlp-main.704

Li Zheng, Boyu Chen, Hao Fei, Fei Li, Shengqiong Wu, Lizi Liao, Donghong Ji, and Chong Teng. 2024a. Self-adaptive fine-grained multi-modal data augmentation for semi-supervised muti-modal coreference resolution. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8576–8585. https://doi.org/10.1145/3664647.3680966, PubMed: 38712678

Li Zheng, Hao Fei, Ting Dai, Zuquan Peng, Fei Li, Huisheng Ma, Chong Teng, and Donghong Ji. 2025. Multi-granular multimodal clue fusion for meme understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26057–26065. https://doi.org/10.1609/aaai.v39i24.34801

Li Zheng, Hao Fei, Fei Li, Bobo Li, Lizi Liao, Donghong Ji, and Chong Teng. 2024b. Reverse multi-choice dialogue commonsense inference with graph-of-thought. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19688–19696. https://doi.org/10.1609/aaai.v38i17.29942

Li Zheng, Donghong Ji, Fei Li, Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, and Chong Teng. 2023a. Ecqed: Emotion-cause quadruple extraction in dialogs. *arXiv preprint arXiv:2306.03969*.

Li Zheng, Fei Li, Yuyang Chai, Chong Teng, and Donghong Ji. 2023b. A bi-directional multi-hop inference model for joint dialog sentiment classification and act recognition. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 235–248. Springer. https://doi.org/10.1007/978-3-031-44693-1_19