machine learning & knowledge extraction

Article

Automatic Extraction and Visualization of Interaction Networks for German Fairy Tales

David Schmidt * and Frank Puppe

Artificial Intelligence and Knowledge Systems, Julius-Maximilians-Universität Würzburg, 97074 Würzburg, Bayern, Germany; frank.puppe@uni-wuerzburg.de
* Correspondence: david.b.schmidt@uni-wuerzburg.de

Abstract: Interaction networks are a method of displaying the significant characters in a narrative text and their interactions. We automatically construct interaction networks from dialogues in German fairy tales by the Brothers Grimm and subsequently visualize these networks. This requires the combination of algorithms for several tasks: coreference resolution for the identification of characters and their appearances, as well as speaker/addressee detection and the detection of dialogue boundaries for the identification of interactions. After an evaluation of the individual algorithms, the predicted networks are evaluated against benchmarks established by networks based on manually annotated coreference and speaker/addressee information. The evaluation focuses on specific components of the predicted networks, such as the nodes, as well as the overall network, employing a newly devised score. This is followed by an analysis of various types of errors that the algorithms can make, like a coreference resolution algorithm not realizing that the frog has transformed into a prince, and their impact on the created networks. We find that the quality of many predicted networks is satisfactory for use cases in which the reliability of edges and character types are not of critical importance. However, there is considerable room for improvement.

Keywords: character networks; social networks; coreference resolution; speaker attribution; fairy tales

1. Introduction

In the realm of narrative texts, characters are pivotal elements shaping the storyline. One approach to visually represent these integral components is through character networks. These networks are graphical representations where characters are depicted as nodes and their interactions or relationships are illustrated as connecting edges. An example of such a network can be observed in Figure 1, showcasing the network for *Aschenputtel (021)* (we use the original German names throughout the text and include their number for easier reference; see Appendix A for the corresponding English names). Character networks may manifest in two primary forms: static and dynamic. Static networks offer a holistic view, encapsulating the entire text, whereas dynamic networks evolve, mirroring the narrative's progression through chapters or scenes. The dynamism in these networks is particularly insightful, revealing changes in characters—as seen in *Hänsel und Grethel (015)*, where the witch dies, and *Der Froschkönig oder der eiserne Heinrich (001)*, where the frog transforms into a human being—or shifts in relationships, such as the transition from ally to adversary in *Rumpelstilzchen (055)*.

Such character networks can be helpful in a variety of ways. Visualized, they provide a quick and easy-to-understand overview of a text's important characters and their interactions with (and/or relations to) each other. Their structure, on the other hand, can help to analyze a corpus of documents. For example, Propp developed a set of character types that appear in Russian folk tales and a set of generalized steps of which they consist (e.g., "The hero leaves his/her home") [1]. We found these to not fit the German fairy tales by the Brothers Grimm, and think character networks can be helpful in developing similar sets

check for updates

Citation: Schmidt, D.; Puppe, F. Automatic Extraction and Visualization of Interaction Networks for German Fairy Tales. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 2447–2493. https://doi.org/10.3390/make6040121

Academic Editor: Jianlong Zhou

Received: 3 September 2024
Revised: 22 October 2024
Accepted: 23 October 2024
Published: 1 November 2024

of character types and plot steps for the German tales. They can also answer questions like the following: do heroes and antagonists interact with (i.e., in our case speak to) each other in each fairy tale? With the help of the character networks, we find that this is not the case for the fairy tales used in this paper. In 8 out of 62 fairy tales, there is no interaction between hero and antagonist, and in another 5 documents, not all heroes interact with all antagonists.

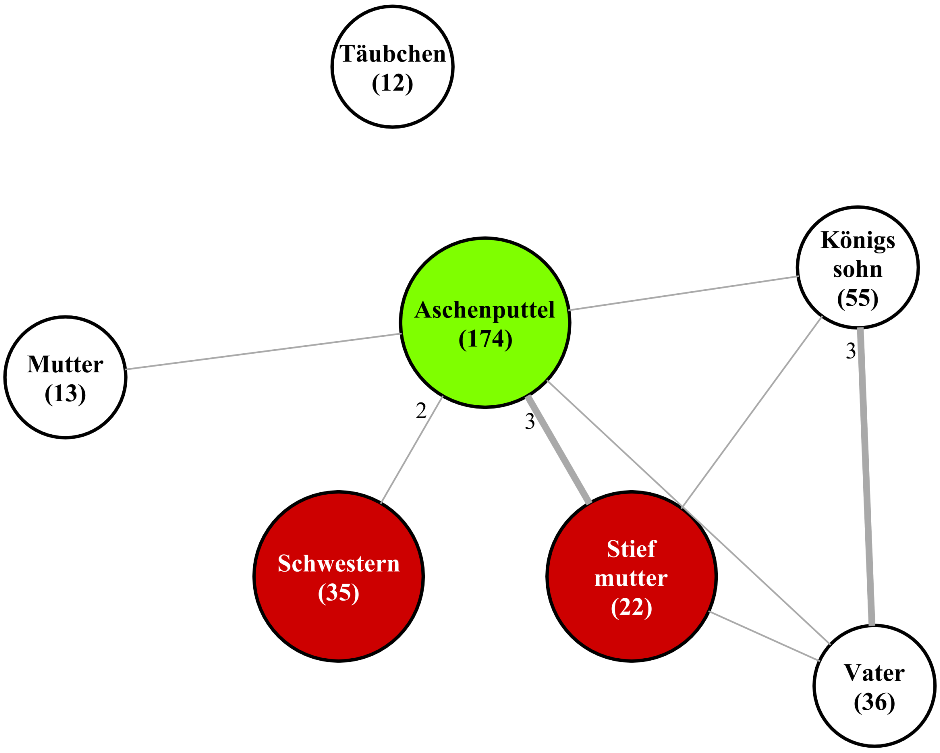


Figure 1. Visualization of the character network of the fairy tale *Aschenputtel (021)*. Numbers in the nodes show the characters' frequency of occurrence, and numbers at the edges show the frequency of the interactions. The green color marks the hero, and the red color the antagonists.

This paper present an automatic approach for the construction and visualization of character networks from German fairy tales by the Brothers Grimm. Our focus is on static networks with interactions as edges, which can serve as a foundation for more elaborate character networks that also display properties of and relationships between characters. The interactions are based on dialogues. Characters engaged in a conversation, whether as speakers or listeners, are considered to be interacting like in [2].

We evaluate the results of the individual algorithms involved in the creation of the character networks, as well as the quality of several aspects of the created networks. Additionally, we devise a new score in order to judge the quality of the complete network, and perform several ablation studies in order to measure the impact of various types of errors made by the algorithms on the predicted character networks.

The structure of this paper is arranged as follows: Section 2 discusses related work in this field and is followed by Section 3, which introduces the data utilized in this study. Next, Section 4 delves into the core algorithms underpinning the automatic creation of character networks, coreference resolution and speaker attribution. The construction process of character networks is elucidated in Section 5. Section 6 details the visualization techniques employed for these networks. Section 7 presents an evaluation of coreference resolution and speaker attribution algorithms as well as the generated networks. The impact of different algorithmic components on the networks is analyzed in Section 8.

2. Related Work

In the field of character network extraction from narrative texts, both manual and automatic approaches are employed, spanning various media types including literature, comics, and movies. For a comprehensive overview, see [3] as an example. This paper, however, narrows its focus to character networks that are automatically built from narrative text.

Elson et al. [4] developed a methodology to construct character networks from conversations in 60 nineteenth-century novels and serials. Their objective was to test two literary hypotheses. They utilized the Stanford NER tagger to extract noun phrases and created clusters by generating name variations and attributed speakers to direct speech through a rule-based algorithm. While they evaluated their method for detecting conversations between characters in four novels, they did not assess the networks themselves.

Agarwal et al. [5] formed networks representing social interactions. They applied an SVM with tree kernels to predict social events, testing their system on “Alice in Wonderland”. The network they extracted was statistically comparable to a gold-standard network, despite the social event detection algorithm achieving only a 61% F1 score.

Ardanuy et al. [6] employed StanfordNER for identifying name mentions, and additionally identified titles, first and last names, and assigned genders. They used these data for coreference resolution. In their networks, links indicated co-occurrence in paragraphs. Instead of evaluating the networks directly, they clustered novels based on features from the networks for the unsupervised attribution of genre and author.

Trovati et al. [7] detected character references using patterns and categorized relationships between characters as friendly, hostile, or unknown based on verb usage. They evaluated their approach on “The Legends Of King Arthur And His Knights” by James Knowles, achieving 80% precision and 68% recall in identifying relationships, with 55% accuracy in relationship type.

Waumans et al. [2] built networks from conversations in 47 novels, requiring manual preprocessing to mark direct speech and scene breaks. Their comparison with manually annotated networks on “Harry Potter and the Philosopher’s Stone” showed 56.9% accuracy for identifying speakers in direct speech and 82.4% in dialogues; 73.3% of predicted network edges matched the correct network.

Dekker et al. [8] evaluated four NER tools on the first chapters of 20 classic and 20 modern novels, assessing their robustness to language changes over time. The tools performed better on modern novels. They also constructed interaction networks based on sentence co-occurrences, observing significant variations in network features like average degree and path length, but no clear distinctions between classic and modern novels.

Edwards et al. [9] used the script of the TV show “Friends” to compare manually identified interaction networks with those generated automatically based on scene co-occurrences and based on direct speech. They found high correlations between these networks in various metrics, including size and edge density.

Krug [10] focused on German novels, comparing networks predicted automatically from the novels with those built from manually annotated expert summaries. They utilized the Kuhn–Munkres algorithm for mapping the nodes in the predicted networks to the nodes in the gold networks, evaluating character nodes and interactions through various experiments. These included co-occurrence in sentences and paragraphs, and communication in direct speech utterances. They assessed networks using topological properties and Spearman’s rank correlation coefficient, and also evaluated the detection of specific types of relations like family and social connections, using the summaries as a reference.

Agarwal et al. [11] built character networks for 500 works of fanfiction of Harry Potter, counting co-occurrences within sentences as interactions. Using the eigenvalues of the graphs’ Laplacian matrices, they found the closest graphs for each document and then applied classifiers like SVMs to these closest graphs in order to determine the genre of the document.

Marienberg-Milikowsky et al. [12] built three types of character networks for novels of two Israeli writers: one network where edges between characters are based on manually annotated communication between them, another where they are based on automatically detected appearance in the same paragraph, and a third where they are based on semantic similarity, which is determined via a word2vec model trained on the corresponding novel. Pairs of these graphs are combined into a new one where the edges display the differences between the original graphs, e.g., which pairs of similar characters appear together often, and which do so only rarely. They used these as starting points for potentially interesting further studies of the novels, which is probably why they did not evaluate the quality of the automatically predicted networks.

Perri et al. [13] automatically built character networks for the works of Tolkien's Legendarium, where edges represent co-occurrence on the sentence-level. They initially used BookNLP (https://github.com/booknlp/booknlp, accessed on 22 October 2024) for coreference resolution but were not satisfied with the results. Since the documents in their corpus contain a large number of named references, they decided to use a set of rules that are only applied to names but yield high precision. They applied graph neural networks on the predicted networks in order to extract additional information about the narratives from them, like missing edges caused by their method for coreference resolution.

To summarize, there is a lot of work regarding the creation of character networks, but most only evaluate parts of the process (e.g., speaker attribution or relation detection). The ones that evaluate the complete networks carry this out by comparing properties like the average degree or betweenness centrality. In contrast, we perform a more fine-grained evaluation by first mapping nodes in the predicted and hand-annotated networks to each other, which allows us to identify which nodes and edges are missing in each network. The work of Krug [10] is similar to ours as it also performs the mapping step beforehand, but uses expert summaries instead of the text for a lot of evaluations. Additionally, we introduce a score for the complete networks based on the accuracy of predicted nodes and edges.

Knowledge graphs are similar to interaction networks in that they display connections between entities. Their main difference is that they are designed to show relationships between entities, as opposed to interactions. This could help enrich the interaction networks with additional information (i.e., the relationships between the characters). However, we are not aware of any works that apply knowledge graph construction to narrative texts. For a survey of knowledge graph construction, see [14] as an example.

3. Data

The data set we used in this work is part of the FairyNet Corpus [15]. It contains the fairy tales of the seventh edition of the Children's and Household Tales by the Brothers Grimm for the German language. These tales contain self-contained stories, so we can build complete networks from the texts alone. Their main advantage comes from usually being shorter than a lot of other narrative texts like novels, so they can be annotated more easily and more quickly. In contrast, corpora for coreference resolution that consist of novels, like DROC [16] and Litbank [17], only annotate a part of each narrative due to the amount of work required. The fact that they are shorter and contain less characters also makes them easier to compare. Additionally, many current algorithms for coreference resolution have high memory requirements that are proportional to the document's length and make using them for longer documents difficult. (In fact, even some of the fairy tales prove to be challenging. *Die zwei Brüder (060)*, which is not used in this work, is too large to be predicted by ASP on a graphics card with 24 GB of memory). We expect that the methods we use in our experiments can be applied to full-size novels as well (at least when using the appropriate coreference resolution algorithms, e.g., long-doc-coref). The only limitation we see is the likely worse performance of the coreference resolution algorithms, based on experiments by Krug [10] where he compares the performance on the complete Effi Briest novel to that on DROC fragments.

In each document, character references (mentions) are annotated by hand. The mentions do not cover complete noun phrases like in Ontonotes [18] (e.g., [die sieben jungen

Geißlein]/[the seven young goats]) but only the heads like in DROC [16] (e.g., die sieben jungen [Geißlein]/the seven young [goats]). Each mention is hand-annotated with the following information:

- Coreference ID (the mentions referring to the same character receive the same ID);
- Semantic category (human, legendary being, transcendent being, generic, mixed);
- Syntactic category (name, noun phrase, pronoun);
- Biological sex (i.e., Mädchen/girl is marked as female, not neuter);
- Grammatical number.

The data set's annotation of characters includes various types of entities. The criteria for what is annotated as a character in the narrative texts are as follows:

- Humans and Humanoid Beings: This category includes not only humans but also humanoid beings such as dwarves and giants. These entities are annotated as characters regardless of their role or extent of participation in the narrative.
- Animals: The annotation of animals as characters is context-dependent. In narratives where animals are the main characters (like *Die Bremer Stadtmusikanten (027)*), all animals are annotated. However, in texts where animals are not central, only those animals that exhibit the ability to speak (or are perceived as having this ability, like the frogs and dogs in *Der gute Handel (007)*) are annotated as characters.
- Other Beings: Entities that do not fall into the categories of humans, humanoids, or animals are also considered characters if they engage in speech or action within the narrative. This broadens the scope of character annotation to include even objects like brooms, wells or moons if necessary (e.g., in *Brüderchen und Schwesterchen (011)*).
- Generic References and Transcendent Beings: Although not central to the character networks, generic references to types or groups of characters, as well as transcendent beings like God or the Devil, are also annotated as characters.

In some of the fairy tales, characters are disguised (like the queen/stepmother in *Sneewittchen (053)*) or transformed into animals or objects (e.g., Johannes in *Der treue Johannes (006)*). They are annotated as the same entity throughout the narrative, even if the reader learns about it afterwards (*König Drosselbart (052)*).

In documents 001 to 070 (out of 66 in total, 5 of the documents from 001 to 070 are not annotated because they are written in dialects that the average German speaker does not understand, while another 2 documents (1 of which is at least twice as long as all the other documents) are skipped because of the coreference resolution models, and 2 other documents are split into several documents because they contain several distinct fairy tales), direct speech utterances, their speakers and addressees and dialogues are also annotated by hand. For documents 071 to 130, this information is instead annotated by algorithms, which is why these documents are not used for evaluation but only for training the coreference resolution models.

Since some characters are more important than others, we also annotate which characters are the most important ones: fairy tale heroes (Märchenheld) and antagonists. The hero is the character who is most important and whom the reader can most easily relate to. In *Der Wolf und die sieben jungen Geislein (005)*, for example, we annotate the little goats as heroes even though the wolf is a very important character as well. In most fairy tales, there is only one hero but there are also a few where we label several characters as heroes, for instance in *Hänsel und Grethel (015)* and *Die Bremer Stadtmusikanten (027)*. An antagonist is a character who is in serious conflict with the hero, often by wanting to harm him/her (e.g., the stepmother in *Sneewittchen (053)*), prevent him/her from achieving his/her goals or take his/her place (e.g., the stepsisters in *Aschenputtel (021)*). However, there are also instances where the hero may be considered the primary instigator of the conflict between hero and antagonist. For example, in the fairy tales *Der Froschkönig oder der eiserne Heinrich (001)* and *Rumpelstilzchen (055)*, the princess and the miller's daughter both refuse to fulfil their respective commitments, thereby initiating the conflict and turning characters that have been helpful previously into antagonists. The conflict with the antagonist has to be an important part of the plot and not only an obstacle that has to be overcome. For instance, we do not consider the creatures that the tailor has to defeat in *Das tapfere Schneiderlein (020)* or the wild boar that the younger brother kills in *Der singende Knochen (028)* to be antagonists. Instead, we annotate the king who repeatedly invents new tasks in order to get rid of the tailor, and the older brother who kills his sibling, as antagonists.

In most cases, it is clear who the the hero is in a fairy tale, but there are also more difficult cases, two of which we will elaborate on (we find it very difficult, if not impossible, to define these types of characters without leaving any room for interpretation). In *Läuschen und Flöhchen (030)*, we do not annotate any character as a hero since none of them appears more than once or seems to be more important than the others (the plot is basically "A talks to B", "B talks to C" and so on, until the last character reacts in a way that kills everyone). For similar reasons, we do not annotate any character as an antagonist. On the other hand, both the cat and the mouse can be seen as the hero in *Katze und Maus in Gesellschaft (002)*. In the end, we decide to label the mouse as a hero and the cat as an antagonist, partially because the cat fits the label of antagonist better and the mouse seems more relatable. For antagonists, there are more cases where it can be difficult to decide whether a character should count as an antagonist or not. See *Die zwölf Brüder (009)* for an example, where the king wants to kill his sons at the beginning of the fairy tale. This is a situation similar to that in *Hänsel und Grethel (015)*, where the stepmother wants to get rid of the children, so it could be argued that the king should be an antagonist. However, we ultimately decide against this labeling because the conflict in this situation is not between the king and the protagonist, and the conflict is a small part of the plot (the sons simply move to the forest and live there without any apparent problems).

All evaluations in this paper are performed on the documents with numbers 001 to 070. For the neural networks used for coreference resolution, these documents are randomly split into five parts. For each part, the algorithms are trained on the remaining four parts, the documents 071 to 130 and DROC. That means each coreference resolution algorithm is trained five times so we can make predictions for all documents from 001 to 070. Since the rule-based algorithms do not require a training set, they are simply applied to all documents at once.

The network files presented in [15] are not used in this work.

4. Basic Steps for Interaction Networks from Dialogues

For the construction of character interaction networks from dialogues in narrative texts, it is imperative to initially identify the characters and their respective references, as well as the speakers and addressees of each direct speech utterance. Furthermore, it is crucial to determine which direct speech utterances are part of the same dialogue. This process is divided into two distinct tasks: coreference resolution and speaker attribution. Coreference resolution algorithms are tasked with identifying characters and their references in the text, while speaker attribution algorithms focus on determining the speakers and addressees of the direct speech (grouping the utterances into dialogues in the process).

This section delves into both these critical steps and discusses the specific algorithms employed in our experiments. It is important to note the interdependency of these tasks. For effective speaker attribution, it is beneficial to understand the number of entities present in the context of a direct speech utterance, a question that can be addressed through coreference resolution. Conversely, coreference resolution algorithms often utilize information about the speakers and addressees of direct speech utterances. For example, first-person pronouns in an utterance typically refer to the speaker of that utterance.

In our experiments, we employed a rule-based coreference resolution algorithm within the speaker attribution algorithm. Although this approach is less effective than neural network-based methods for complete documents, as evidenced in Section 7.1, it is significantly simpler to apply and sufficiently effective for the smaller document segments involved in the speaker attribution algorithm (coreference resolution is only performed within a dialogue and its context, separately for each dialogue). Notably, with manually annotated coreference information, the results for speaker attribution remained unchanged, while addressee attribution improved by a modest 1.4 points.

4.1. Coreference Resolution

Coreference resolution is the task of identifying which mentions found in a text refer to the same entity. It is therefore one of the most important tasks when building character networks. Identifying all speakers correctly is of no use when we do not know which of these speakers refer to the same character and which refer to different characters.

In this work, we experimented with several different algorithms for coreference resolution: an adaptation of the Sieves algorithm [19] to German [20], c2f [21], long-doc-coref [22] and ASP [23]. In addition to that, we conducted some experiments using ChatGPT for coreference resolution. These experiments with ChatGPT are not strictly comparable to the other coreference resolution algorithms as they were not performed fully automatically (e.g., the outputs were checked manually for grave errors).

4.1.1. Sieves Algorithm

This algorithm consists of rules that are grouped into so-called sieves [19]. The sieves are ordered by their precision and applied one after the other so each sieve can build upon the results of the previous sieves. Most sieves contain rules that are based on string matching with decreasingly stringent conditions. Other sieves use the results of the speaker detection in order to group first-/second-person pronouns in a direct speech utterance with their speaker/addressee or resolve relative pronouns, appositions or copulae based on dependency parse trees. Pronouns (with the exception of first-/second-person pronouns inside direct speech utterances and reflexive pronouns) are exclusively handled in two separate sieves that are applied at the end since their precision is very low.

The implementation we use [20] does not include mention detection or speaker detection. Instead, we use the results of the speaker detection algorithm and a rule-based algorithm for character reference detection [24].

4.1.2. Higher-Order Coreference Resolution with Coarse-to-Fine Inference

c2f [21] is an extension of e2e [25], the first end-to-end neural network for coreference resolution. It starts by building representations for all possible contiguous spans in a text up to a predefined number of tokens. These span representations are based on token representations produced by an encoder like ELMo [26] (in [21]) or Bert [27]. They are scored by a feed-forward neural network and the lowest-scoring 70% of spans are discarded. The remaining spans are paired with their possible antecedents, and another feed-forward neural network scores each pair. The highest-scoring possible antecedent is then picked as antecedent.

The first difference in c2f compared to e2e is the introduction of a coarse but fast scorer used to prune the possible antecedents of each mention, which lessens the runtime impact of the slow pair-wise coreference scorer of the original paper. The other major difference is the higher-order component that iteratively refines the mention representations depending on the results of the pair-wise coreference scorer. This allows each mention to incorporate information about other mentions in the same cluster and the scorer to make better decisions.

Note that pair-wise scoring is performed for all pairs at once, which requires a lot of memory and makes the model difficult if not impossible to use on long documents (depending on the hardware at one's disposal).

We used the implementation of [28] for our experiments with severinsimmler/literary-german-bert (https://huggingface.co/severinsimmler/literary-german-bert, accessed on 24 October 2024) as the pre-trained Bert model. The maximum span width was set to 5 since the spans in the corpus we used for our experiments are much shorter than in a lot of other corpora like Ontonotes.

4.1.3. Long Document Coreference with Bounded Memory Neural Networks

Like e2e, long-doc-coref [22] encodes the document and predicts mentions in the first step. It does not, however, score pairs of mentions in order to find an antecedent for each mention. Long-doc-coref instead stores representations of a limited number of entities and iterates over the mentions in the order in which they appear in the text. Each mention is scored with the existing entity clusters to decide whether it is added to one of the existing clusters, added to a new cluster (which replaces one of the existing clusters if memory capacity is reached) or ignored (either because it is not actually a mention or because all memory slots are full and its cluster is less important than the existing entity clusters).

In our experiments, we used the learned bounded memory variant, which uses a neural network to decide which (if any) cluster should be replaced if the current mention belongs to a new cluster but all memory slots are full. Since the model itself requires little memory, we were able to use a large Bert model as an encoder: deepset/gbert-large (https://huggingface.co/deepset/gbert-large, accessed on 24 October 2024).

4.1.4. Autoregressive Structured Prediction with Language Models

ASP [23] uses, as the name implies, an autoregressive language model for tasks that have a structured output, like named entity recognition, coreference resolution and relation extraction. Instead of encoding the structures in a string, it builds them by predicting several types of actions.

It iterates over the document text token by token and predicts at each step one of three structure-building actions: [*,] and copy. The brackets are inserted into the text and are supposed to mark the beginning and end of mentions, while copy simply copies the current token from the input to the output. For each closing bracket, the model also predicts which of the preceding opening brackets it is paired with. Each opening bracket can be paired with an arbitrary number of closing brackets in this way. As an example, the output for the text “US President Joe Biden took office in 2021” would be “[* US] President Joe Biden] took office in 2021” because there are two mentions: “US” and “US President Joe Biden”. For the coreference resolution task, the model additionally predicts an antecedent for each mention by linking their respective closing brackets.

As the pre-trained language model, we used t5-base (https://huggingface.co/t5-base, accessed on 24 October 2024), since larger models did not work due to memory restrictions.

4.1.5. Coreference Resolution with ChatGPT

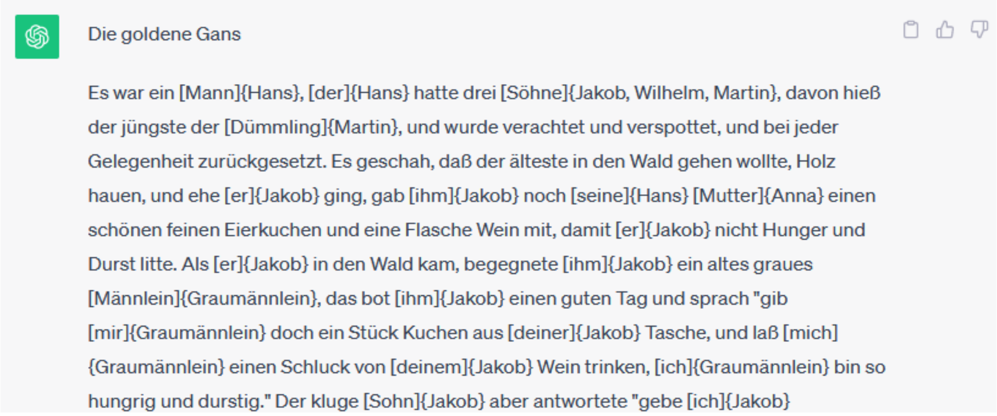
ChatGPT (https://openai.com/blog/chatgpt, accessed on 24 October 2024) is a chatbot by OpenAI based on GPT-3.5 and GPT-4. Most of our experiments were conducted with the GPT-4 version of May and June 2023, with the exception of 23 documents for the ChatGPT S experiment, which were annotated with the October version. ChatGPT is built to output coherent text, mostly in continuous form, so the general idea is to provide it with a document’s text and have it insert cluster IDs into the text. In contrast to other approaches that only output lists (e.g., a cluster ID for each mention), this approach facilitates automatic processing by enabling the straightforward connection of these cluster IDs to their respective textual mentions, which helps in mitigating the impact of reliability issues. Specifically, if ChatGPT overlooks or skips a mention, the presence of cluster IDs within the text ensures that subsequent IDs can still be assigned to the correct mentions in spite of this omission.

Since prompt inputs and outputs were limited, the text of each document was first split at sentence borders into parts with at most 600 tokens (the limits have increased since we ran our experiments, so the document parts can be larger now). Each mention was surrounded with square brackets, and one of the prompts in Figure 2 was appended to each part. The first part received the first prompt, and all other parts the second prompt.

The parts were then entered into the chat window individually but in the same chat session so the model could (at least theoretically) have access to the previous text parts (in practice, the context windows of language models are limited, so the model will eventually not have access to all of the previously processed text if the document is large enough). Figure 3 shows an example of what the answer of ChatGPT is supposed to look like. The answers of ChatGPT were copied out of the chat window and concatenated in a text file. This file was then parsed and the results were annotated in copies of the original files. Since ChatGPT might omit input tokens or add extra tokens, the parser aligned the answer with the input text using the Needleman–Wunsch Algorithm [29], similar to what is carried out in [30].

> <text>
> -
> Give each character in the text above a unique identifier, e.g., a name. Repeat the text and, after each mention that is marked by brackets, add the unique identifier of the corresponding character in curly braces into the text.
>
> <text>
> - - - - - - - - - - - - - - - - - - -
> Continue repeating the text and adding, after each mention that is marked by brackets, the unique identifier of the corresponding character in curly braces into the text.

Figure 2. The prompts used for coreference resolution with ChatGPT. The upper prompt is used for the first part of each document, and the lower prompt for all other parts. <text> is a placeholder for the text of the document part to be processed.



Die goldene Gans

Es war ein [Mann]{Hans}, [der]{Hans} hatte drei [Söhne]{Jakob, Wilhelm, Martin}, davon hieß der jüngste der [Dümmling]{Martin}, und wurde verachtet und verspottet, und bei jeder Gelegenheit zurückgesetzt. Es geschah, daß der älteste in den Wald gehen wollte, Holz hauen, und ehe [er]{Jakob} ging, gab [ihm]{Jakob} noch [seine]{Hans} [Mutter]{Anna} einen schönen feinen Eierkuchen und eine Flasche Wein mit, damit [er]{Jakob} nicht Hunger und Durst litte. Als [er]{Jakob} in den Wald kam, begegnete [ihm]{Jakob} ein altes graues [Männlein]{Graumännlein}, das bot [ihm]{Jakob} einen guten Tag und sprach "gib [mir]{Graumännlein} doch ein Stück Kuchen aus [deiner]{Jakob} Tasche, und laß [mich] {Graumännlein} einen Schluck von [deinem]{Jakob} Wein trinken, [ich]{Graumännlein} bin so hungrig und durstig." Der kluge [Sohn]{Jakob} aber antwortete "gebe [ich]{Jakob}

Figure 3. Example of the answers ChatGPT produces. The mentions are marked by square brackets and the identifiers ChatGPT gave the characters are marked by curly braces.

At the time of these experiments, there was no API access available, so the interactions with ChatGPT had to be carried out manually. Some of the problems that occurred during the experiments were as follows:

- Sometimes the model did not repeat all of the input text but stopped at some point, and had to be prompted to continue. This could have led to whitespace being inserted inside words.
- In some cases, the model did not repeat the input text at all but tried to continue the story instead. Telling it "you are supposed to repeat the text" often led to it performing the task correctly. In the cases where that did not work, the complete document was processed anew (in a new chat session).
- The model sometimes marked additional words as mentions (e.g., things like a castle) or did not mark some of the given mentions (often reflexive pronouns or generic references, which should have little to no impact on the character networks).
- The model did not always conform to the specified output format. The two common mistakes were omitting the square brackets or putting the identifier and curly braces

inside the square brackets. Both could be fixed with the help of regular expressions and slight adjustments in the parsing script.

- The responses to the first part of a document often contained unnecessary comments and the responses to telling it "you are supposed to repeat the text" usually started with an apology. This superfluous text was removed when copying the responses to the text files.
- Some fairy tales contain violent content (e.g., in *Der Räuberbräutigam (040)*, a young woman is killed and her corpse is cut into pieces). Mostly, ChatGPT just added a note that this content may violate the content policy, but in a few rare cases, it outright refused the request.
- The October version changed the identifiers within a document from one prompt to the next, e.g., from names to numbers, several times.

There was also a positive observation: in a lot of cases, ChatGPT did not create a new identifier for groups of characters but instead listed the individual characters the group consisted of, e.g., {Hänsel, Gretel} for the mention Kinder (children). This is probably useful for character networks but current coreference resolution models usually do not carry this out.

Our experiments with ChatGPT are similar to those in the document template of [31], who use a slightly different pattern for marking mentions and clusters. They use several language models for coreference resolution and compare their performance to that of supervised coreference resolution models.

4.2. Dialogue-Based Speaker Detection

After finding the characters and where they appear in a text via coreference resolution, we need to identify the speakers and addressees of the direct speech utterances and group these utterances into dialogues in order to find out where the characters interact with each other. The speaker detection algorithm we use is an optimized version (the changes made to [32] are only small, mainly adjusted weights, expanded word lists and updated propagation rules) of the algorithm by [32] (also described in [10]) and consists of two major stages:

1. Find explicitly mentioned speakers (e.g., sie/she in Figure 4) and addressees (e.g., Mädchen/girl in Figure 4) for each direct speech utterance;
2. Propagate the speakers and addressees found in the first stage to other utterances within the same dialogue.

Both of these steps require preparatory steps. In order to detect the explicit speaker or addressee, it is necessary to identify the context of each utterance in which the speaker or addressee may occur. The propagation step, on the other hand, necessitates that direct speech utterances be grouped into dialogues. These dialogues are later also used for predicting interactions in the character network.

"Heda, Grethel," rief sie dem Mädchen zu, "sei flink und trage Wasser"
"Heda, Grethel," she shouted to the girl, "be quick and carry water"

Figure 4. Example for a direct speech utterance that is split into two parts with an explicitly mentioned speaker (sie/she) and addressee (Grethel/Mädchen/girl).

It is important to note that we skip the prediction of direct speech utterances in our experiments and use hand-annotated utterances instead (direct speech utterances are almost always marked by quotation marks in these fairy tales, but the algorithm we experimented with failed reliably when it encountered nested utterances).

4.2.1. Determining contexts for utterances

As a preprocessing step for the detection of speakers/addressees, all direct speech utterances in the text are collected and the context of each utterance is determined (divided into a preceding and a following context). The preceding context of an utterance is the text before the utterance up to the previous sentence boundary, and the following context is the text after the utterance up to the next sentence boundary. The context may not be part of another direct speech utterance unless we are dealing with an embedded utterance. An utterance that is immediately proceeded and followed by other utterances would therefore have no context at all. Each context is also assigned a boundLevel, indicating how strongly it is connected to its direct speech utterance. For example, contexts that are in the same sentence as the utterance, only separated by a comma, or that (in the case of preceding contexts) end with a colon receive a high boundLevel, while those that are clearly separated, e.g., by an exclamation or question mark, receive a low boundLevel.

4.2.2. Explicit Speaker Detection

The contexts are needed for the first stage of the speaker resolution algorithm. All mentions in an utterance's context are considered candidates for explicitly mentioned speakers and are scored. Those in a context that do not have a high boundLevel only receive a slightly positive score if they are the only candidate in their context. All others can receive additional positive scores, e.g., if they are a subject (especially if the corresponding predicate is in a list of "speaking" verbs) or if they are in a preceding context that ends with a colon (with no comma between the candidate and the colon). The resolution of addressees works the same, only with some additional candidates from within the utterances (e.g., Grethel in Figure 4) and different scoring conditions (e.g., positive scores for being an object instead of a subject or negative scores for possessive pronouns).

4.2.3. Dialogue Detection

Prior to the initiation of the second stage of our speaker detection algorithm, several preparatory steps are necessary. One critical step involves the consolidation of direct speech utterances that are fragmented. This situation may occur when an utterance is interrupted by narrative inserts, like in Figure 4. By merging such utterances and consequentially regarding them as a single utterance, we preserve the integrity of the dialogue flow, which is essential for accurate speaker and addressee identification. In addition to utterance consolidation, the algorithm assesses the distribution of the boundLevel across utterances within a single paragraph. If it is determined that only one utterance within a paragraph possesses a high boundLevel, the algorithm assigns the identified speaker of that utterance to all other utterances within the same paragraph.

The construction of dialogues represents another fundamental component of the process. Utterances are grouped into the same dialogue if they share overlapping contexts without an intervening change of scene or time, as indicated by words such as "Später/Later" or "Nachdem/After". Furthermore, utterances that succeed one another with no intervening text are also considered part of the same dialogue. Lastly, dialogues that are contiguous with each other and are separated by contexts that immediately follow one another are subject to merging. This merger is contingent upon the first dialogue in the pair having a singular identified speaker. In order to determine the number of speakers in a dialogue, the algorithm performs coreference resolution on the part of the text covered by the dialogue and its contexts using the Sieves algorithm (Section 4.1.1).

4.2.4. Speaker Propagation

In the second stage of the speaker detection algorithm, we introduce a set of rules that enable the propagation of information to establish speaker and addressee pairs across dialogues. This stage is predicated on several foundational principles that govern the interaction patterns frequently observed in dialogues. A pivotal rule pertains to the concept of response cues within utterances. An utterance that is identified as a response (e.g., Er antwortete/He answered) indicates a shift in roles, whereby the speaker of the current utterance is the addressee of the preceding one, and reciprocally, the speaker of the preceding utterance becomes the addressee of the current one. This back-and-forth pattern is a common structural feature in dialogues and serves as a reliable indicator of speaker–addressee relationships.

In situations where a dialogue is composed of only two speaking entities, the algorithm employs a simplifying assumption to fill vacant speaker or addressee slots in utterances. When one of these slots is already occupied by an identified speaker or addressee, the other slot can be filled by the remaining entity by default. This binary structure allows for a straightforward allocation of roles within the conversational exchange. The same principle can be applied if a dialogue features a single speaking entity and the context includes two entities in total.

Afterwards, a set of propagation rules is applied to the dialogues. Each rule has conditions for the filled speaker and addressee slots and can itself fill one or more of the empty slots if all conditions are met. For example, if the identities of the speaker and addressee in the initial utterance are known and the speaker of the subsequent utterance is the addressee of the first, it is inferred that the speaker of the first utterance assumes the role of the addressee in the second. If there are still empty slots for speakers and addressees, a more risky procedure is used next. For every pair of utterances that are in the same paragraph, do not have the same speaker and are not separated by any intervening text, the algorithm simply fills the speaker slots of both with the addressee of the other, and vice versa. This is partially based in on the assumption that there are only two characters involved in most dialogues, which does not hold true for all (narrative) texts. However, in the case of the fairy tales used in this work, it is largely accurate (about 6% of dialogues involve more than two characters, measured after all experiments). The propagation rules are applied a second time after this.

5. Building Interaction Networks

In this section, we discuss how we build the interaction networks after we have identified the characters and their mentions as well as the direct speech utterances and their speakers & addressees. The necessary steps and how they are connected can be seen in Figure 5. Note that the steps in green boxes are the same for both gold and system networks; their differences (i.e., errors in the system networks) come from the algorithms used for the steps in blue boxes.

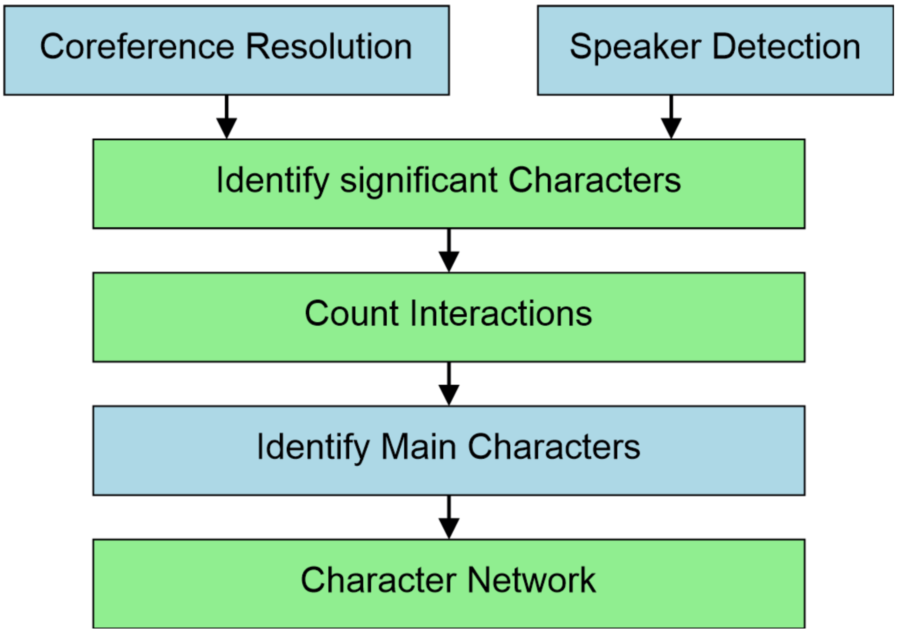


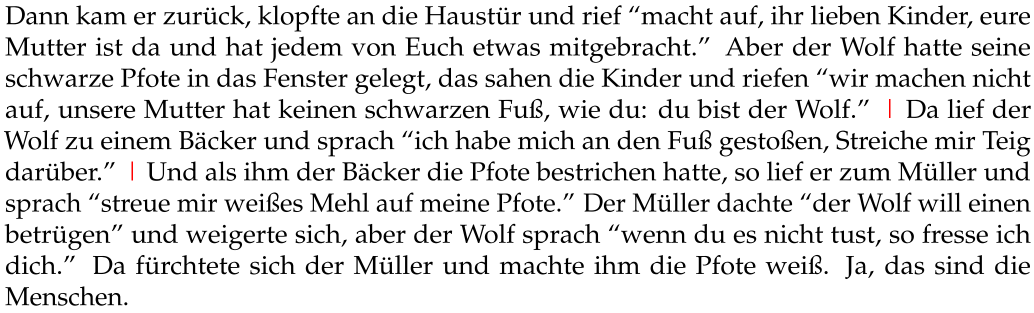
Figure 5. The steps required to build a character network based on the participation in dialogues. Those in green boxes are the same for both gold and system networks.

5.1. Significant Characters

The foundational step in constructing an interaction network from a literary text is the identification of significant characters. These characters form the nodes of the network. The criterion for a character's significance is twofold. Firstly, a character must be denoted as a speaker in at least one instance of direct speech within the narrative. Secondly, the character must be referenced on more than three separate occasions throughout the text, like in the work of [4]. Characters also must have proper denominations or identifiers that set them apart, ensuring that each node in the network represents a distinct and identifiable figure within the story. For this reason, characters that are solely referenced by pronouns are excluded in this process.

5.2. Interactions

Once the characters have been established, the subsequent phase involves finding their interactions. For the purpose of our network, an interaction is counted when both characters are involved in the same dialogue, either as speaker or addressee. An example with three dialogues/interactions can be seen in Figure 6. It is important to note that the network treats interactions as symmetrical, the directionality of the dialogue is not recorded, thereby disregarding who is the speaker and who is the addressee. This approach simplifies the network and reflects the reciprocal nature of a conversation where the participation of both parties is essential. Furthermore, we standardize the weight of interactions. Each interaction, regardless of the length or substance of the dialogue, is counted equally. This decision is rooted in the aim to quantify the mere occurrence of dialogues rather than their qualitative or quantitative dimensions.



Dann kam er zurück, klopfte an die Haustür und rief "macht auf, ihr lieben Kinder, eure Mutter ist da und hat jedem von Euch etwas mitgebracht." Aber der Wolf hatte seine schwarze Pfote in das Fenster gelegt, das sahen die Kinder und riefen "wir machen nicht auf, unsere Mutter hat keinen schwarzen Fuß, wie du: du bist der Wolf." | Da lief der Wolf zu einem Bäcker und sprach "ich habe mich an den Fuß gestoßen, Streiche mir Teig darüber." | Und als ihm der Bäcker die Pfote bestrichen hatte, so lief er zum Müller und sprach "streue mir weißes Mehl auf meine Pfote." Der Müller dachte "der Wolf will einen betrügen" und weigerte sich, aber der Wolf sprach "wenn du es nicht tust, so fresse ich dich." Da fürchtete sich der Müller und machte ihm die Pfote weiß. Ja, das sind die Menschen.

Figure 6. Three dialogues, of which each counts as one interaction from *Der Wolf und die sieben jungen Geislein (005)*. | was inserted into the text to mark the boundaries between the dialogues.

The next step is to determine the interaction strength between characters within the fairy tale's network. Even though documents vary significantly in length and in the number of direct speech utterances or dialogues, we use a fixed value in order to establish an interaction strength threshold. This is based on our observation that most of the edges in the gold networks have a very low weight (about 61% have a weight of 1). The interaction frequencies are categorized into two distinct levels:

1. Weak Interactions: If the count of interactions between any two characters is less than three, it is classified as a weak interaction. These interactions indicate a sporadic or peripheral connection between characters.
2. Strong Interactions: Interactions that reach or exceed a count of three are deemed strong. These represent the most frequently interacting characters, highlighting significant relationships that are likely to be central to the narrative's progression.

5.3. Main Characters

In order to underscore their importance for the fairy tale, the main characters (heroes and antagonists) are distinguished by their look and placement in the visualizations of the character networks, which makes their identification crucial. To address this, we developed an algorithm aimed at predicting these central characters based on the network, without differentiating between heroes and antagonists. Our scoring system for characters is based on several factors: The first two are the character's size (quantified by the number of times they are mentioned in the text) and their total number of interactions within the network. The third is the character's spread, i.e., the distance between their first and last occurrence, normalized by the text length. We also include a bonus for characters who appear in the headline of the text to further refine this scoring, recognizing the narrative emphasis often placed on these characters. After computing the scores, the characters are sorted in descending order, and the algorithm selects characters for the list of central characters in an iterative manner. This process continues until either eight central characters have been identified or a character with a score less than 40% of the previously chosen character's score is encountered. The threshold value and the bonus score are determined with the help of 20 additional documents, which are not used in this work otherwise.

6. Visualization

The final step is the actual visualization of the character networks. In these networks, characters are represented by nodes, and the interactions between them by edges. Each node is labeled with the corresponding character's most frequently used name or noun phrase, as well as its frequency (i.e., its number of mentions). In an effort to mark the most important characters and their roles, heroes are visually distinguished by a green and antagonists by a red background, while all other characters are presented with a neutral white one. The edges that connect these nodes are not uniform; instead, their breadth is proportionate to the strength of interaction they represent (partitioned into weak and strong), offering a visual indication of the interaction's intensity or frequency. Each edge is also labeled with its interaction count. For an example, see Figure 7, which displays the character network for the fairy tale *Rapunzel (012)*.

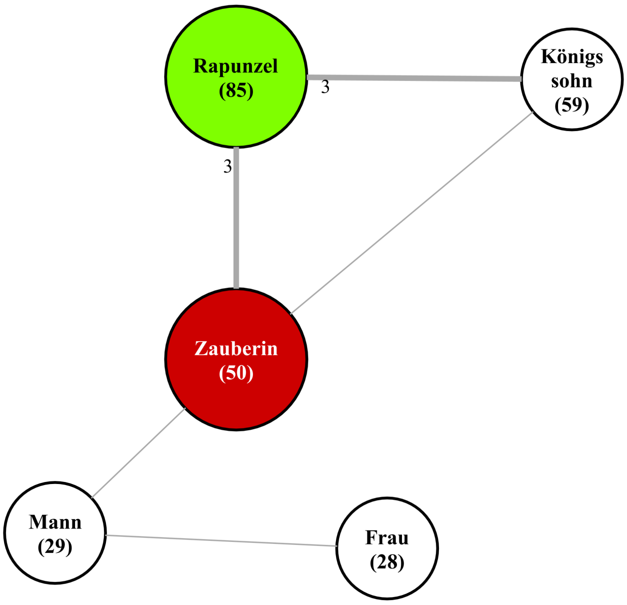


Figure 7. Visualization of the character network of the fairy tale *Rapunzel (012)*. Numbers in the nodes show the characters' frequency of occurrence, and numbers at the edges show the frequency of the interactions. The green color marks the hero, and the red color the antagonists.

In some fairy tales, groups of characters appear as a node in the network in addition to the individual characters that the group consists of. Figure 8 shows such a network on the right: there are individual nodes for Hänsel and Grethel, as well as a node for Kinder (children), which represents the group consisting of Hänsel and Grethel. In order to visualize the relationship between these nodes, we decided to enlarge these group nodes and place the individual nodes inside them. Since none of the coreference resolution algorithms we experimented with in this work are capable of resolving group entities to their individual characters, this part-of relation is only visualized in the gold networks but not in the predicted ones.

We tested multiple graph visualization frameworks but found their node placement unsatisfactory (for example, see Figure 8, which displays the network for *Hänsel und Grethel (015)* using the force-directed layout by GraphViz). As a solution, we devised a bespoke algorithm for node positioning in networks with a limited number of nodes and with some nodes being more important than others. The resulting graphs are then visualized using GraphViz [33]. For details, see Appendix B.

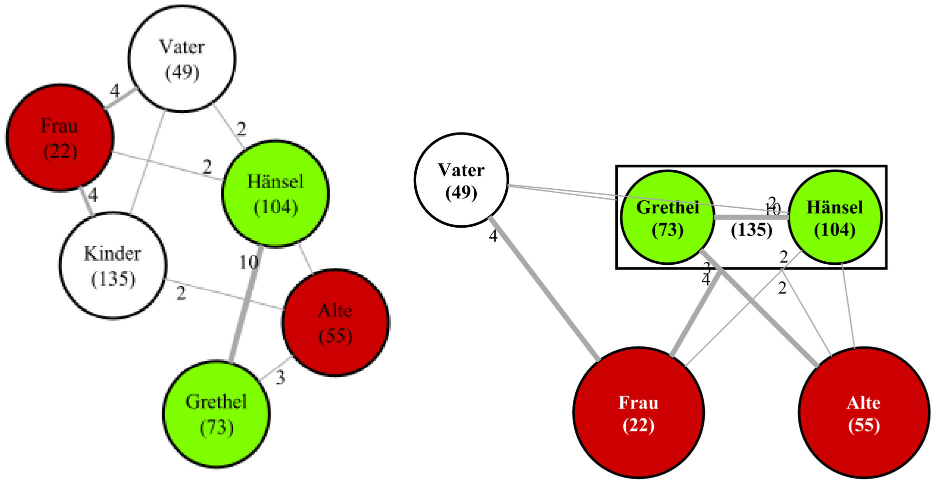


Figure 8. Visualization of the character network of the fairy tale *Hänsel und Grethel (015)* using the force-directed layout by GraphViz (**left**) and our node positioning algorithm (**right**).

Resulting Networks

In this section, we will perform a very short qualitative evaluation of some visualized networks, before we proceed to a more expansive quantitative evaluation in the next section. We present examples contrasting predicted character networks with those based on manually annotated information. This comparison, illustrated in Figure 9, encompasses three documents: *Der Froschkönig oder der eiserne Heinrich (001)*, *Der Wolf und die sieben jungen Geislein (005)*, and *Hänsel und Grethel (015)*. For coreference resolution, we employ the c2f algorithm.

- Der Froschkönig oder der eiserne Heinrich (001): The predicted network for this document aligns closely with the manually annotated network. It accurately captures the nodes, node labels and existing edges, and even correctly identifies the main characters. There are only two discrepancies: the edge between the princess (Königstochter) and the frog (Frosch) has an incorrect label (but the correct strength), and between the frog (Frosch) and the king (König), both the strength and the label of the edge are inaccurately represented.
- Der Wolf und die sieben jungen Geislein (005): In the predicted network for this tale, one notable issue arises: the miller (Müller) and the youngest child (jüngste) are entirely absent as nodes. Aside from this deviation, the rest of the network closely mirrors the manually annotated version.
- Hänsel und Grethel (015): This narrative shows more significant differences between the predicted and manually annotated networks. The predicted network correctly identifies the nodes Hänsel and Grethel as main characters (and also the group node for the children (Kinder)), but overlooks the two antagonists: the witch (labeled as Alte) and the stepmother (labeled as Frau). Additionally, it includes an extraneous node for the duck (Ente), a character not labeled in our documents due to its lack of dialogue, despite Grethel's interaction with it. The predicted network incorrectly

includes additional edges, such as those between the father (Vater) and Grethel, the stepmother (Frau) and Grethel, and between the children (Kinder) node and the individual children (Hänsel & Grethel). Notably, there are no edges missing in the predicted network. The predicted networks are also an example of networks where the positioning of the character nodes could still be improved, as the edge between father and Hänsel in the predicted network crosses the children node.

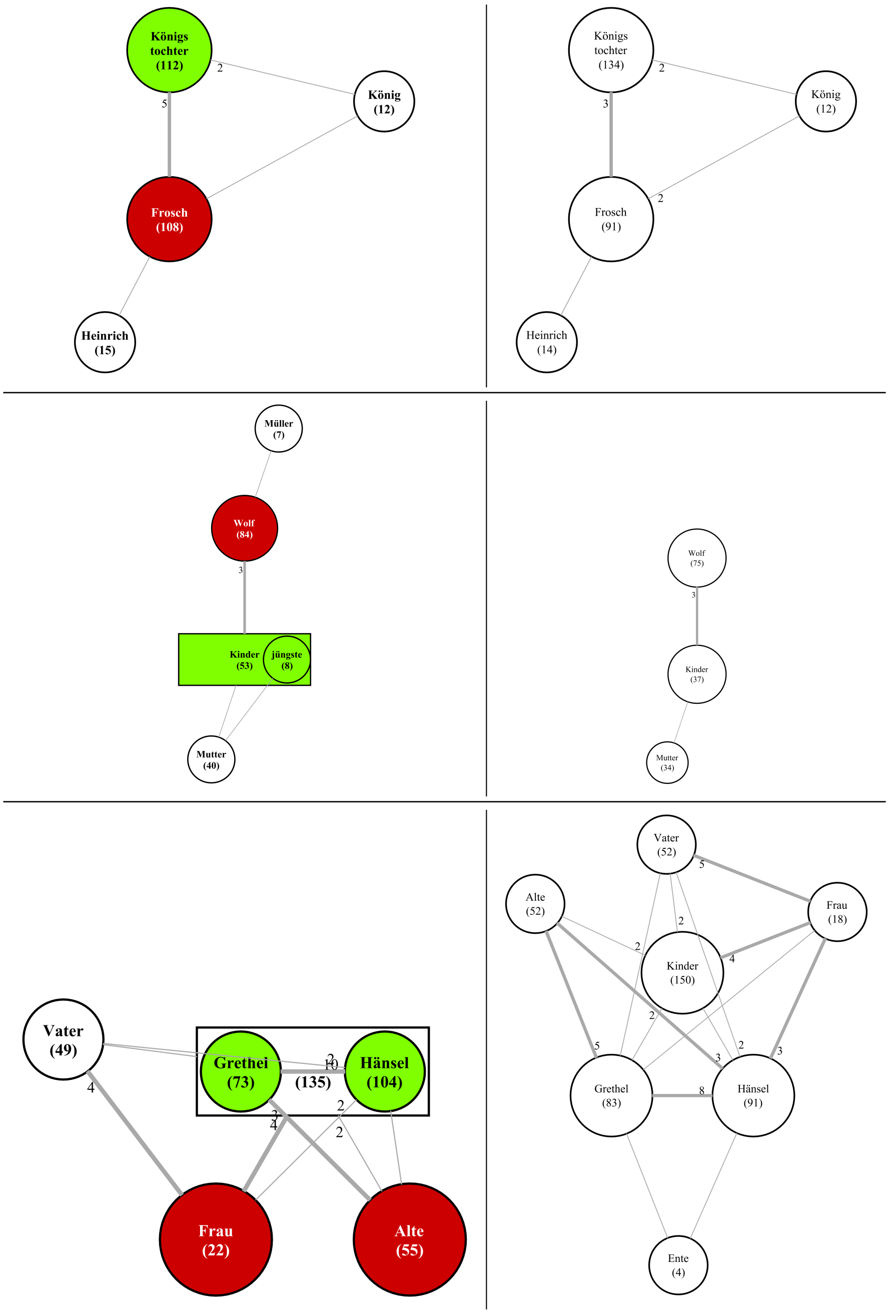


Figure 9. Visualized networks for three documents: *Der Froschkönig oder der eiserne Heinrich (001)*, *Der Wolf und die sieben jungen Geislein (005)* and *Hänsel und Grethel (015)*. On the left are the networks based on manually annotated information, and on the right are the predicted networks with c2f as a coreference resolution algorithm. Numbers in the nodes show the characters' frequency of occurrence, and numbers at the edges show the frequency of the interactions. The green color marks the hero, and the red color the antagonists.

7. Evaluation

This section deals with the evaluation of the character networks. We start with a short evaluation of the individual algorithms used for coreference resolution and speaker detection, before turning to the predicted networks. They will be evaluated in terms of their components, such as the nodes, as well as as a whole, which includes a newly developed network score that is introduced beforehand.

7.1. Coreference Resolution

We evaluate the coreference algorithms on the documents with numbers 001 to 070. For the neural networks (c2f, long-doc-coref and ASP), we split these documents into five random folds and evaluate each fold by training the algorithm on the other four folds, the documents numbered 071–130 as well as DROC [16]. Since we used ChatGPT only for coreference resolution but not for mention detection, we evaluated it twice: once with hand-annotated (gold) mentions (ChatGPT GM) and once with automatically annotated (system) mentions (ChatGPT SM). We used a rule-based algorithm for this. Using the mentions produced by one of the neural networks would be an interesting experiment as well. However, since annotation with ChatGPT is very time-consuming, we decided not to carry this out a third time. It is important to note when regarding the results of this evaluation that in contrast to those with the other algorithms, the experiments with ChatGPT were not performed in a fully automatic fashion, but involved manual steps, including checking the output for gross mistakes like continuing the given text instead of repeating it.

In light of the annotation scheme used in the CoNLL format and the jsonlines formats used by the coreference resolution algorithms, where each token is annotated with the name of the speaking entity, we opted not to supply the coreference algorithms based on neural networks with any information regarding speakers. Providing the algorithms with the speaking entities' names for each token reveals coreference information, even with automatically annotated speakers, e.g., that the first-person pronouns in two utterances are coreferent. Note that this decision had minimal impact on the results, with a 0.1-point difference in coreference metrics at most compared to that when using gold-annotated speakers and addressees. An exception was made for the rule-based Sieves algorithm, which received hand-annotated information about speakers and addressees. We do not consider this to be a problem because, as we will see below, the Sieves algorithm still performs significantly worse than its competitors in spite of this advantage.

7.1.1. Mention Detection

In Table 1, we can see the performance of the algorithms with regards to mention prediction. long-doc-coref and ASP perform the best with F1 scores of 95.8 and 95.9, while the performance of c2f is slightly worse with two fewer points, due to a lower recall. The rule-based algorithm has the worst results with an F1 score 5.5 points lower than that of c2f, mostly because of its very low recall in comparison to that of the other algorithms (7 points below that of c2f, therefore having the lowest recall among the neural networks).

Table 1. Results of the mention detection.

Model	Precision	Recall	F1
Rules	94.1	83.8	88.2
c2f	97.2	90.8	93.7
long-doc	96.8	95.0	95.8
ASP	95.7	96.1	95.9

7.1.2. Coreference Resolution

As metrics for the coreference evaluation, we used MUC [34], B^3 [35], $CEAF_E$ [36], the CoNLL-Score [37] (average of MUC, B^3 and $CEAF_E$), BLANC [38] and LEA [39]. Table 2 shows the results of the coreference algorithms on the documents with numbers 001 to 070. We can see that (unsurprisingly) the Sieves algorithm is significantly worse than all other approaches. Among the neural-network-based approaches, ASP performs the best with respect to all metrics. The results of c2f and long-doc-coref are slightly worse but mostly close to each other (at most 1 point F1); the only exception of this is the CEAF_E metric (and consequently, to a lesser degree, the CoNLL score) where long-doc-coref is about 18.5 F1 points (6.1 for CoNLL) better than c2f. ChatGPT with system mentions is on par with c2f with regards to the CoNLL score but worse with regards to BLANC and LEA. However, it is still much better than the rule-based Sieves algorithm.

Table 2. Results of the coreference resolution algorithms.

Model	**MUC**	**B^3**	**CEAF_E**	**CoNLL**	**BLANC**		**LEA**	
	F1	**F1**	**F1**	**F1**	**F1**	**P**	**R**	**F1**
Sieves	84.8	46.8	34.9	55.5	53.7	54.8	34.8	41.3
c2f	93.3	70.9	46.3	70.1	75.5	70.9	67.7	68.8
long-doc	92.7	71.2	64.8	76.2	76.2	69.4	67.3	67.8
ASP	92.9	72.9	67.9	77.9	77.9	69.5	71.5	70.1
ChatGPT SM	93.5	67.5	48.1	69.7	67.4	84.8	53.8	64.3
ChatGPT GM	95.2	84.1	62.6	80.6	86.7	89.5	76.0	81.3

From the results of the experiment with gold mentions (ChatGPT GM), we can see that ChatGPT can perform better than the neural networks when it is given perfect mentions. The difference is large (more than eight points) when looking at B^3, BLANC and LEA. With respect to CEAF_E, ChatGPT is worse than long-doc-coref and ASP, which is why it is fewer than three points better in terms of the CoNLL score.

7.2. Speaker Attribution

The speaker detection algorithm is evaluated on the documents with numbers 001 to 070 (with hand-annotated direct speech utterances), and the results of the evaluation are presented in Table 3. For the detection of speakers, the algorithm demonstrated a high level of precision and recall, with an F1 score of 92.3. In contrast, the algorithm's performance in addressee detection was significantly lower, achieving an F1 score of only 73.5.

Table 3. Results of the speaker detection algorithm. Train marks the part of the documents that were seen during the optimization of the algorithm, and test those that were not.

	Eval Level	**Data**	**Precision**	**Recall**	**F1**
Speakers	Utterance	Seen	93.2	90.6	91.9
Addressees	Utterance	Seen	80.1	71.6	75.6
Speakers	Utterance	Unseen	93.7	93.3	93.5
Addressees	Utterance	Unseen	70.7	65.1	67.8
Speakers	Utterance	All	93.3	91.3	92.3
Addressees	Utterance	All	77.5	69.9	73.5
Speakers	Dialogue	All	95.6	95.4	95.5
Addressees	Dialogue	All	80.1	73.4	76.6
Both	Dialogue	All	91.5	89.4	90.4

Comparing the documents that were not seen during the optimization of the algorithm's scores with those that were leads to some surprising results: while the attribution of speakers is better, the attribution of addressees is considerably worse, mostly due to a drop in precision of about 9.4 points. This is particularly unexpected considering the attributions of speakers and addressees influence each other during the second stage of the algorithm. A short experiment showed that the difference in scores for the attribution of addressees does not appear to be caused by overfitting on the data seen during the optimization: removing scores added during the optimization and resetting the values of the rest to their state from before the optimization lead to results that are worse by 2.5 points in terms of the F1 score.

Dialogue-Level Evaluation

The evaluation of the speaker attribution algorithm was extended to include a dialogue-level assessment, given that dialogues serve as the foundational element for character interactions. It utilized manually annotated coreference information and dialogue boundaries (internally, the speaker detection algorithm still used its predicted dialogue boundaries). For each dialogue, we identified the characters that were gold-annotated as speakers and those predicted as such. Characters appearing in both sets were classified as true positives (TPs), while those absent from the gold set were considered false positives (FPs), and those missing from the predicted set were labeled as false negatives (FNs). This process was also applied for addressee attribution. Additionally, an evaluation was conducted for the combined attribution of speakers and addressees, comparing the sets of characters involved in the dialogue in any capacity.

The attribution of speakers at the dialogue level demonstrated a marked improvement, scoring 3.2 points higher than when evaluated at the level of individual utterances. This suggests more accurate speaker identification when considering the broader context of a dialogue. The attribution of addressees at the dialogue level also showed an increase in the F1 score compared to the individual utterance level (3.1 points). The combined attribution of speakers and addressees attained an F1 score of 90.4, which is about 5 points lower than for speakers alone but much better than for addressees alone.

7.3. Character Networks

Now that the individual evaluations of the algorithms for coreference resolution and speaker detection are completed, it is time to evaluate the predicted character networks as well. Before we present the evaluation results, we will discuss how the networks and their components are evaluated.

7.3.1. Procedure

While measures such as average degree and betweenness centrality are commonly employed to assess network structures, the objective of our investigation is to ascertain the efficacy of algorithms in predicting pivotal characters and their interrelations. To this end, our evaluation procedure diverges from standard practices, and is similar to the evaluation performed by [10].

A primary issue in aligning predicted networks with gold-standard networks relates to character identification. Specifically, we must determine the correspondence between characters within the predicted network and their counterparts in the gold network. This problem is analogous to that in the task in coreference resolution and is approached through the application of the Kuhn–Munkres algorithm [40], similar to the CEAF metric for coreference resolution [36]. In our context, the algorithm utilizes character names and noun phrases as the basis for establishing similarity (names and noun phrases that are used several times for both characters are counted several times as well). Matches exhibiting zero similarity (i.e., no shared name or noun phrase) are excluded from consideration, which typically occurs when there are no actual correspondences or they have been matched to different characters.

Nodes and Edges

Before evaluating character interactions, we first assess the accuracy of the various algorithms in predicting the nodes within the networks. Our approach consists of counting the number of correctly matched nodes/characters and noting those that were missed or incorrectly added by the algorithms. From these counts, we can compute precision, recall and F1 score. However, note that this does not reflect the quality of the characters' mentions (e.g., deleting 90% of a character's mentions would still lead to a match here).

Following character mapping, the interactions within the network can be evaluated, which can be carried out at different levels. At the textual level, the first analysis can be carried out on the specific instances of character interactions, i.e., whether the algorithms correctly predict which characters are involved in each dialogue. The conditions for the prediction of an instance being correct are relatively strict: the utterances covered by the dialogue, as well as the involved characters (we chose this strict condition mostly due to its simplicity), have to match.

Subsequent to the text-level evaluation, we delve into the network level, adopting a gradation of criteria that range from the least to the most stringent:

1. Interaction Existence: The foundational step is to verify the presence of at least one interaction between corresponding characters in both the predicted and gold networks. This is visually manifested as an edge between the respective nodes in the network diagram.
2. Interaction Strength: The evaluation escalates to scrutinizing the qualitative aspect of the interactions by gauging their strength. As described in Section 5, interactions are categorized as weak or strong, and here it is assessed whether the algorithm accurately captures this dimension of the character relationships.
3. Interaction Count: The most exacting measure is the precise tally of interactions. This stringent criterion demands that the number of interactions between characters matches exactly between the predicted and gold networks.

For the evaluation of interaction instances and interaction existence, we count the number of true positives, false positives and false negatives and calculate precision, recall and F1 score. The evaluation of interaction strength and count is conducted using the accuracy metric. In order to be able to judge each of these parts of the network separately, only the elements that can actually be predicted correctly are included in the evaluation. That means, for example, that edges where one (or both) of the involved character nodes is not present in one of the networks are ignored.

Degree and Betweenness Centrality

We also adopt the approach used in [5] to measure the differences in network centrality. This involves constructing a vector for each network, where each position in the vector corresponds to the degree or betweenness centrality of a node in the network. We then compute the Euclidean distance between these vectors to quantify the differences between the predicted and gold-standard networks. Importantly, nodes without a counterpart in the other network are paired with a dummy node that has a degree and betweenness centrality of 0.

Network Score

Additionally, a score is devised for judging the quality of the complete network. To calculate this score, we first count the number of true positives (TP), false positives (FP), and false negatives (FN). Each element (node or edge) that exists in both networks is counted as a TP, regardless of its type or strength. Elements that only exist in the gold network are counted as FN, and those that only exist in the predicted network are counted as FP. The importance of nodes and edges is reflected in their assigned weight when counting TP/FP/FN. Main nodes (heroes and antagonists) receive a weight of 4, while other nodes receive a weight of 2 and edges receive a weight of 1. It is of greater importance to identify a character than to ascertain an interaction between characters. Furthermore, the potential number of edges is considerably higher than the number of nodes, which increases their influence on the overall score. With regard to the nodes, the omission of a main character, such as the hero, is a more significant error than the omission of a character that only appears in a few sentences. Edges that are connected to a node that only exists in one of the networks contribute only half of their weight to the TP/FP count as the predicted network is already punished by the TP/FP for the node.

Thus far, node types and edge strengths do not influence the score, so in order to penalize incorrect node types and edge strengths, half of the element's weight is added to both the FP and FN (e.g., a main node that exists in the predicted network but is not a main node there would receive four TPs as well as two FPs and two FNs). Precision, recall, and F1 score are then computed using the TP/FP/FN counts in the usual way.

Node Labels

Lastly, we evaluate the labels assigned to character nodes in automatically predicted character networks. This assessment is crucial as it determines the utility and recognizability of the labels, which are integral to understanding the character relationships within a narrative. An exaggerated illustration of suboptimal labeling is assigning generic labels such as "man", "woman", "boy", or "girl" to all human characters. While technically accurate, these labels lack specificity and are therefore of limited utility in a character network. A more nuanced approach is necessary, as characters often embody multiple roles or identities within a story. For instance, in the tale *Rumpelstilzchen (055)*, the hero is initially introduced as "Müllerstochter" (miller's daughter) and later becomes the queen. Similarly, siblings of heroes can be labeled as either the children of a particular character or as brothers or sisters.

Our method for evaluating character node labeling involves annotating the correct labels for characters in the network. We then compare each predicted character label against these annotations. If the label matches one of the annotated labels, or corresponds to the label of the corresponding gold character, it is considered correct. Predicted characters that cannot be matched to any gold character are excluded from this evaluation. It is important to note that the labeling in the gold networks does not always align with human intuition. For example, in scenarios with multiple sons, they may not be directly referred to as "son" (Sohn) in the narrative but rather described with a modifier like "oldest" (ältester). Consequently, the labeling algorithm might only identify the label "oldest" (ältester), while a human annotator would likely label the character as "oldest son" (ältester Sohn).

The data set for the evaluation consists once more of the documents with numbers 001 to 070.

7.3.2. Main Results

First, we look at the detection of individual instances of interactions. The results for this are shown in Table 4. An overview of these results indicates that all algorithms struggle to predict specific instances of character interactions, as evidenced by the universally very low scores in this metric. Incorrect dialogue boundaries are largely responsible for these low scores, as we will see in Section 8.6.

Table 4. Precision, recall and F1 score for the detection of individual interaction instances.

	Instance		
	P	**R**	**F1**
Sieves	12.4	14.8	13.5
c2f	22.3	25.1	23.6
long-doc	19.5	22.4	20.8
ASP	21.5	24.6	23.0
GPT SM	21.6	19.5	20.5
GPT GM	25.1	28.8	26.8

Next, we turn to the prediction of characters and interaction at the network level. There are two kinds of averages we can compute: macro-averages and micro-averages. Macro-averages compute precision, recall and F1 score for each document separately and then average them. This approach allows us to judge the quality of a complete predicted network on average. However, since all documents are weighted equally, irrespective of their size, individual errors in small networks have far more impact on the average score than those in large networks. For micro-averages, we instead add the TP, FP and FN of all documents and compute the precision, recall and F1 score from these sums. This approach enables us to judge the quality of the prediction of parts of the networks (e.g., character nodes), regardless of network size. Because both approaches have their advantages, we show them both in Table 5. For detailed results, see Appendix C, where we list counts and F1 scores for each document and coreference resolution algorithm separately.

Table 5. Evaluation results of the constructed character networks. The top half shows macro-averages, and the lower half micro-averages. Speaker/ addressee detection is always performed by the algorithm described in Section 4.2.

	Character						**Interaction**				**Network**		
	Existence			**Main**			**Existence**			**Str**	**Score**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
Sieves	70.4	74.5	70.0	57.5	65.8	56.6	61.2	65.0	60.3	66.8	60.4	65.2	61.2
c2f	91.4	83.2	86.0	83.2	76.3	74.2	70.6	92.5	78.3	85.9	78.9	77.7	77.7
long-doc	88.2	88.3	87.5	80.1	69.0	68.8	71.0	83.2	73.9	84.5	75.0	79.2	76.7
ASP	90.1	82.2	84.8	83.2	68.3	70.7	71.9	89.0	77.4	80.1	77.5	76.1	76.1
GPT SM	84.9	80.1	80.1	72.2	69.8	65.7	72.8	73.2	70.7	73.6	74.1	72.4	72.0
GPT GM	90.6	93.2	91.2	81.1	74.3	72.0	76.3	89.5	80.3	87.2	79.3	85.0	81.7
Sieves	66.0	72.3	69.0	54.4	66.7	59.9	65.1	72.2	68.5	80.7	56.6	64.5	60.3
c2f	89.6	80.7	84.9	72.3	70.4	71.4	67.1	92.6	77.8	86.7	76.0	75.7	75.9
long-doc	86.8	86.3	86.6	69.1	65.0	67.0	64.4	83.6	72.8	86.1	72.8	77.4	75.0
ASP	87.6	79.9	83.6	74.8	66.4	70.3	67.7	88.5	76.7	82.1	74.3	74.5	74.4
GPT SM	78.7	78.7	78.7	62.7	69.8	66.1	73.6	73.6	73.6	86.2	68.3	71.0	69.6
GPT GM	88.3	92.1	90.1	67.2	70.0	68.6	73.0	90.3	80.7	91.0	76.3	84.0	80.0

We can see that macro- and micro-averages are mostly within three points of each other. The only major outliers are the interaction strength of ChatGPT SM (a difference of about 13 points) and the interaction scores for the Sieves algorithm. Here, the macro F1 score for interaction existence and the macro accuracy of interaction strength are about 8 and 14 points higher, respectively. This discrepancy is not reflected in the network scores, where the difference is only 1.2 points. The following discussion is restricted to the micro-averages.

We find that algorithms using neural networks perform similarly in predicting characters, with only a 3.0-point difference in their F1 scores. Among them, long-doc-coref stands out due to its higher recall (but slightly lower precision). ChatGPT scores are about 5 F1 points lower than those for these algorithms, and the Sieves algorithm is even further behind, with an additional 10-point gap. When using manually annotated mentions, ChatGPT's performance improves significantly, surpassing that of long-doc-coref by 3.5 F1 points. This improvement is mostly due to increased recall, where the difference is more than twice as much as for precision.

The performance of the algorithm tasked with identifying the main characters, while not exemplary, is reasonably satisfactory given its simplicity. When evaluated on manually annotated networks, it achieves a precision of 76.6% and a recall of 69.1%. The combined efficiency, reflected in an F1 score of 72.6%, calls for further work in this area. An analysis shows that most of the characters who are not found by the algorithm are antagonists, due to their sometimes low number of mentions (and consequently also low interaction counts). This is especially the case for documents with several antagonists like *Der wunderliche Spielmann (008)*. The algorithm's performance on predicted networks (shown in the 'Main' column in Table 5) is worse, as anticipated, though the difference is usually less than six points.

The results of the interaction evaluation are also detailed in Table 5. While not approaching levels that could be considered satisfactory, the F1 scores for interaction existence are nevertheless significantly higher than those for instance prediction. It is noteworthy here that in most cases, the recall values exceed the precision values by more than 10 points. This implies that systems commit more errors by predicting non-existent interactions than by failing to identify actual ones. The results for the prediction of interaction strengths is better, with accuracy scores between 80.7 for the Sieves algorithm and 91.0 for ChatGPT with gold mentions.

The network scores are roughly on the level of the interaction existence scores. The Sieves algorithm has the lowest score (60.3), followed by ChatGPT with predicted mentions (about 9 points higher). The neural networks are close to each other again, and ChatGPT with gold mentions has the highest scores with 80.0 points.

It is notable that the Sieves algorithm displays markedly inferior performance in comparison to that of the algorithms based on neural networks. This outcome is consistent with its established limitations in the coreference evaluation scenario. The neural network-based approaches once again achieve similar results.

Turning our attention to ChatGPT, it demonstrates competitive performance with neural network models in predicting interaction strengths. However, its effectiveness diminishes when assessing the other aspects of the character networks. When given gold mentions, ChatGPT outperforms the neural network-based approaches at everything except the prediction of main characters. This shows great potential for an automatic system using ChatGPT for coreference resolution, assuming it is possible to automatically detect and correct the cases where it is not performing the task it is given.

7.3.3. Other Results

Table 6 presents the results for count accuracy, degree difference and betweenness centrality difference. Count accuracy is, as expected, significantly lower than strength accuracy. A surprising aspect is the relatively high score of the Sieves algorithm in comparison to the other algorithms. The results for degree and betweenness centrality difference are mostly consistent with the other results. Only the difference in betweenness centrality of ChatGPT with gold mentions is surprising, since it is higher than the differences for the systems based on neural networks.

Table 6. Evaluation results for the precise interaction count, the differences in degree and betweenness centrality, and the detected labels (i.e., if the label given to a character node is a reasonable one).

	Count	**Degree**	**BC**	**Labels**
	Acc	**Δ**	**Δ**	**Acc**
Sieves	53.8	5.15	10.55	87.1
c2f	53.4	3.14	5.73	91.5
long-doc	56.7	3.43	5.87	86.7
ASP	52.9	3.16	6.83	86.3
GPT SM	50.5	3.39	8.21	87.2
GPT GM	65.2	2.76	7.72	93.6

The last part of the evaluation regards the labels applied to the character nodes. All algorithms achieved high accuracy, with scores of 86% and above. Notably, the c2f algorithm and ChatGPT with gold mentions outperformed others, achieving scores of 92% and 94%, respectively.

8. Ablation Studies/Error Analysis

As we have seen, coreference resolution plays a pivotal role in the construction and accuracy of character networks in narrative texts. The quality of these networks is heavily contingent on the precision of coreference resolution algorithms. In this section, we delve into a comprehensive analysis of how various errors in coreference resolution impact the quality of the resulting character networks. We also analyze how the prediction of dialogue segments within the narrative affects the prediction of interactions between characters. Lastly, given the observed discrepancies in the identification of speakers and addressees, we also evaluate the potential impact of excluding addressee information when building character networks. This aspect of the study aims to determine whether disregarding addressee data might lead to more accurate representations of character networks.

It is important to note that all tables in this section use micro-averages and show the differences in scores compared to those in the default evaluation seen in Table 5, unless stated otherwise. Positive values in the tables of this section indicate an improvement over the baseline results, while negative values mean the changes lead to a worse result.

8.1. Missing and Added Characters

The most apparent errors in these networks are the presence of characters that appear exclusively in one version of the network—characters that are either missing in the predicted network or are erroneously added by the algorithms. The causes for these discrepancies can be varied. One common issue is the improper merging of characters by the coreference resolution algorithm. In such cases, parts of one character might be merged into another, or conversely, parts of a character might be incorrectly split into a separate entity. Another frequent source of error is the misattribution of a character as the speaker or addressee in direct speech utterances, or a combination with errors made by the coreference resolution algorithm. In this part of the analysis, we aim to quantify the impact of these missing and added characters on the overall accuracy of the character networks. Our approach involves a focused evaluation where these anomalous characters are excluded from the comparison between predicted and gold-standard networks.

The results of this analysis can be seen in Table 7. First, note that due to our evaluation procedure, this experiment had no influence on the interaction in the network. The scores for the character nodes are unremarkable as precision and/or recall increased to 100 as anticipated. The F1 scores for character existence as well as the network scores show a high correlation with the increases in precision and recall. It is therefore not surprising that the Sieves algorithm has the overall highest increase in network F1 score, followed by ChatGPT SM, while ChatGPT with gold mentions has the lowest increase.

Table 7. Differences in scores compared to Table 5 when missing (top part), added (middle part) and both missing and added (bottom) characters are ignored during the evaluation.

	Character						**Interaction**				**Network**		
	Existence			**Main**			**Existence**			**Str**	**Score**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
Sieves	0.0	27.7	10.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	17.0	6.5
c2f	0.0	19.3	9.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	13.2	6.1
long-doc	0.0	13.7	6.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.8	3.9
ASP	0.0	20.1	9.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	12.8	5.9
GPT SM	0.0	21.3	9.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	12.7	5.7
GPT GM	0.0	7.9	3.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5	2.0
Sieves	34.0	0.0	14.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.1	0.0	11.0
c2f	10.4	0.0	4.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.5	0.0	3.0
long-doc	13.2	0.0	6.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.3	0.0	4.2
ASP	12.4	0.0	5.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.7	0.0	3.7
GPT SM	21.3	0.0	9.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.4	0.0	7.2
GPT GM	11.7	0.0	5.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.6	0.0	4.0
Sieves	34.0	27.7	31.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.1	17.0	20.3
c2f	10.4	19.3	15.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.5	13.2	9.7
long-doc	13.2	13.7	13.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.3	8.8	8.6
ASP	12.4	20.1	16.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.7	12.8	10.2
GPT SM	21.3	21.3	21.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.4	12.7	14.1
GPT GM	11.7	7.9	9.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.6	4.5	6.1

8.2. Merge Errors

As highlighted in the previous section, coreference resolution algorithms can (among other errors) erroneously split a single character into multiple entities or merge distinct characters into a single entity. To measure the effects of these merge errors, we employ a methodology that leverages manually annotated documents in order to try to automatically correct these mistakes. The results from the corrected documents are then compared with the original, unmodified results. The automatic fixes work as follows:

- In cases where two or more separate characters are merged into one, we split this aggregated entity into two (or more, theoretically) separate characters. The criterion for this split is as follows: each of the gold-standard characters must constitute at least one-third of the mentions within the erroneously merged character.
- When the algorithm incorrectly divides a single character into several separate entities, we address this by merging these fragmented entities back into a unified character. This process involves identifying all the automatically generated characters that correspond to the same gold-standard character and merging them with one another. The criterion for this is based on mention overlap: a generated character is identified as part of the gold-standard character with which it shares the highest number of mentions.

It is important to note that these automatic fixes are not perfect solutions but are implemented as a means to partially rectify the errors made by the coreference resolution algorithms. While they may not comprehensively resolve all inaccuracies, these fixes provide a basis for better understanding the extent and impact of merge errors in character networks.

For incorrectly merged characters, the results displayed in Table 8 might be unexpected, especially the large decrease in scores for the detection of character nodes, as we tried to improve results with this experiment. This improvement is reflected in an increase in recall for the detected character nodes (although it is very small for ChatGPT). The decrease in precision is much larger, however, so there is an overall decrease in F1 score. This decrease means that a lot of the character nodes created by this experiment cannot be matched to characters in the gold nodes because they have already been matched to an existing (better) character (alternatively, a new node may take the place of a previously matched node, leaving this node without a match as a consequence). This leads to the conclusion that the algorithms only rarely merge two characters completely, but instead often merge part of a character into another. Overall, ChatGPT is the least affected by these modifications, and all other systems show similar changes.

In contrast, the results for characters that have been split align more closely with expectations. All algorithms demonstrate an increase in precision and F1 score for the character nodes, interaction existence and the network score. The Sieves algorithm shows the most considerable improvement, with an increase in the network F1 score by 10.1 points. The results of interaction strength are somewhat surprising as the accuracy of the Sieves algorithm, c2f and long-doc decreases by up to 2.5 points.

Combining both modifications—merging split characters and splitting merged characters—yields some intriguing results. The F1 score for character existence decreases by up to 11.4 points. On the other hand, the scores for the detection of main characters as well as interaction existence and interaction strength improve for the most part. With the exception of the Sieves algorithm (+1.9 points) and ChatGPT SM (+0.1 points), these increases cannot compensate for the decrease in character existence, so the network F1 scores decrease by up to 2.6 points.

Table 8. The top shows the difference compared to the automatically constructed networks when incorrectly merged characters are split; the middle when split characters are merged; and the bottom when characters are both split and merged.

	Character						**Interaction**				**Network**		
	Existence			**Main**			**Existence**			**Str**	**Score**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
Sieves	−37.0	8.4	−26.3	3.4	3.1	3.3	4.8	−9.5	−2.4	4.1	−24.6	4.7	−16.5
c2f	−47.5	12.9	−26.8	0.4	1.7	1.0	4.5	−2.1	2.1	3.6	−28.6	9.6	−15.0
long-doc	−45.5	7.3	−29.3	−3.3	−2.4	−2.8	7.4	−5.1	2.2	2.8	−27.2	3.8	−16.6
ASP	−45.1	12.3	−25.4	−5.4	3.5	−0.7	5.6	−2.5	2.5	7.7	−26.9	8.7	−14.0
GPT SM	−30.0	1.1	−18.2	1.8	−2.8	−0.4	1.3	−0.8	0.2	0.6	−16.6	0.5	−9.6
GPT GM	−29.3	0.8	−17.9	2.8	0.0	1.4	0.1	−2.5	−0.9	−0.1	−15.2	0.3	−9.1
Sieves	27.4	−6.8	8.0	12.6	11.4	12.2	7.1	13.5	9.9	−2.5	22.8	−1.3	10.1
c2f	9.1	−2.1	2.6	−1.0	1.5	0.2	0.7	0.5	0.7	−1.9	6.0	−1.6	1.9
long-doc	10.0	−5.3	1.6	5.6	−0.1	2.5	4.5	7.5	5.7	−0.3	8.1	−2.2	3.0
ASP	9.1	−2.2	2.6	2.3	−0.9	0.5	0.2	4.1	1.6	0.9	6.0	−1.0	2.4
GPT SM	17.5	−0.3	7.7	6.5	−1.9	2.5	1.6	14.5	7.5	2.2	13.6	2.8	8.1
GPT GM	8.0	−0.3	3.9	5.2	0.0	2.6	0.0	4.8	1.9	0.4	5.8	1.4	3.7
Sieves	−12.7	3.0	−6.6	13.2	2.3	8.4	9.4	3.2	6.4	7.1	0.0	3.9	1.6
c2f	−22.1	9.9	−7.6	−0.2	2.9	1.3	4.4	−3.8	1.4	3.3	−9.0	7.2	−1.8
long-doc	−21.1	1.6	−11.4	5.4	0.8	2.9	9.6	5.3	8.0	4.6	−6.6	3.1	−2.4
ASP	−23.4	8.6	−9.2	2.0	5.3	3.8	6.9	1.7	4.9	8.6	−8.9	7.6	−1.6
GPT SM	−15.2	1.1	−8.0	6.2	−0.9	2.8	2.7	14.0	8.0	3.3	−3.2	3.8	0.1
GPT GM	−17.9	0.5	−10.1	4.6	0.0	2.3	0.0	3.2	1.3	−0.3	−7.1	1.5	−3.5

8.3. Main Characters

In exploring the character networks of narrative texts, a key question arises: do algorithms yield better results when restricted to main characters? Given that main characters, such as heroes and antagonists, often appear more frequently in narratives, it is plausible to assume that their interactions could be more reliably predicted. However, this increased frequency also presents more opportunities for the coreference resolution algorithms to err. The analysis, as reflected in the results presented in Table 9, aims to explore this aspect by evaluating networks limited to these main characters.

Table 9. Difference in scores when the networks are restricted to the main characters. In the top half, the characters in the predicted networks only include those that are mapped to a main character (hero or antagonist) in the manually annotated networks; in the bottom half, they only include the characters that have been predicted to be main characters.

	Character						**Interaction**				**Network**		
	Existence			**Main**			**Existence**			**Str**	**Score**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
Sieves	34.0	15.7	24.6	45.6	3.2	22.4	23.5	−1.7	10.0	−16.2	29.2	10.5	19.7
c2f	10.4	13.5	12.1	27.7	−3.7	8.6	21.8	−0.3	12.8	−5.5	9.2	5.7	7.4
long-doc	13.2	10.2	11.6	30.9	0.1	11.8	19.8	10.5	16.1	−2.8	11.5	5.2	8.4
ASP	12.4	18.9	15.8	25.2	−2.9	7.4	19.6	−2.8	9.8	−5.0	9.6	8.4	9.0
GPT SM	21.3	14.9	18.0	37.3	−2.7	14.2	20.2	−3.8	6.4	−16.2	17.2	7.6	12.3
GPT GM	11.7	6.7	9.3	32.8	−2.9	11.7	15.5	6.1	11.6	−0.3	9.4	1.8	5.8
Sieves	−31.7	15.7	−19.7	45.6	7.3	25.1	24.4	5.1	14.4	−16.0	−17.7	12.2	−8.7
c2f	−47.2	13.5	−26.4	27.7	−3.7	8.6	21.8	−0.3	12.8	−5.5	−27.3	5.7	−14.9
long-doc	−44.7	10.2	−27.9	30.9	0.1	11.8	19.8	10.5	16.1	−2.8	−25.2	5.2	−14.6
ASP	−42.1	18.9	−21.3	25.2	−1.7	8.3	18.5	0.8	11.0	−6.1	−23.5	9.1	−11.2
GPT SM	−37.0	14.9	−21.0	37.3	1.4	17.1	14.6	−3.8	4.3	−16.2	−22.4	8.8	−11.3
GPT GM	−45.8	6.7	−30.7	32.8	0.6	14.2	15.5	6.1	11.6	−0.3	−28.3	3.0	−18.1

Restricting the characters in the networks using manually annotated information is straightforward, given that heroes and antagonists are annotated. However, applying similar restrictions to the predicted networks poses challenges. There are two primary methods considered for this purpose:

1. Restricting Characters Mapped to a Main Character in the Gold Network: This approach leads to overly optimistic results. The accuracy appears artificially high because the process virtually eliminates the presence of incorrect nodes in the network.
2. Identifying Main Characters in the Predicted Networks Using an Algorithm: This method incorporates additional errors stemming from the algorithm tasked with identifying the main characters. It offers a more realistic assessment but is affected by the inherent inaccuracies of the additional algorithm.

Given the limitations and distinct natures of both methods, the decision is made to conduct experiments using each approach.

As expected, both character and interaction existence scores show an increase in the first experiment. This is likely due to the reduction in incorrectly identified characters in the network. Contrary to that, the scores for character existence decrease by more than 20 points in the second experiment. This is not unexpected, given that this method also suffers from the errors made by the algorithm for identifying main characters. In both experiments, the detection of main characters is improved, while the strength accuracy is decreased.

Overall, the findings suggest that coreference resolution algorithms might be more effective at predicting the existence of interactions between main characters (at least for characters that have been found), but less so at predicting the existence of the nodes and the interactions' strength.

8.4. Significant Changes of Characters

One common narrative device involves characters disguising themselves and assuming alternate identities, often being referred to by different names. This poses a considerable challenge for coreference resolution algorithms, as the algorithms must recognize that the differently named entities in the narrative are, in fact, the same character. This challenge is further amplified in stories like *König Drosselbart (052)*, where the revelation that two characters are the same individual is only made towards the end of the narrative. Transformations present a similar challenge. These include scenarios where characters shift forms, such as a human turning into an animal or an object, or vice versa. These transformations often lead to a change in how characters are referred to within the text, complicating the task of maintaining consistent character identity across the narrative. Additionally, changes in social status, particularly through marriage, also pose a unique challenge. For instance, when a girl or young woman marries into royalty and becomes a queen, the narrative shift in addressing her can disrupt the continuity of character identity as perceived by coreference algorithms.

In this subsection, our focus is to evaluate the extent to which these kinds of significant character changes impede the accuracy of coreference resolution algorithms—and, as a consequence, the quality of the resulting character networks. To this end, we restrict the data set to only contain the documents that feature such character metamorphoses (disguise: 4 documents; transformation: 12 documents; marriage: 10 documents; all types: 22 documents (some documents contain several types of changes)). We aim to assess whether treating disguised, transformed, or newly titled characters as separate entities yields predicted character networks that are more congruent with the gold-standard networks.

The results, as shown in Table 10, vary considerably across the different coreference resolution algorithms. In the case of disguises, long-doc-coref and ASP encounter a decrease in almost all scores. In contrast, the Sieves algorithm shows improvements in most scores. The scores for the detection of main characters are notable in that only ChatGPT with predicted mentions shows any changes at all. With regards to the network score, the Sieves algorithm and ChatGPT SM show an improvement, while the others show a decline.

Table 10. Differences in scores when various types of significant changes in a character lead to a new entity. From top to bottom: disguise, transformation, marriage, and all three. The baseline scores used for these comparisons were calculated exclusively on the documents that contain the corresponding type of change.

	Character						**Interaction**				**Network**		
	Existence			**Main**			**Existence**			**Str**	**Score**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
Sieves	5.1	−4.1	2.2	16.7	−16.7	7.4	9.9	15.8	12.5	7.3	6.3	2.0	5.2
c2f	3.7	−7.6	−1.9	0.0	−12.2	−8.3	−3.6	−0.8	−2.6	−6.8	−0.4	−9.9	−4.8
long-doc	0.0	−11.0	−6.1	0.0	−6.7	−4.4	−9.1	1.1	−4.2	−14.9	−3.7	−12.6	−8.6
ASP	8.7	−3.6	1.6	0.0	−8.0	−8.2	−10.2	3.9	−4.7	−5.0	0.4	−6.8	−3.3
GPT SM	8.6	−2.3	4.5	7.1	−16.3	−1.6	−2.3	−2.2	−2.2	7.7	4.5	−2.6	1.7
GPT GM	6.2	−6.3	1.0	0.0	−10.1	−7.9	−8.4	3.0	−2.8	−4.8	0.8	−8.7	−3.4
Sieves	2.3	−2.5	−0.1	0.0	−4.6	−1.6	−1.6	4.8	1.5	−2.4	1.0	−2.0	−0.5
c2f	1.4	−4.4	−2.0	−4.2	−7.5	−5.8	−3.5	1.5	−2.1	−0.3	−1.3	−5.7	−3.7
long-doc	1.2	−5.0	−2.0	5.9	0.2	2.1	0.5	9.5	3.7	2.3	1.0	−2.9	−0.9
ASP	1.4	−4.0	−1.9	−5.9	−4.5	−5.1	−0.9	7.8	1.8	−4.9	−0.5	−4.8	−2.9
GPT SM	3.6	−1.9	0.8	1.4	−4.1	−0.7	−2.4	3.7	0.6	0.0	1.9	−1.8	0.0
GPT GM	3.4	−2.9	0.4	1.8	−0.4	0.6	−3.1	2.8	−1.1	0.1	1.6	−2.9	−0.4
Sieves	3.0	−3.4	0.3	0.0	−10.9	−5.8	−9.6	2.7	−3.8	−2.9	0.3	−4.1	−1.5
c2f	2.7	−4.9	−1.7	0.0	−6.5	−3.8	−7.3	−3.4	−6.1	−0.5	−1.0	−7.4	−4.3
long-doc	5.0	−3.3	0.7	1.8	−4.1	−1.6	−9.6	−5.5	−8.4	−5.7	0.7	−5.9	−2.5
ASP	4.3	−3.7	−0.5	0.0	−9.2	−6.5	−6.7	−0.1	−4.8	−5.5	−1.0	−5.7	−3.5
GPT SM	5.7	−1.5	2.3	6.3	3.5	4.9	−12.9	−0.6	−6.2	−5.4	3.7	−2.0	1.0
GPT GM	3.5	−5.0	−0.7	3.1	0.0	1.6	−2.9	4.7	0.2	−1.0	2.4	−4.8	−0.9
Sieves	3.8	−3.4	0.7	4.4	−10.7	−1.4	−2.7	6.6	1.2	−1.7	2.0	−2.6	0.1
c2f	2.2	−6.2	−2.7	−2.5	−8.8	−6.0	−6.5	−2.3	−5.4	−1.9	−1.7	−8.8	−5.3
long-doc	3.2	−5.8	−1.5	4.1	−2.5	−0.2	−5.9	1.3	−3.3	−4.6	0.5	−6.4	−2.9
ASP	3.7	−4.8	−1.3	−3.3	−8.2	−7.0	−5.4	3.3	−2.7	−5.5	−0.8	−6.9	−4.0
GPT SM	5.8	−2.3	2.0	4.6	−5.0	0.6	−6.3	1.1	−2.6	−1.4	3.2	−2.8	0.5
GPT GM	4.7	−4.6	0.3	2.4	−2.8	−0.3	−3.3	4.3	−0.3	−1.3	2.5	−5.0	−0.9

When it comes to transformations, all algorithms except ChatGPT achieve worse results with respect to the character nodes. The differences in other scores are more varied. Overall, most algorithms show only small changes with regards to the network score (0.6 points or less); only c2f and ASP have slightly higher decreases of 3.7 and 2.9 points, respectively.

Regarding marriages, we can see small changes to the scores for character existence, and larger changes for the other scores. Most of these changes are negative. The network scores decrease for almost all algorithms as well.

When evaluating all character changes together, most changes in scores are negative. The network F1 score of c2f decreases the most (by 5.3 points), and the Sieves algorithm and ChatGPT SM are the only algorithms that have an increase, even if only small (0.1 and 0.5 points).

Overall, we can conclude that the kinds of character changes we discuss in this section mostly cause no or only small problems for the algorithms. The only times the network score showed an algorithm clearly suffering from character changes were when the Sieves algorithm and ChatGPT SM handled disguises. On the other hand, the network scores did not decrease much in most cases, suggesting that the algorithms do not yet handle these types of character changes perfectly.

8.5. Addressees

In the section about the speaker detection algorithm (Section 4.2), we observed a significant disparity in the detection accuracy for addressees compared to that for speakers, with the latter being more accurately identified (92.3%) than the former (73.5%). As characters addressed in a dialogue often also serve as speakers within the same dialogue, a plausible modification to the interaction detection method might be to only recognize interactions when both characters appear as speakers in the dialogue (i.e., no longer using addressees).

Table 11 shows the results of this experiment. There is no impact on the detection of the character nodes, which is to be expected. In contrast, interaction strength accuracy sees improvements, and interaction existence sees decreases across the board. The detection of main characters is worse as well for the most part.

Table 11. Comparison of the networks where addressees are not counted for interactions with the default networks.

	Character						**Interaction**				**Network**		
	Existence			**Main**			**Existence**			**Str**	**Score**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
Sieves	0.0	0.0	0.0	1.7	0.9	1.4	−8.0	−1.1	−5.1	7.6	−0.4	0.9	0.2
c2f	0.0	0.0	0.0	−2.7	−2.6	−2.7	−5.9	0.8	−3.9	0.7	−1.0	−0.1	−0.6
long-doc	0.0	0.0	0.0	−3.3	2.5	−0.3	−5.9	−2.5	−4.8	0.8	−0.5	0.0	−0.2
ASP	0.0	0.0	0.0	−7.5	−0.9	−3.9	−5.4	1.2	−3.2	4.7	−0.4	0.0	−0.2
GPT SM	0.0	0.0	0.0	−0.5	4.7	1.7	−8.4	3.4	−3.0	6.2	0.0	1.5	0.7
GPT GM	0.0	0.0	0.0	−0.8	2.5	0.7	−8.7	−0.7	−5.8	0.7	−1.1	0.2	−0.6

In summary, excluding addressees from the character network construction leads to less accurate predictions regarding the existence of edges between characters. However, this omission appears to enhance the accuracy of the networks in terms of the strength of these interactions. For the most part, both changes cancel each other out in the network scores, which change by less than one point.

8.6. Dialogue Boundaries

The delineation of dialogue boundaries (i.e., which direct speech utterances belong to the same dialogue) plays a pivotal role in shaping the interaction networks within our systems. These boundaries, if inadequately defined, can significantly influence the network dynamics. Specifically, overly narrow dialogue boundaries may intensify the frequency of interactions, thereby strengthening network edges. However, this could also result in the omission of some edges. Conversely, excessively broad boundaries may lead to the inclusion of extraneous edges within the network.

To investigate the impact of these predicted dialogue boundaries on the networks, we conducted an experiment where networks were constructed based on manually annotated dialogue boundaries. It is important to note that the speaker attribution algorithm employed in our study continued to utilize its own predicted dialogues. This experimental setup allows us to precisely assess the influence of boundary prediction on the accuracy of the generated interaction networks.

Since there was no change to the character nodes, Table 12 displays the changes in scores for the detection of individual interaction instances instead. One of the most striking results is the substantial increase in scores for the detection of interaction instances, with improvements of up to 43 points observed. The exception here is the Sieves algorithm, which only improved by 21 points. This remarkable improvement clearly indicates that the failure to accurately predict the precise dialogue boundaries was the primary reason for the very low interaction instance scores in the original evaluation. Comparing these results with those of the coreference evaluation in Table 2, we can see that the increases roughly correlate with the coreference scores: the higher the coreference scores, the more the algorithms benefit from correct dialogue boundaries.

In addition to the detection of interaction instances, existence F1 scores also show improvements, albeit to a lesser extent. The highest increase is noted for ChatGPT with gold mentions, while ChatGPT SM shows the smallest improvement. The exception is once again the Sieves algorithm, which shows a decrease here. A notable aspect of this improvement is that the increase in F1 scores for interaction existence is predominantly due to a major increase in precision, with gains of 10 points or more, while recall, interestingly, actually decreases. In contrast to the previously mentioned metrics, strength accuracy decreases for all algorithms. The network score shows improvements, even though they are below two points for most algorithms.

Table 12. Score differences of networks where dialogue boundaries are gold-annotated (compared to the networks built with default settings).

	Interaction							Network		
	Instance			Existence			Str	Score		
	P	R	F1	P	R	F1	Acc	P	R	F1
Sieves	21.0	21.1	21.1	7.2	−7.4	−0.2	−3.5	2.2	−1.0	0.8
c2f	34.2	33.6	34.0	10.8	−4.5	4.9	−2.0	3.3	−0.8	1.1
long-doc	33.1	35.0	34.1	12.0	−2.5	5.9	−0.9	3.8	−0.3	1.8
ASP	33.9	35.9	34.8	9.6	−2.3	4.8	−0.4	3.4	−0.3	1.5
GPT SM	30.3	25.3	27.6	13.1	−4.5	3.3	−0.3	2.1	−0.4	0.9
GPT GM	42.9	43.4	43.2	18.8	−3.3	8.6	−0.7	5.0	0.0	2.6

Overall, these results underscore the importance of accurately delineating dialogue boundaries for parts of the constructed networks. However, the network scores suggest that their influence on the networks as a whole is rather limited—at least with the weights we use currently.

8.7. Influence of Individual Components

In this section, we will examine how the character networks are improved when we gradually replace the results of the individual components used for predicting the networks (e.g., coreference resolution) with gold information. The results of this evaluation can be seen in Table 13. In contrast to the preceding tables, this one does not indicate differences from the baseline evaluation; rather, it shows the achieved scores themselves. This is because our interest also lies in the extent to which the experiments fall short of achieving perfect scores. For comparison with the baseline experiments, we include the results of the coreference resolution models based on neural networks.

Using gold coreference information unsurprisingly gives a large boost, mostly to the detection of the character nodes. On the other hand, adding gold speakers makes hardly any difference. Gold addressees lead to a slightly larger but still small increase for interaction existence scores. Dialogue boundaries cause a much larger increase than speakers and addressees. This was indicated by the results of the previous section, where better coreference scores correlated with larger increases due to better dialogue boundaries. While the scores for character existence and interactions are perfect at this point, the network score is still at only a 92.9-point F1 score due to the errors caused by the algorithm for main character detection. A lot of these errors come from antagonists that were not predicted to be main characters, as we can see from the improved scores when all antagonists are predicted as main characters using gold information. Note that we cannot easily correct all errors related to antagonists because the algorithm is designed to predict main characters in general, without differentiating between heroes and antagonists (i.e., false negatives can be corrected, while false positives cannot).

We find that the order in which gold information is added has some influence on the results of this evaluation. Flipping the order of speakers, addressees and dialogue boundaries leads to lower improvements due to the boundaries and larger improvements caused by speakers and addressees. An exception to this is the antagonists, for which the quality of speaker and addressee attribution as well as dialogue boundary detection has almost no impact. Overall, coreference resolution offers the most room for improvement, followed by the detection of antagonists and dialogue boundaries.

Table 13. Evaluation results when we incrementally replace the results of individual components with gold information. In contrast to the other tables in this section, this table does not show differences to the baseline evaluation.

	Character						Interaction				Network		
	Existence			Main			Existence			Str	Score		
	P	R	F1	P	R	F1	P	R	F1	Acc	P	R	F1
c2f	89.6	80.7	84.9	72.3	70.4	71.4	67.1	92.6	77.8	86.7	76.0	75.7	75.9
long-doc	86.8	86.3	86.6	69.1	65.0	67.0	64.4	83.6	72.8	86.1	72.8	77.4	75.0
ASP	87.6	79.9	83.6	74.8	66.4	70.3	67.7	88.5	76.7	82.1	74.3	74.5	74.4
Gold CR	100.0	100.0	100.0	73.7	70.7	72.2	70.2	94.4	80.5	91.2	83.8	90.6	87.1
+ Speakers	100.0	100.0	100.0	73.3	71.5	72.4	69.3	96.5	80.6	92.0	83.5	91.2	87.2
+ Addressees	100.0	100.0	100.0	73.0	72.4	72.7	70.3	98.2	81.9	91.6	83.8	91.6	87.5
+ Dialogues	100.0	100.0	100.0	76.6	69.1	72.6	100.0	100.0	100.0	100.0	92.9	92.9	92.9
+ Antagonists	100.0	100.0	100.0	82.2	97.6	89.2	100.0	100.0	100.0	100.0	97.7	97.7	97.7
Gold CR	100.0	100.0	100.0	73.7	70.7	72.2	70.2	94.4	80.5	91.2	83.8	90.6	87.1
+ Dialogues	100.0	100.0	100.0	73.9	71.5	72.7	89.9	94.1	91.9	93.4	89.9	90.9	90.4
+ Speakers	100.0	100.0	100.0	73.9	71.5	72.7	90.6	96.5	93.4	94.8	90.2	91.6	90.9
+ Addressees	100.0	100.0	100.0	76.6	69.1	72.6	100.0	100.0	100.0	100.0	92.9	92.9	92.9
Gold CR	100.0	100.0	100.0	73.7	70.7	72.2	70.2	94.4	80.5	91.2	83.8	90.6	87.1
+ Antagonists	100.0	100.0	100.0	79.6	98.4	88.0	70.2	94.4	80.5	91.2	87.6	95.1	91.2

9. Discussion and Conclusions

In this research, we explored the feasibility of automatically constructing character networks from dialogues within German fairy tales by predicting these networks and subsequently assessing their alignment with networks built from gold-annotated data. The evaluation reveals satisfactory results in predicting character nodes, though it also highlights a need for improvement in predicting interactions. This is primarily attributed to the limitations in the coreference resolution and speaker attribution processes. It is noteworthy that none of the coreference resolution algorithms demonstrated clearly superior performance in comparison to the others. In their current state, the predicted networks could be useful for a quick overview over the tales' characters, as long as one keeps their shortcomings in mind (e.g., unreliable identification of main characters). For an analysis of the fairy tales, the predicted networks are likely not good enough yet in order to draw reliable conclusions in most cases. It is important to note, however, that the current state of the networks is just an intermediate step on the way towards more complex networks (see the paragraph about future work below), and the added information will make the networks more reliable overall (some early experiments with large language models showed very promising results for relation extraction and sentiment analysis).

In Section 8, we were able to gain some interesting insights. For example, the coreference resolution algorithms seem to have no particular problems with significant character changes like transformations from human into animal or vice versa. The low quality of addressee attribution does not appear to degrade the quality of the character networks, as predicted networks do not align more closely to their hand-annotated counterparts when both do not count addressees as participants in a dialogue/interaction. Additionally, while the currently predicted dialogue boundaries lead to abysmal scores for the detection of individual instances of interactions between characters, they only have a minor impact on the overall network score. This changes when the quality of the coreference resolution algorithms improves drastically. It can be concluded that coreference resolution is the most important area for improvement. After this has significantly improved, better results in the detection of the dialogue boundaries can have a large impact on the networks' quality. Meanwhile, further improvements to the attribution of speakers will have a minor impact in general, and improvements to addressee attribution will only be beneficial if all other components provide high-quality results.

Looking ahead, several avenues for future work emerge, offering potential enhancements to the current framework:

- Incorporation of Character Attributes and Properties: Future research could include detailed attributes of characters, such as their form (human, animal, etc.), their state (alive or dead), their age (growing up, grown-up, old) or their financial situation (rich or poor). Notably, the prediction of gender, based on morphological information provided by the RFTagger [41], is highly reliable for characters in these fairy tales. However, the visualization of gender in the networks may be deemed superfluous, given the ease of inferring it from character names.
- Analysis of Character Relations: Exploring the nature of relationships between characters, such as familial ties (e.g., parent–child) or the sentiment underlying their interactions (positive, neutral, or negative), could provide deeper insights into the narrative structure. First experiments using large language models (ChatGPT 4o) showed promising results for the main characters.
- Dynamic Network Evolution: Tracking the evolution of character networks across the narrative, particularly from scene to scene, presents an intriguing direction. However, the challenging nature of automatic scene detection, as can be seen in [42,43] for example, suggests that this might not be feasible without significant advancements in scene detection algorithms and might have to be performed manually for now.
- Resolving Group Membership and Speech Attribution: A notable challenge lies in identifying group members and attributing group dialogues to individual characters. Presently, groups may appear as separate entities alongside individual members in the networks, without clear indications of their interconnectedness. For instance, in the network for *Hänsel und Grethel (015)* (Figure A1), the node "Kinder" represents the group comprising Hänsel and Grethel, but does not explicitly connect to the individual characters.

Enhancing the methodology for the automatic construction of character networks necessitates not only the expansion of network elements but also the refinement of existing predictive algorithms. The detection of dialogues, a critical component for attributing speakers and addressees, presents an area ripe for improvement. The initial design, focused on the performance of the attribution process, might have overlooked the precision of boundary predictions, suggesting potential for more nuanced detection mechanisms. An error analysis shows that the majority of errors is caused by the system merging utterances into the same dialogue. This might be improved by adding more keywords like verbs of movement, though this would also lead to more errors of the opposite type (the problem with most keywords is that one can find example sentences where the keyword does not actually signify the beginning of a new dialogue). We do not see a straightforward way to combine these keywords in an effective way, so the most promising approach seems to be training a neural network to decide whether two utterances belong to the same dialogue, since it can more easily take all of the context into consideration instead of just a few keywords.

Furthermore, while this study explored certain coreference resolution algorithms, there are alternative algorithms that score better on benchmarks like Ontonotes but have not been examined within this context. Despite the close performance of neural-network-based algorithms in predicting networks, incorporating algorithms that perform better in established benchmarks could offer marginal improvements. A simple way to improve the performance of the coreference resolution algorithms (and thus the quality of the predicted networks) further, at least in theory, is simply exchanging the language models they use for a larger version. However, due to hardware restrictions, this is not something we can explore at the moment. An ensemble approach, leveraging the strengths of three or more existing algorithms, might enhance coreference resolution performance even further.

A pivotal area requiring attention is the prediction of main characters, particularly antagonists. The accurate identification of these characters is fundamental to understanding narrative dynamics. Incorporating sentiment analysis, especially the prediction of negative sentiments and the detection of negative interactions with the hero, could significantly improve character prediction. Given that most characters interact with the hero, focusing on the nature of these interactions—particularly negative ones—may provide critical insights into character roles and relationships within the narrative.

The observation that certain characters, despite their non-verbal but significant presence in the narrative, are absent in the visualized networks, underscores a limitation in the current approach. This is particularly noticeable in fairy tales such as *Marienkind (003)* and *Die zwölf Brüder (009)*, where pivotal characters like the parents of the hero or the king who marries the hero remain unrepresented due to their lack of dialogue. To address this, a potential area for future work involves expanding the scope of character inclusion within these networks. This expansion would not solely rely on characters' participation in dialogues but also consider their relational importance to the main characters (a brief analysis suggests that at least half of them could be recovered that way and an exploratory evaluation of an algorithm for relation extraction by Krug [10] leads us to believe that this is also possible in the predicted networks). Such an approach could encompass a broader spectrum of narrative elements, offering a more comprehensive view of the story's social dynamics and opportunities for more and deeper literary analysis—especially when combined with the other extensions of the networks that we have explained above.

Author Contributions: Conceptualization, F.P.; methodology, D.S.; software, D.S.; validation, D.S.; formal analysis, D.S.; investigation, D.S.; resources, F.P.; data curation, D.S.; writing—original draft preparation, D.S.; writing—review and editing, F.P. and D.S.; visualization, D.S.; supervision, F.P.; project administration, F.P.; funding acquisition, F.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in our institute's Gitlab repository at https://gitlab.informatik.uni-wuerzburg.de/kallimachos/extraction-of-interaction-networks (accessed on 24 October 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Fairy Tale Titles

Table A1 lists the fairy tales that were used in this work. For each fairy tale, it shows the fairy tales's number as well as its German and English title (according to https://en.wikisource.org/wiki/Grimm%27s_Household_Tales,_Volume_1, accessed on 24 October 2024).

Table A1. The numbers and the German and English titles of the fairy tales that were used in this work.

#	German Title	English Title
001	Der Froschkönig oder der eiserne Heinrich	The Frog-King, or Iron Henry
002	Katze und Maus in Gesellschaft	Cat and Mouse in Partnership
003	Marienkind	Our Lady's Child
004	Mährchen von einem der auszog das Fürchten zu lernen	The Story of the Youth who went forth to learn what Fear was
005	Der Wolf und die sieben jungen Geislein	The Wolf and the Seven Little Kids
006	Der treue Johannes	Faithful John
007	Der gute Handel	The Good Bargain
008	Der wunderliche Spielmann	The Wonderful Musician
009	Die zwölf Brüder	The Twelve Brothers
010	Das Lumpengesindel	The Pack of Ragamuffins
011	Brüderchen und Schwesterchen	Brother and Sister
012	Rapunzel	Rapunzel
013	Die drei Männlein im Walde	The Three Little Men in the Wood
014	Die drei Spinnerinnen	The Three Spinners
015	Hänsel und Grethel	Hänsel and Grethel
016	Die drei Schlangenblätter	The Three Snake-Leaves
017	Die weiße Schlange	The White Snake
018	Strohhalm Kohle und Bohne	The Straw, the Coal, and the Bean
020	Das tapfere Schneiderlein	The Valiant Little Tailor
021	Aschenputtel	Cinderella

Table A1. *Cont.*

#	German Title	English Title
022	Das Räthsel	The Riddle
024	Frau Holle	Mother Holle
025	Die sieben Raben	The Seven Ravens
026	Rothkäppchen	Little Red-Cap
027	Die Bremer Stadtmusikanten	The Bremen Town-Musicians
028	Der singende Knochen	The Singing Bone
029	Der Teufel mit den drei goldenen Haaren	The Devil with the Three Golden Hairs
030	Läuschen und Flöhchen	The Louse and the Flea
031	Das Mädchen ohne Hände	The Girl without Hands
032	Der gescheidte Hans	Clever Hans
033	Die drei Sprachen	The Three Languages
034	Die kluge Else	Clever Elsie
035	Der Schneider im Himmel	The Tailor in Heaven
036	Tischchen deck dich Goldesel und Knüppel aus dem Sack	The Wishing-Table, the Gold-Ass, and the Cudgel in the Sack
037	Daumesdick	Thumbling
038	Die Hochzeit der Frau Füchsin	The Wedding of Mrs. Fox
039	Die Wichtelmänner	The Elves
040	Der Räuberbräutigam	The Robber Bridegroom
041	Herr Korbes	Herr Korbes
042	Der Herr Gevatter	The Godfather
043	Frau Trude	Frau Trude
044	Der Gevatter Tod	Godfather Death
046	Fitchers Vogel	Fitcher's Bird
048	Der alte Sultan	Old Sultan
049	Die sechs Schwäne	The Six Swans
050	Dornröschen	Little Briar-Rose
051	Fundevogel	Fundevogel
052	König Drosselbart	King Thrushbeard
053	Sneewittchen	Little Snow-White
054	Der Ranzen das Hütlein und das Hörnlein	The Knapsack, the Hat, and the Horn
055	Rumpelstilzchen	Rumpelstiltskin
056	Der Liebste Roland	Sweetheart Roland
057	Der goldene Vogel	The Golden Bird
058	Der Hund und der Sperling	The Dog and the Sparrow
059	Der Frieder und das Catherlieschen	Frederick and Catherine
061	Das Bürle	The Little Peasant
063	Die drei Federn	The Three Feathers
064	Die goldene Gans	The Golden Goose
065	Allerleirauh	Allerleirauh
069	Jorinde und Joringel	Jorinda and Joringel
070	Die drei Glückskinder	The Three Sons of Fortune

Appendix B. Visualization

Appendix B.1. Node Positioning Algorithm

Central characters, namely the heroes and antagonists, are anchored uniformly on an inner circle. Their positions are fixed, establishing a structured core within the visualization. Peripheral characters, by contrast, are arranged on an outer circle. Their exact positions are determined using a force-directed placement strategy. Simplifying the model to a one-dimensional space, the movement of these nodes is confined to a circular path with potential positions ranging from 0 to 2π radians. A node reaching the endpoint of 2π seamlessly transitions back to the starting point at 0, and the other way around. Handling the forces between nodes is simplified by this circular representation, since they can only have two directions: clockwise or counter-clockwise. Additionally, the influence of the inner nodes on the outer nodes can easily be modeled by treating them as if they were at the same position on the outer circle.

The interactive dynamics between nodes are governed by two types of forces: repulsive forces act between all nodes to avoid overlap and ensure clarity, while attractive forces exist between directly connected nodes to better visually encapsulate relationships. Notably, the inner circle nodes only exert attractive but not repulsive forces on their outer circle counterparts.

Our algorithm is sequential in its application of forces. Initially, only attractive forces from the inner to the outer nodes are considered. This negates the necessity to search for ideal starting points, as the nodes converge at nearly identical positions regardless of their initial location. If the outer node in the algorithm is inadvertently positioned diametrically opposite its optimal location, this may cause issues like edges unnecessarily crossing inner nodes. To address this problem, the algorithm assesses whether relocating the node to the opposite position would result in a better layout, and repositions the node if necessary. An example of this is given in Figure A1, with the left image showing the position of the node Kinder after the application of the attracting forces from the inner nodes, and the right image showing the optimal position it is moved to.

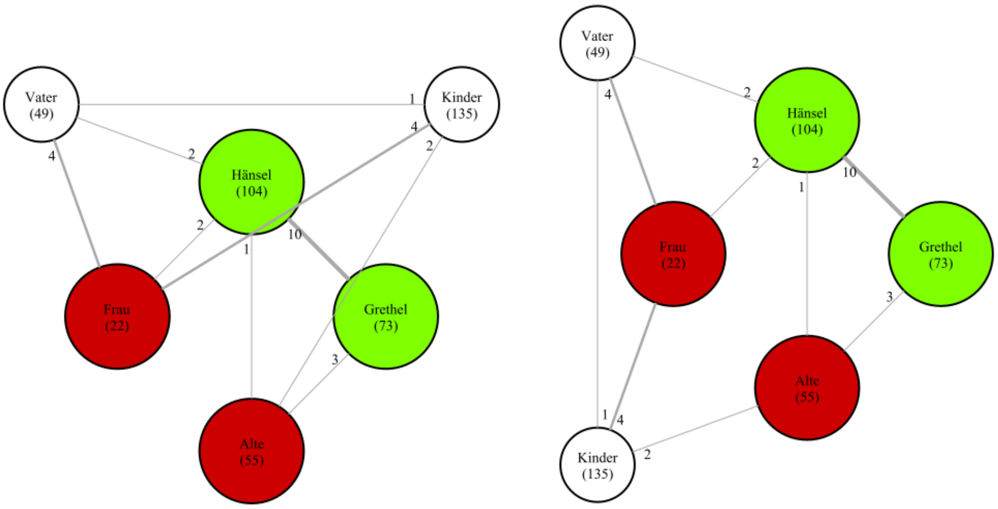


Figure A1. An example from *Hänsel und Grethel (015)*, where the first part of the node positioning algorithm has moved the node Kinder to a position (**left** image) that is diametrically opposite to its optimal location (**right** image).

Subsequently, attractive forces are introduced between outer nodes, pulling together nodes linked by an edge while minimizing the presence of unrelated nodes between them. In the last stage, repulsive forces are integrated to balance the network spatially, thus improving the visualization's clarity.

An additional refinement is performed for cases where edges from outer nodes to other (inner or outer) nodes intersect with inner nodes. The algorithm examines small positional adjustments of the outer nodes to avoid such overlaps. The adjustment is closely controlled to prevent nodes from encroaching on their neighbors, thus preserving the integrity of the layout.

Plural Nodes

The node positioning algorithm described above has no means to enforce a close proximity of group nodes and individual nodes, so we let it handle these cases as a single node instead. This means that the individual nodes are merged into the group node (including the edges to other nodes) before the node positioning algorithm is applied, and split into separate nodes afterwards. Note that this procedure is only carried out for group nodes when the individual nodes are also part of the network, but not for networks where only the group node appears, e.g., the sisters (Schwestern) in Figure 1.

Appendix C. Detailed Results

Tables A2–A7 show detailed results for each algorithm and document. See Table A1 for the fairy tale titles that correspond to each number. Note that for main characters and interaction strength, we counted TPs/FPs/FNs/TNs with respect to the labels "main" or "strong", so a TN is a support character or a weak label that is identified correctly (as there are only two labels in both cases). It is also important to note that the TP/FP/FN counts for the network score are doubled compared to the weights given in Section 7.3.1 due to the data structures in our implementation (which use integers, not doubles).

Table A2. Detailed results for the Sieves algorithm.

					Character										**Interaction**												
		Existence					**Main**				**Existence**					**Strength**				**Count**				**Network** **Score**			
	TP	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**TP**	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**corr**	**inc**	**Acc**	**TP**	**FP**	**FN**	**P**	**R**	**F1**
001	4	2	0	0.80	2	1	0	1	0.80	6	0	2	0.86	0	2	2	2	0.00	0	6	0.00	30	19	6	0.61	0.83	0.71
002	2	0	0	1.00	2	0	0	0	1.00	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	18	0	0	1.00	1.00	1.00
003	5	2	0	0.83	2	0	0	3	1.00	2	0	4	0.50	2	0	0	0	1.00	0	2	0.00	30	11	4	0.73	0.88	0.80
004	11	4	4	0.73	1	0	0	10	1.00	24	12	0	0.80	2	0	0	22	1.00	22	2	0.92	72	39	21	0.65	0.77	0.71
005	4	4	1	0.62	2	1	0	1	0.80	6	2	0	0.86	0	0	2	4	0.00	4	2	0.67	30	25	8	0.55	0.79	0.65
006	4	2	5	0.53	1	1	0	2	0.67	6	0	2	0.86	2	0	2	2	0.67	2	4	0.33	26	14	30	0.65	0.46	0.54
007	4	2	3	0.62	1	0	1	2	0.67	8	4	0	0.80	2	2	2	2	0.50	0	8	0.00	32	27	21	0.54	0.60	0.57
008	3	0	2	0.75	1	0	2	0	0.50	4	2	0	0.80	0	0	0	4	0.00	2	2	0.50	28	10	22	0.74	0.56	0.64
009	6	3	2	0.71	0	3	1	2	0.00	12	0	0	1.00	0	2	0	10	0.00	8	4	0.67	40	30	24	0.57	0.62	0.60
010	1	1	4	0.29	0	1	0	0	0.00	0	0	0	0.00	0	0	0	0	0.00	0	1	0.00	4	10	29	0.29	0.12	0.17
011	5	8	4	0.45	1	1	2	1	0.40	4	2	6	0.50	2	0	2	0	0.67	0	4	0.00	36	60	39	0.38	0.48	0.42
012	4	0	1	0.89	1	0	1	2	0.67	4	2	2	0.67	2	0	2	0	0.67	2	2	0.50	28	7	13	0.80	0.68	0.74
013	6	4	4	0.60	2	2	1	1	0.57	4	2	12	0.36	0	0	2	2	0.00	2	2	0.50	40	41	45	0.49	0.47	0.48
014	5	3	3	0.62	0	1	1	3	0.00	4	2	4	0.57	0	0	2	2	0.00	0	4	0.00	28	25	26	0.53	0.52	0.52
015	6	2	0	0.86	2	0	2	2	0.67	18	8	0	0.82	4	2	4	8	0.57	6	12	0.33	58	33	11	0.64	0.84	0.72
016	3	5	0	0.55	1	1	1	0	0.50	2	0	4	0.50	0	0	0	2	0.00	2	0	1.00	22	42	10	0.34	0.69	0.46
017	5	3	2	0.67	1	0	1	3	0.67	4	0	4	0.67	0	0	0	4	0.00	4	0	1.00	32	26	18	0.55	0.64	0.59
020	5	5	4	0.53	1	0	0	4	1.00	6	4	0	0.75	2	0	0	4	1.00	4	2	0.67	30	34	23	0.47	0.57	0.51
021	6	5	1	0.67	2	3	0	1	0.57	8	8	6	0.53	2	0	2	4	0.67	6	2	0.75	40	56	22	0.42	0.65	0.51
022	6	2	0	0.86	1	0	1	4	0.67	2	4	6	0.29	0	0	0	2	0.00	2	0	1.00	34	18	10	0.65	0.77	0.71
024	4	1	3	0.67	1	0	2	1	0.50	2	4	4	0.33	0	0	0	2	0.00	2	0	1.00	30	17	28	0.64	0.52	0.57
025	2	1	3	0.50	0	1	0	1	0.00	0	2	0	0.00	0	0	0	0	0.00	0	1	0.00	8	10	20	0.44	0.29	0.35
026	4	0	2	0.80	2	1	0	1	0.80	6	4	2	0.67	2	0	0	4	1.00	2	4	0.33	30	6	14	0.83	0.68	0.75
027	3	1	2	0.67	2	0	0	1	1.00	4	0	0	1.00	0	0	0	4	0.00	4	0	1.00	24	6	18	0.80	0.57	0.67
028	3	1	2	0.67	0	1	1	1	0.00	0	4	0	0.00	0	0	0	0	0.00	0	1	0.00	16	14	21	0.53	0.43	0.48
029	8	2	2	0.80	2	5	0	1	0.44	10	18	4	0.48	0	0	2	8	0.00	4	6	0.40	50	46	25	0.52	0.67	0.58
030	2	0	6	0.40	0	1	0	1	0.00	0	2	0	0.00	0	0	0	0	0.00	0	1	0.00	8	4	33	0.67	0.20	0.30
031	9	4	1	0.78	1	2	1	5	0.40	22	8	2	0.81	0	2	2	18	0.00	10	12	0.45	66	49	17	0.57	0.80	0.67
032	2	0	1	0.80	1	1	0	0	0.67	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	14	2	7	0.88	0.67	0.76
033	3	3	0	0.67	1	2	0	0	0.50	2	0	2	0.67	0	0	0	2	0.00	2	0	1.00	18	23	6	0.44	0.75	0.55
034	6	2	0	0.86	1	0	0	5	1.00	18	2	0	0.95	2	0	2	14	0.67	16	2	0.89	46	18	1	0.72	0.98	0.83
035	3	0	0	1.00	1	0	0	2	1.00	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	22	0	0	1.00	1.00	1.00
036	11	4	0	0.85	4	2	1	4	0.73	16	14	14	0.53	2	0	4	10	0.50	6	10	0.38	80	66	24	0.55	0.77	0.64
037	8	3	2	0.76	1	0	0	7	1.00	10	6	2	0.71	2	0	0	8	1.00	8	2	0.80	46	27	14	0.63	0.77	0.69
038-1	3	1	0	0.86	1	1	0	1	0.67	4	0	0	1.00	0	0	0	4	0.00	2	2	0.50	20	6	2	0.77	0.91	0.83
038-2	3	0	0	1.00	1	2	0	0	0.50	4	2	0	0.80	0	0	0	4	0.00	4	0	1.00	20	6	4	0.77	0.83	0.80
039-1	2	0	1	0.80	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	2	6	0.88	0.70	0.78
040	5	1	1	0.83	0	1	2	2	0.00	8	4	0	0.80	0	2	0	6	0.00	4	4	0.50	36	20	18	0.64	0.67	0.65
042	2	1	2	0.57	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	0	2	0.00	14	6	12	0.70	0.54	0.61
043	2	0	1	0.80	1	1	0	0	0.67	0	0	0	0.00	0	0	0	0	0.00	0	1	0.00	12	2	12	0.86	0.50	0.63

Table A2. *Cont.*

	Character									Interaction												Network					
	Existence				Main					Existence				Strength					Count			Score					
	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
044	5	2	0	0.83	2	2	0	1	0.67	8	6	0	0.73	0	0	2	6	0.00	4	4	0.50	36	20	5	0.64	0.88	0.74
046	2	4	2	0.40	1	0	1	0	0.67	0	0	2	0.00	0	0	0	0	0.00	0	1	0.00	16	29	16	0.36	0.50	0.42
048	4	2	1	0.73	1	0	1	2	0.67	4	4	4	0.50	0	0	2	2	0.00	0	4	0.00	28	24	13	0.54	0.68	0.60
049	6	1	1	0.86	0	1	2	3	0.00	4	4	8	0.40	0	0	0	4	0.00	4	0	1.00	36	19	23	0.65	0.61	0.63
050	3	4	2	0.50	1	1	0	1	0.67	0	0	2	0.00	0	0	0	0	0.00	0	1	0.00	16	41	17	0.28	0.48	0.36
051	4	1	1	0.80	3	1	0	0	0.86	4	6	0	0.57	2	0	0	2	1.00	0	4	0.00	32	19	7	0.63	0.82	0.71
052	3	6	0	0.50	2	1	0	0	0.80	2	0	4	0.50	0	0	2	0	0.00	0	2	0.00	22	44	7	0.33	0.76	0.46
053	7	8	2	0.58	2	3	0	2	0.57	6	12	8	0.38	2	0	2	2	0.67	2	4	0.33	42	76	26	0.36	0.62	0.45
054	6	4	4	0.60	0	1	3	2	0.00	0	0	10	0.00	0	0	0	0	0.00	0	1	0.00	36	40	41	0.47	0.47	0.47
055	5	3	0	0.77	1	0	1	3	0.67	10	0	0	1.00	2	0	0	8	1.00	8	2	0.80	38	22	4	0.63	0.90	0.75
056	5	3	0	0.77	0	1	2	2	0.00	6	2	0	0.86	0	0	0	6	0.00	4	2	0.67	34	26	10	0.57	0.77	0.65
057	5	2	2	0.71	1	2	1	1	0.40	6	4	0	0.75	0	0	2	4	0.00	4	2	0.67	34	36	23	0.49	0.60	0.54
058	3	1	1	0.75	2	0	0	1	1.00	6	0	0	1.00	2	0	0	4	1.00	4	2	0.67	26	5	5	0.84	0.84	0.84
059	5	3	1	0.71	1	1	0	3	0.67	6	8	0	0.60	2	2	0	2	0.67	2	4	0.33	30	28	8	0.52	0.79	0.62
061	7	2	2	0.78	1	0	1	5	0.67	14	4	2	0.82	2	2	2	8	0.50	8	6	0.57	50	23	18	0.68	0.74	0.71
063	3	2	1	0.67	1	2	0	0	0.50	2	2	0	0.67	0	0	2	0	0.00	0	2	0.00	18	18	14	0.50	0.56	0.53
064	5	3	5	0.56	1	0	0	4	1.00	6	2	0	0.86	0	0	2	4	0.00	4	2	0.67	30	17	28	0.64	0.52	0.57
065	4	2	3	0.62	0	1	1	2	0.00	4	2	2	0.67	0	0	0	4	0.00	2	2	0.50	24	17	25	0.59	0.49	0.53
069	3	0	0	1.00	2	1	0	0	0.80	0	0	4	0.00	0	0	0	0	0.00	0	1	0.00	20	2	6	0.91	0.77	0.83
070	1	2	5	0.22	0	1	0	0	0.00	0	0	0	0.00	0	0	0	0	0.00	0	1	0.00	4	18	31	0.18	0.11	0.14

Table A3. Detailed results for c2f.

	Character									Interaction												Network					
Doc	Existence				Main					Existence				Strength					Count			Score					
#	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
001	4	0	0	1.00	2	0	0	2	1.00	8	0	0	1.00	2	0	0	6	1.00	4	4	0.50	32	0	0	1.00	1.00	1.00
002	2	0	0	1.00	2	0	0	0	1.00	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	18	0	0	1.00	1.00	1.00
003	3	1	2	0.67	1	0	1	1	0.67	2	2	2	0.50	2	0	0	0	1.00	0	2	0.00	22	10	15	0.69	0.59	0.64
004	11	0	4	0.85	1	0	0	10	1.00	22	14	0	0.76	2	4	0	16	0.50	8	14	0.36	70	16	24	0.81	0.74	0.78
005	3	0	2	0.75	2	0	0	1	1.00	4	0	0	1.00	2	0	0	2	1.00	4	0	1.00	24	0	10	1.00	0.71	0.83
006	7	0	2	0.88	1	0	0	6	1.00	8	6	0	0.73	0	2	0	6	0.00	6	2	0.75	40	7	14	0.85	0.74	0.79
007	6	1	1	0.86	1	0	1	4	0.67	12	0	0	1.00	4	0	0	8	1.00	6	6	0.50	44	11	9	0.80	0.83	0.81
008	4	0	1	0.89	1	0	3	0	0.40	6	2	2	0.75	0	0	0	6	0.00	2	4	0.33	38	14	18	0.73	0.68	0.70
009	7	1	1	0.88	1	3	1	2	0.33	12	2	0	0.92	0	0	0	12	0.00	8	4	0.67	48	23	15	0.68	0.76	0.72
010	5	0	0	1.00	2	2	0	1	0.67	6	4	0	0.75	0	0	0	6	0.00	4	2	0.67	34	8	4	0.81	0.89	0.85
011	6	1	3	0.75	2	0	1	3	0.80	8	2	2	0.80	2	0	2	4	0.67	2	6	0.25	44	13	25	0.77	0.64	0.70
012	5	0	0	1.00	2	1	0	2	0.80	10	0	0	1.00	4	0	0	6	1.00	6	4	0.60	38	2	2	0.95	0.95	0.95
013	8	1	2	0.84	1	0	1	6	0.67	22	10	0	0.81	2	2	0	18	0.67	16	6	0.73	62	20	22	0.76	0.74	0.75

Table A3. *Cont.*

	Character									Interaction													Network				
Doc	Existence				Main					Existence				Strength					Count			Score					
#	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
014	7	0	1	0.93	1	2	0	4	0.50	12	8	0	0.75	0	2	2	8	0.00	8	4	0.67	44	14	11	0.76	0.80	0.78
015	6	1	0	0.92	2	1	2	1	0.57	18	8	0	0.82	8	4	0	6	0.80	6	12	0.33	58	26	12	0.69	0.83	0.75
016	3	3	0	0.67	2	0	0	1	1.00	6	0	0	1.00	0	2	0	4	0.00	2	4	0.33	26	15	1	0.63	0.96	0.76
017	6	1	1	0.86	1	0	1	4	0.67	10	6	0	0.77	0	0	0	10	0.00	4	6	0.40	42	15	9	0.74	0.82	0.78
018	3	0	0	1.00	3	0	0	0	1.00	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	30	0	0	1.00	1.00	1.00
020	7	1	2	0.82	1	0	0	6	1.00	6	8	0	0.60	2	0	0	4	1.00	4	2	0.67	38	13	15	0.75	0.72	0.73
021	6	1	1	0.86	1	0	2	3	0.50	10	8	4	0.63	2	2	2	4	0.50	2	8	0.20	46	24	19	0.66	0.71	0.68
022	5	0	1	0.91	1	1	0	3	0.67	6	4	0	0.75	0	2	0	4	0.00	4	2	0.67	30	7	12	0.81	0.71	0.76
024	5	0	2	0.83	1	0	1	3	0.67	6	6	0	0.67	0	0	2	4	0.00	4	2	0.67	34	11	21	0.76	0.62	0.68
025	4	0	1	0.89	1	0	0	3	1.00	2	2	0	0.67	0	0	0	2	0.00	0	2	0.00	22	2	5	0.92	0.81	0.86
026	4	0	2	0.80	2	1	0	1	0.80	8	4	0	0.80	2	2	0	4	0.67	4	4	0.50	32	7	13	0.82	0.71	0.76
027	3	0	2	0.75	1	0	1	1	0.67	2	2	0	0.67	0	0	0	2	0.00	2	0	1.00	22	6	23	0.79	0.49	0.60
028	4	0	1	0.89	1	0	0	3	1.00	2	4	2	0.40	0	0	0	2	0.00	2	0	1.00	22	4	11	0.85	0.67	0.75
029	8	0	2	0.89	2	3	0	3	0.57	14	14	0	0.67	2	0	0	12	1.00	4	10	0.29	54	20	16	0.73	0.77	0.75
030	3	0	5	0.55	0	3	0	0	0.00	2	4	0	0.50	0	0	0	2	0.00	2	0	1.00	14	10	32	0.58	0.30	0.40
031	9	1	1	0.90	1	0	1	7	0.67	24	6	0	0.89	2	2	0	20	0.67	18	6	0.75	68	19	10	0.78	0.87	0.82
032	3	0	0	1.00	1	0	0	2	1.00	4	2	0	0.80	0	0	4	0	0.00	0	4	0.00	20	4	2	0.83	0.91	0.87
033	2	0	1	0.80	1	0	0	1	1.00	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	14	0	5	1.00	0.74	0.85
034	4	0	2	0.80	1	0	0	3	1.00	10	0	0	1.00	2	4	0	4	0.50	2	8	0.20	30	2	14	0.94	0.68	0.79
035	3	0	0	1.00	1	0	0	2	1.00	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	22	0	0	1.00	1.00	1.00
036	8	1	3	0.80	1	0	4	3	0.33	20	10	4	0.74	2	4	2	12	0.40	10	10	0.50	72	35	38	0.67	0.65	0.66
037	8	1	2	0.84	1	1	0	6	0.67	12	8	2	0.71	2	0	0	10	1.00	6	6	0.50	48	17	15	0.74	0.76	0.75
038-1	3	0	0	1.00	1	0	0	2	1.00	4	2	0	0.80	0	0	0	4	0.00	4	0	1.00	20	2	0	0.91	1.00	0.95
038-2	3	0	0	1.00	1	2	0	0	0.50	4	2	0	0.80	0	0	0	4	0.00	4	0	1.00	20	6	4	0.77	0.83	0.80
039-1	2	0	1	0.80	1	0	0	1	1.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	0	4	1.00	0.78	0.88
040	5	1	1	0.83	1	1	1	2	0.50	8	0	0	1.00	0	2	0	6	0.00	4	4	0.50	36	13	14	0.73	0.72	0.73
041	2	0	0	1.00	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	2	2	0.88	0.88	0.88
042	2	1	2	0.57	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	8	12	0.64	0.54	0.58
043	3	0	0	1.00	2	0	0	1	1.00	4	0	0	1.00	0	0	0	4	0.00	4	0	1.00	24	0	0	1.00	1.00	1.00
044	3	1	2	0.67	1	0	1	1	0.67	2	2	0	0.67	2	0	0	0	1.00	2	0	1.00	22	11	15	0.67	0.59	0.63
046	2	2	2	0.50	1	0	1	0	0.67	2	0	0	1.00	0	0	2	0	0.00	0	2	0.00	18	22	15	0.45	0.55	0.49
048	5	0	0	1.00	2	0	0	3	1.00	4	4	4	0.50	0	0	2	2	0.00	0	4	0.00	32	5	5	0.86	0.86	0.86
049	6	0	1	0.92	1	1	1	3	0.50	8	4	0	0.80	0	0	0	8	0.00	6	2	0.75	40	10	13	0.80	0.75	0.78
050	5	1	0	0.91	1	0	1	3	0.67	4	8	0	0.50	0	0	0	4	0.00	4	0	1.00	32	20	4	0.62	0.89	0.73
051	5	0	0	1.00	3	1	0	1	0.86	6	6	0	0.67	2	0	0	4	1.00	0	6	0.00	38	8	2	0.83	0.95	0.88
052	3	2	0	0.75	1	0	1	1	0.67	4	0	2	0.80	2	0	0	2	1.00	0	4	0.00	24	15	6	0.62	0.80	0.70
053	7	4	2	0.70	2	0	0	5	1.00	14	8	2	0.74	6	0	0	8	1.00	14	0	1.00	50	35	12	0.59	0.81	0.68
054	7	1	3	0.78	1	0	2	4	0.50	8	8	0	0.67	0	0	0	8	0.00	6	2	0.75	48	20	22	0.71	0.69	0.70
055	5	0	0	1.00	2	0	0	3	1.00	10	2	0	0.91	2	0	0	8	1.00	8	2	0.80	38	2	0	0.95	1.00	0.97
056	4	1	1	0.80	1	0	1	2	0.67	4	4	0	0.67	0	0	0	4	0.00	0	4	0.00	28	13	9	0.68	0.76	0.72

Table A3. *Cont.*

					Character										Interaction										Network		
Doc		**Existence**					**Main**				**Existence**					**Strength**				**Count**					**Score**		
#	**TP**	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**TP**	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**corr**	**inc**	**Acc**	**TP**	**FP**	**FN**	**P**	**R**	**F1**
057	5	0	2	0.83	1	0	1	3	0.67	4	8	2	0.44	2	0	0	2	1.00	2	2	0.50	32	12	20	0.73	0.62	0.67
058	4	0	0	1.00	1	0	1	2	0.67	8	0	0	1.00	2	0	2	4	0.67	2	6	0.25	32	5	5	0.86	0.86	0.86
059	6	3	0	0.80	1	0	0	5	1.00	8	4	0	0.80	2	2	0	4	0.67	2	6	0.25	36	25	1	0.59	0.97	0.73
061	6	2	3	0.71	1	0	1	4	0.67	8	4	4	0.67	4	0	0	4	1.00	4	4	0.50	40	20	24	0.67	0.62	0.65
063	4	0	0	1.00	2	2	0	0	0.67	4	0	0	1.00	0	0	2	2	0.00	0	4	0.00	28	5	5	0.85	0.85	0.85
064	7	1	3	0.78	1	0	1	5	0.67	6	2	0	0.86	2	0	0	4	1.00	4	2	0.67	42	12	19	0.78	0.69	0.73
065	6	0	1	0.92	1	1	0	4	0.67	10	6	0	0.77	2	2	0	6	0.67	4	6	0.40	38	9	10	0.81	0.79	0.80
069	3	0	0	1.00	2	1	0	0	0.80	0	0	4	0.00	0	0	0	0	0.00	0	1	0.00	20	2	6	0.91	0.77	0.83
070	6	0	0	1.00	1	2	1	2	0.40	2	2	0	0.67	0	0	0	2	0.00	0	2	0.00	34	10	8	0.77	0.81	0.79

Table A4. Detailed results for long-doc-coref.

					Character										Interaction										Network		
		Existence					**Main**				**Existence**					**Strength**				**Count**					**Score**		
	TP	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**TP**	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**corr**	**inc**	**Acc**	**TP**	**FP**	**FN**	**P**	**R**	**F1**
001	4	2	0	0.80	1	0	1	2	0.67	4	0	4	0.67	0	0	2	2	0.00	2	2	0.50	28	16	9	0.64	0.76	0.69
002	2	0	0	1.00	2	0	0	0	1.00	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	18	0	0	1.00	1.00	1.00
003	5	2	0	0.83	2	0	0	3	1.00	2	0	4	0.50	2	0	0	0	1.00	0	2	0.00	30	14	4	0.68	0.88	0.77
004	13	2	2	0.87	1	0	0	12	1.00	26	10	2	0.81	2	4	0	20	0.50	12	14	0.46	82	27	15	0.75	0.85	0.80
005	3	1	2	0.67	2	0	0	1	1.00	2	2	2	0.50	2	0	0	0	1.00	2	0	1.00	22	7	12	0.76	0.65	0.70
006	9	0	0	1.00	0	1	1	7	0.00	12	16	6	0.52	0	2	2	8	0.00	8	4	0.67	52	24	14	0.68	0.79	0.73
007	6	1	1	0.86	1	0	1	4	0.67	12	0	0	1.00	4	0	0	8	1.00	6	6	0.50	44	11	9	0.80	0.83	0.81
008	4	0	1	0.89	1	0	3	0	0.40	8	2	0	0.89	0	0	0	8	0.00	4	4	0.50	40	14	16	0.74	0.71	0.73
009	7	0	1	0.93	1	2	0	4	0.50	14	2	0	0.93	0	2	0	12	0.00	10	4	0.71	46	7	13	0.87	0.78	0.82
010	4	0	1	0.89	1	1	0	2	0.67	4	2	0	0.80	0	0	0	4	0.00	2	2	0.50	24	4	11	0.86	0.69	0.76
011	6	1	3	0.75	2	0	1	3	0.80	8	6	2	0.67	4	0	0	4	1.00	4	4	0.50	44	16	24	0.73	0.65	0.69
012	5	1	0	0.91	2	3	0	0	0.57	10	0	0	1.00	2	0	2	6	0.67	8	2	0.80	38	17	7	0.69	0.84	0.76
013	9	1	1	0.90	1	0	1	7	0.67	24	10	2	0.80	2	2	0	20	0.67	18	6	0.75	68	22	18	0.76	0.79	0.77
014	6	0	2	0.86	1	0	0	5	1.00	10	8	0	0.71	0	4	2	4	0.00	4	6	0.40	38	11	13	0.78	0.75	0.76
015	5	1	1	0.83	3	2	0	0	0.75	12	8	0	0.75	6	2	0	4	0.86	4	8	0.33	44	19	16	0.70	0.73	0.72
016	3	1	0	0.86	2	0	0	1	1.00	6	0	0	1.00	0	2	0	4	0.00	4	2	0.67	26	5	1	0.84	0.96	0.90
017	6	1	1	0.86	1	0	1	4	0.67	10	6	0	0.77	0	0	0	10	0.00	8	2	0.80	42	18	9	0.70	0.82	0.76
018	3	0	0	1.00	1	0	2	0	0.50	2	0	4	0.50	0	0	0	2	0.00	2	0	1.00	26	8	12	0.76	0.68	0.72
020	8	1	1	0.89	1	0	1	6	0.67	6	16	4	0.37	2	0	2	2	0.67	4	2	0.67	46	27	14	0.63	0.77	0.69
021	5	2	2	0.71	3	1	0	1	0.86	8	4	6	0.62	2	0	0	6	1.00	4	4	0.50	40	15	17	0.73	0.70	0.71
022	4	1	2	0.73	1	2	0	1	0.50	4	4	0	0.67	0	0	0	4	0.00	2	2	0.50	24	19	18	0.56	0.57	0.56
024	4	1	3	0.67	1	0	2	1	0.50	4	2	2	0.67	0	0	2	2	0.00	0	4	0.00	32	16	27	0.67	0.54	0.60
025	5	0	0	1.00	1	0	0	4	1.00	2	4	2	0.40	0	0	0	2	0.00	2	0	1.00	26	4	2	0.87	0.93	0.90

Table A4. *Cont.*

					Character										**Interaction**									**Network**			
		Existence					**Main**				**Existence**					**Strength**				**Count**				**Score**			
	TP	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**TP**	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**corr**	**inc**	**Acc**	**TP**	**FP**	**FN**	**P**	**R**	**F1**
026	4	0	2	0.80	1	0	1	2	0.67	8	4	0	0.80	2	2	0	4	0.67	4	4	0.50	32	9	15	0.78	0.68	0.73
027	5	0	0	1.00	2	0	2	1	0.67	6	6	2	0.60	0	0	0	6	0.00	4	2	0.67	42	14	10	0.75	0.81	0.78
028	5	0	0	1.00	0	1	2	2	0.00	0	2	6	0.00	0	0	0	0	0.00	0	1	0.00	28	12	16	0.70	0.64	0.67
029	8	0	2	0.89	1	0	1	6	0.67	14	10	0	0.74	2	2	0	10	0.67	10	4	0.71	54	15	15	0.78	0.78	0.78
030	6	0	2	0.86	0	6	0	0	0.00	10	20	0	0.50	0	0	0	10	0.00	10	0	1.00	34	32	22	0.52	0.61	0.56
031	9	2	1	0.86	1	0	1	7	0.67	22	12	2	0.76	0	0	2	20	0.00	16	6	0.73	66	29	12	0.69	0.85	0.76
032	3	0	0	1.00	1	0	0	2	1.00	4	2	0	0.80	4	0	0	0	1.00	0	4	0.00	20	2	0	0.91	1.00	0.95
033	3	0	0	1.00	1	0	0	2	1.00	4	0	0	1.00	0	0	2	2	0.00	2	2	0.50	20	1	1	0.95	0.95	0.95
034	6	0	0	1.00	1	0	0	5	1.00	16	2	2	0.89	2	0	2	12	0.67	10	6	0.62	44	3	3	0.94	0.94	0.94
035	3	0	0	1.00	0	1	1	1	0.00	6	0	0	1.00	0	0	0	6	0.00	4	2	0.67	22	6	6	0.79	0.79	0.79
036	9	0	2	0.90	2	2	2	3	0.50	16	24	4	0.53	2	4	2	8	0.40	8	8	0.50	68	39	36	0.64	0.65	0.64
037	9	1	1	0.90	1	0	0	8	1.00	10	6	8	0.59	2	0	0	8	1.00	2	8	0.20	50	13	13	0.79	0.79	0.79
038-1	3	1	0	0.86	1	2	0	0	0.50	4	0	0	1.00	0	0	0	4	0.00	2	2	0.50	20	10	4	0.67	0.83	0.74
038-2	3	0	0	1.00	1	1	0	1	0.67	4	2	0	0.80	0	0	0	4	0.00	4	0	1.00	20	4	2	0.83	0.91	0.87
039-1	2	1	1	0.67	1	0	0	1	1.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	4	4	0.78	0.78	0.78
040	5	1	1	0.83	1	0	1	3	0.67	8	0	0	1.00	0	2	0	6	0.00	4	4	0.50	36	11	12	0.77	0.75	0.76
041	2	0	0	1.00	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	2	2	0.88	0.88	0.88
042	3	0	1	0.86	1	0	0	2	1.00	2	0	2	0.67	0	0	0	2	0.00	0	2	0.00	18	0	7	1.00	0.72	0.84
043	3	0	0	1.00	1	0	1	1	0.67	4	0	0	1.00	0	0	0	4	0.00	4	0	1.00	24	4	4	0.86	0.86	0.86
044	4	1	1	0.80	2	0	0	2	1.00	6	0	0	1.00	2	0	0	4	1.00	4	2	0.67	30	6	5	0.83	0.86	0.85
046	3	1	1	0.75	2	1	0	0	0.80	4	2	0	0.80	0	0	2	2	0.00	0	4	0.00	24	12	8	0.67	0.75	0.71
048	5	0	0	1.00	1	0	1	3	0.67	8	4	0	0.80	0	0	2	6	0.00	4	4	0.50	36	9	5	0.80	0.88	0.84
049	6	0	1	0.92	1	1	1	3	0.50	8	4	0	0.80	0	0	0	8	0.00	6	2	0.75	40	10	13	0.80	0.75	0.78
050	4	1	1	0.80	1	0	1	2	0.67	4	8	0	0.50	0	0	0	4	0.00	4	0	1.00	28	16	8	0.64	0.78	0.70
051	5	1	0	0.91	3	1	0	1	0.86	6	6	0	0.67	2	0	0	4	1.00	0	6	0.00	38	20	2	0.66	0.95	0.78
052	3	1	0	0.86	1	0	1	1	0.67	4	0	2	0.80	0	0	2	2	0.00	2	2	0.50	24	10	7	0.71	0.77	0.74
053	8	1	1	0.89	1	0	1	6	0.67	14	2	4	0.82	4	0	0	10	1.00	10	4	0.71	54	11	13	0.83	0.81	0.82
054	6	0	4	0.75	1	1	2	2	0.40	2	6	4	0.29	0	0	0	2	0.00	2	0	1.00	38	16	33	0.70	0.54	0.61
055	4	1	1	0.80	2	0	0	2	1.00	6	0	0	1.00	2	0	0	4	1.00	2	4	0.33	30	10	6	0.75	0.83	0.79
056	5	1	0	0.91	1	0	1	3	0.67	6	4	0	0.75	0	0	0	6	0.00	2	4	0.33	34	13	4	0.72	0.89	0.80
057	7	2	0	0.88	1	1	2	3	0.40	4	10	6	0.33	2	0	0	2	1.00	2	2	0.50	44	31	16	0.59	0.73	0.65
058	4	1	0	0.89	1	0	1	2	0.67	8	0	0	1.00	2	0	2	4	0.67	4	4	0.50	32	11	5	0.74	0.86	0.80
059	6	2	0	0.86	1	0	0	5	1.00	8	2	0	0.89	2	0	0	6	1.00	2	6	0.25	36	14	0	0.72	1.00	0.84
061	7	4	2	0.70	1	0	1	5	0.67	14	4	2	0.82	2	2	2	8	0.50	8	6	0.57	50	35	18	0.59	0.74	0.65
063	4	1	0	0.89	2	2	0	0	0.67	4	0	0	1.00	0	0	2	2	0.00	0	4	0.00	28	11	5	0.72	0.85	0.78
064	9	4	1	0.78	1	0	1	7	0.67	8	0	2	0.89	0	0	2	6	0.00	6	2	0.75	52	28	12	0.65	0.81	0.72
065	6	1	1	0.86	1	1	0	4	0.67	10	6	0	0.77	2	0	0	8	1.00	6	4	0.60	38	19	9	0.67	0.81	0.73
069	3	0	0	1.00	1	0	1	1	0.67	0	2	4	0.00	0	0	0	0	0.00	0	1	0.00	20	6	8	0.77	0.71	0.74
070	6	1	0	0.92	1	0	1	4	0.67	2	2	0	0.67	0	0	0	2	0.00	0	2	0.00	34	10	4	0.77	0.89	0.83

Table A5. Detailed results for ASP.

	Character									Interaction												Network					
	Existence				Main					Existence				Strength					Count			Score					
	TP	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**TP**	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**corr**	**inc**	**Acc**	**TP**	**FP**	**FN**	**P**	**R**	**F1**
001	4	0	0	1.00	2	0	0	2	1.00	6	0	2	0.86	2	2	0	2	0.67	4	2	0.67	30	1	3	0.97	0.91	0.94
002	2	0	0	1.00	2	0	0	0	1.00	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	18	0	0	1.00	1.00	1.00
003	3	0	2	0.75	2	0	0	1	1.00	4	0	0	1.00	2	0	0	2	1.00	2	2	0.50	24	0	9	1.00	0.73	0.84
004	11	0	4	0.85	1	0	0	10	1.00	24	18	0	0.73	2	4	0	18	0.50	14	10	0.58	72	20	23	0.78	0.76	0.77
005	3	2	2	0.60	2	0	0	1	1.00	2	0	2	0.67	0	0	2	0	0.00	0	2	0.00	22	11	13	0.67	0.63	0.65
006	7	2	2	0.78	1	2	0	4	0.50	10	10	2	0.62	2	2	2	4	0.50	4	6	0.40	42	27	19	0.61	0.69	0.65
007	6	1	1	0.86	1	0	1	4	0.67	12	4	0	0.86	4	0	0	8	1.00	2	10	0.17	44	16	9	0.73	0.83	0.78
008	4	0	1	0.89	1	0	3	0	0.40	6	2	2	0.75	0	0	0	6	0.00	2	4	0.33	38	14	18	0.73	0.68	0.70
009	6	0	2	0.86	0	2	2	2	0.00	10	2	0	0.91	0	0	0	10	0.00	8	2	0.80	42	14	22	0.75	0.66	0.70
010	5	0	0	1.00	1	0	1	3	0.67	6	4	0	0.75	0	0	0	6	0.00	4	2	0.67	34	8	4	0.81	0.89	0.85
011	6	1	3	0.75	1	0	2	3	0.50	8	4	2	0.73	4	0	0	4	1.00	4	4	0.50	44	18	28	0.71	0.61	0.66
012	4	0	1	0.89	2	0	0	2	1.00	8	0	0	1.00	2	0	2	4	0.67	2	6	0.25	32	1	6	0.97	0.84	0.90
013	8	1	2	0.84	1	0	1	6	0.67	18	14	2	0.69	2	2	0	14	0.67	14	4	0.78	58	26	25	0.69	0.70	0.69
014	6	1	2	0.80	1	2	0	3	0.50	10	8	0	0.71	0	0	2	8	0.00	6	4	0.60	38	20	15	0.66	0.72	0.68
015	6	0	0	1.00	2	1	2	1	0.57	18	8	0	0.82	8	4	0	6	0.80	4	14	0.22	58	20	12	0.74	0.83	0.78
016	3	2	0	0.75	2	1	0	0	0.80	6	0	0	1.00	0	4	0	2	0.00	2	4	0.33	26	15	4	0.63	0.87	0.73
017	5	2	2	0.71	1	0	1	3	0.67	6	2	2	0.75	0	0	0	6	0.00	2	4	0.33	34	18	16	0.65	0.68	0.67
018	3	0	0	1.00	2	0	1	0	0.80	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	30	4	4	0.88	0.88	0.88
020	8	2	1	0.84	1	0	1	6	0.67	8	8	2	0.62	2	0	2	4	0.67	4	4	0.50	48	25	12	0.66	0.80	0.72
021	5	1	2	0.77	1	0	2	2	0.50	12	6	2	0.75	2	4	2	4	0.40	4	8	0.33	44	22	22	0.67	0.67	0.67
022	6	0	0	1.00	1	0	1	4	0.67	8	4	0	0.80	0	0	0	8	0.00	8	0	1.00	40	8	4	0.83	0.91	0.87
024	4	1	3	0.67	1	1	1	1	0.50	4	4	0	0.67	0	0	2	2	0.00	2	2	0.50	28	17	28	0.62	0.50	0.55
025	3	0	2	0.75	1	0	0	2	1.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	18	0	9	1.00	0.67	0.80
026	4	0	2	0.80	1	0	1	2	0.67	8	4	0	0.80	2	2	0	4	0.67	4	4	0.50	32	9	15	0.78	0.68	0.73
027	3	1	2	0.67	2	0	1	0	0.80	4	2	0	0.80	0	2	0	2	0.00	2	2	0.50	28	12	19	0.70	0.60	0.64
028	4	0	1	0.89	1	0	0	3	1.00	4	4	0	0.67	0	0	0	4	0.00	2	2	0.50	24	4	9	0.86	0.73	0.79
029	7	0	3	0.82	1	0	1	5	0.67	12	6	0	0.80	0	4	0	8	0.00	8	4	0.67	48	12	21	0.80	0.70	0.74
030	4	0	4	0.67	0	1	0	3	0.00	4	8	0	0.50	0	0	0	4	0.00	4	0	1.00	20	10	23	0.67	0.47	0.55
031	10	0	0	1.00	1	0	1	8	0.67	26	6	0	0.90	0	6	2	18	0.00	14	12	0.54	74	14	8	0.84	0.90	0.87
032	3	0	0	1.00	1	0	0	2	1.00	4	2	0	0.80	2	0	2	0	0.67	0	4	0.00	20	3	1	0.87	0.95	0.91
033	3	0	0	1.00	1	0	0	2	1.00	4	0	0	1.00	0	0	2	2	0.00	2	2	0.50	20	1	1	0.95	0.95	0.95
034	5	2	1	0.77	1	0	0	4	1.00	12	2	0	0.92	2	2	0	8	0.67	6	6	0.50	36	15	8	0.71	0.82	0.76
035	3	0	0	1.00	1	0	0	2	1.00	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	22	0	0	1.00	1.00	1.00
036	10	3	1	0.83	3	2	2	3	0.60	14	16	14	0.48	2	0	4	8	0.50	6	8	0.43	74	51	33	0.59	0.69	0.64
037	8	2	2	0.80	1	0	0	7	1.00	10	10	2	0.62	2	2	0	6	0.67	4	6	0.40	46	24	15	0.66	0.75	0.70
038-1	3	0	0	1.00	1	0	0	2	1.00	4	2	0	0.80	0	0	0	4	0.00	2	2	0.50	20	2	0	0.91	1.00	0.95
038-2	2	0	1	0.80	0	1	0	1	0.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	10	2	11	0.83	0.48	0.61
039-1	2	0	1	0.80	1	0	0	1	1.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	0	4	1.00	0.78	0.88
040	6	1	0	0.92	1	0	1	4	0.67	12	0	2	0.92	0	0	0	12	0.00	10	2	0.83	44	11	6	0.80	0.88	0.84
041	2	0	0	1.00	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	2	2	0.88	0.88	0.88

Table A5. *Cont.*

	Character									Interaction												Network					
	Existence				Main					Existence				Strength					Count			Score					
	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
042	1	0	3	0.40	0	1	0	0	0.00	0	0	0	0.00	0	0	0	0	0.00	0	1	0.00	4	2	21	0.67	0.16	0.26
043	3	0	0	1.00	2	0	0	1	1.00	4	0	0	1.00	0	0	0	4	0.00	4	0	1.00	24	0	0	1.00	1.00	1.00
044	5	0	0	1.00	1	0	1	3	0.67	8	2	0	0.89	2	0	0	6	1.00	6	2	0.75	36	6	4	0.86	0.90	0.88
046	3	2	1	0.67	2	0	0	1	1.00	4	2	0	0.80	0	0	2	2	0.00	2	2	0.50	24	14	6	0.63	0.80	0.71
048	5	0	0	1.00	1	0	1	3	0.67	4	4	4	0.50	0	0	2	2	0.00	0	4	0.00	32	9	9	0.78	0.78	0.78
049	5	0	2	0.83	1	1	1	2	0.50	6	4	0	0.75	0	0	0	6	0.00	4	2	0.67	34	10	18	0.77	0.65	0.71
050	4	0	1	0.89	1	0	1	2	0.67	2	4	0	0.50	0	0	0	2	0.00	2	0	1.00	26	8	9	0.76	0.74	0.75
051	5	0	0	1.00	3	1	0	1	0.86	6	6	0	0.67	2	0	0	4	1.00	0	6	0.00	38	8	2	0.83	0.95	0.88
052	3	1	0	0.86	1	0	1	1	0.67	4	0	2	0.80	2	0	0	2	1.00	0	4	0.00	24	10	6	0.71	0.80	0.75
053	7	2	2	0.78	1	0	1	5	0.67	16	2	2	0.89	4	0	0	12	1.00	10	6	0.62	52	19	15	0.73	0.78	0.75
054	8	3	2	0.76	3	3	0	2	0.67	4	2	8	0.44	0	0	0	4	0.00	4	0	1.00	48	35	22	0.58	0.69	0.63
055	4	0	1	0.89	1	0	1	2	0.67	8	0	0	1.00	2	0	0	6	1.00	6	2	0.75	32	4	9	0.89	0.78	0.83
056	4	1	1	0.80	1	0	1	2	0.67	2	4	0	0.50	0	0	0	2	0.00	0	2	0.00	26	13	10	0.67	0.72	0.69
057	5	1	2	0.77	1	0	1	3	0.67	4	8	2	0.44	2	0	0	2	1.00	2	2	0.50	32	19	20	0.63	0.62	0.62
058	4	0	0	1.00	2	0	0	2	1.00	8	0	0	1.00	2	0	2	4	0.67	4	4	0.50	32	1	1	0.97	0.97	0.97
059	5	2	1	0.77	1	0	0	4	1.00	6	2	0	0.86	2	2	0	2	0.67	2	4	0.33	30	17	6	0.64	0.83	0.72
061	5	1	4	0.67	1	0	1	3	0.67	10	4	0	0.83	0	4	0	6	0.00	4	6	0.40	38	15	27	0.72	0.58	0.64
063	4	1	0	0.89	2	2	0	0	0.67	4	0	0	1.00	0	0	2	2	0.00	2	2	0.50	28	14	5	0.67	0.85	0.75
064	7	1	3	0.78	1	0	1	5	0.67	10	0	0	1.00	0	0	2	8	0.00	8	2	0.80	46	9	18	0.84	0.72	0.77
065	6	1	1	0.86	1	1	0	4	0.67	10	6	0	0.77	2	0	0	8	1.00	2	8	0.20	38	16	9	0.70	0.81	0.75
069	3	0	0	1.00	1	0	1	1	0.67	0	2	4	0.00	0	0	0	0	0.00	0	1	0.00	20	6	8	0.77	0.71	0.74
070	5	1	1	0.83	1	3	1	0	0.33	2	2	0	0.67	0	0	0	2	0.00	0	2	0.00	30	16	14	0.65	0.68	0.67

Table A6. Detailed results for ChatGPT with predicted mentions.

	Character									Interaction												Network					
	Existence				Main					Existence				Strength					Count			Score					
	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
001	4	1	0	0.89	2	0	0	2	1.00	6	0	2	0.86	2	0	0	4	1.00	2	4	0.33	30	5	2	0.86	0.94	0.90
002	2	0	0	1.00	2	0	0	0	1.00	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	18	0	0	1.00	1.00	1.00
003	3	0	2	0.75	1	0	1	1	0.67	4	0	0	1.00	2	0	0	2	1.00	2	2	0.50	24	4	13	0.86	0.65	0.74
004	11	6	4	0.69	0	0	1	10	0.00	12	4	14	0.57	0	0	0	12	0.00	12	0	1.00	60	50	38	0.55	0.61	0.58
005	3	0	2	0.75	2	0	0	1	1.00	4	2	0	0.80	2	0	0	2	1.00	2	2	0.50	24	2	10	0.92	0.71	0.80
006	6	6	3	0.57	1	2	0	3	0.50	8	2	4	0.73	0	0	2	6	0.00	4	4	0.50	36	61	24	0.37	0.60	0.46
007	4	1	3	0.67	1	0	1	2	0.67	8	2	0	0.89	4	0	0	4	1.00	4	4	0.50	32	12	19	0.73	0.63	0.67
008	5	1	0	0.91	1	0	3	1	0.40	4	0	4	0.67	0	0	0	4	0.00	0	4	0.00	40	22	16	0.65	0.71	0.68
009	7	0	1	0.93	1	3	1	2	0.33	12	2	0	0.92	0	0	0	12	0.00	8	4	0.67	48	12	15	0.80	0.76	0.78
010	2	1	3	0.50	0	1	0	1	0.00	0	0	0	0.00	0	0	0	0	0.00	0	1	0.00	8	11	25	0.42	0.24	0.31

Table A6. *Cont.*

					Character										**Interaction**									**Network**		
		Existence					**Main**				**Existence**					**Strength**					**Count**				**Score**		
	TP	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**TP**	**FP**	**FN**	**F1**	**TP**	**FP**	**FN**	**TN**	**F1**	**corr**	**inc**	**Acc**	**TP**	**FP**	**FN**	**P**	**R**	**F1**
011	6	5	3	0.60	1	1	2	2	0.40	4	4	6	0.44	0	0	2	2	0.00	2	2	0.50	40	47	35	0.46	0.53	0.49
012	5	1	0	0.91	1	0	1	3	0.67	8	0	2	0.89	2	0	2	4	0.67	4	4	0.50	36	10	7	0.78	0.84	0.81
013	6	4	4	0.60	3	3	0	0	0.67	4	2	12	0.36	0	0	2	2	0.00	0	4	0.00	40	38	43	0.51	0.48	0.50
014	4	0	4	0.67	1	2	0	1	0.50	6	2	0	0.86	0	0	2	4	0.00	0	6	0.00	26	7	25	0.79	0.51	0.62
015	6	8	0	0.60	1	0	3	2	0.40	10	0	8	0.71	0	0	6	4	0.00	2	8	0.20	50	63	23	0.44	0.68	0.54
016	3	1	0	0.86	2	1	0	0	0.80	6	0	0	1.00	0	0	0	6	0.00	4	2	0.67	26	8	2	0.76	0.93	0.84
017	7	2	0	0.88	0	0	2	5	0.00	0	6	12	0.00	0	0	0	0	0.00	0	1	0.00	36	30	20	0.55	0.64	0.59
020	9	5	0	0.78	1	0	1	7	0.67	8	4	4	0.67	0	0	4	4	0.00	2	6	0.25	52	37	10	0.58	0.84	0.69
021	7	3	0	0.82	3	4	0	0	0.60	8	12	8	0.44	2	2	2	2	0.50	4	4	0.50	48	43	18	0.53	0.73	0.61
022	3	0	3	0.67	2	0	0	1	1.00	4	0	0	1.00	0	0	0	4	0.00	2	2	0.50	24	0	14	1.00	0.63	0.77
024	4	0	3	0.73	2	1	1	0	0.67	2	2	4	0.40	0	0	2	0	0.00	0	2	0.00	30	9	27	0.77	0.53	0.62
025	3	0	2	0.75	1	0	0	2	1.00	2	4	0	0.50	0	0	0	2	0.00	0	2	0.00	18	4	9	0.82	0.67	0.73
026	6	0	0	1.00	1	0	1	4	0.67	12	2	0	0.92	2	2	0	8	0.67	8	4	0.67	44	7	5	0.86	0.90	0.88
027	2	0	3	0.57	2	0	0	0	1.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	18	0	23	1.00	0.44	0.61
028	3	0	2	0.75	0	1	1	1	0.00	0	2	2	0.00	0	0	0	0	0.00	0	1	0.00	16	8	22	0.67	0.42	0.52
029	9	1	1	0.90	2	2	0	5	0.67	16	6	0	0.84	2	0	0	14	1.00	6	10	0.38	60	22	9	0.73	0.87	0.79
030	6	0	2	0.86	0	6	0	0	0.00	10	20	0	0.50	0	0	0	10	0.00	10	0	1.00	34	32	22	0.52	0.61	0.56
031	9	1	1	0.90	1	0	1	7	0.67	22	4	0	0.92	2	2	0	18	0.67	18	4	0.82	66	17	11	0.80	0.86	0.82
032	3	0	0	1.00	1	0	0	2	1.00	4	2	0	0.80	4	0	0	0	1.00	0	4	0.00	20	2	0	0.91	1.00	0.95
033	3	2	0	0.75	1	1	0	1	0.67	2	0	2	0.67	0	0	2	0	0.00	0	2	0.00	18	16	5	0.53	0.78	0.63
034	6	0	0	1.00	1	0	0	5	1.00	18	2	0	0.95	4	0	0	14	1.00	12	6	0.67	46	2	0	0.96	1.00	0.98
035	3	0	0	1.00	1	0	0	2	1.00	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	22	0	0	1.00	1.00	1.00
037	10	2	0	0.91	1	0	0	9	1.00	18	6	2	0.82	2	0	0	16	1.00	6	12	0.33	62	18	2	0.78	0.97	0.86
038-1	3	0	0	1.00	0	1	1	1	0.00	4	0	0	1.00	0	0	0	4	0.00	2	2	0.50	20	6	6	0.77	0.77	0.77
038-2	3	0	0	1.00	1	2	0	0	0.50	4	2	0	0.80	0	0	0	4	0.00	4	0	1.00	20	6	4	0.77	0.83	0.80
039-1	3	0	0	1.00	1	0	0	2	1.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	18	0	0	1.00	1.00	1.00
040	5	2	1	0.77	1	0	1	3	0.67	6	0	2	0.86	0	0	0	6	0.00	4	2	0.67	34	16	13	0.68	0.72	0.70
042	2	0	2	0.67	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	2	12	0.88	0.54	0.67
043	3	0	0	1.00	1	0	1	1	0.67	2	0	2	0.67	0	0	0	2	0.00	2	0	1.00	22	4	6	0.85	0.79	0.81
044	4	2	1	0.73	2	1	0	1	0.80	6	2	0	0.86	0	0	2	4	0.00	4	2	0.67	30	16	8	0.65	0.79	0.71
046	2	1	2	0.57	2	0	0	0	1.00	2	0	0	1.00	0	0	2	0	0.00	0	2	0.00	18	6	11	0.75	0.62	0.68
048	4	0	1	0.89	1	0	1	2	0.67	6	2	2	0.75	0	0	2	4	0.00	2	4	0.33	30	7	11	0.81	0.73	0.77
049	7	0	0	1.00	1	0	1	5	0.67	8	4	6	0.62	0	0	0	8	0.00	6	2	0.75	44	8	10	0.85	0.81	0.83
050	1	1	4	0.29	1	0	0	0	1.00	0	0	0	0.00	0	0	0	0	0.00	0	1	0.00	8	8	22	0.50	0.27	0.35
051	5	0	0	1.00	3	0	0	2	1.00	6	6	0	0.67	2	0	0	4	1.00	0	6	0.00	38	6	0	0.86	1.00	0.93
052	3	1	0	0.86	1	0	1	1	0.67	4	0	2	0.80	2	0	0	2	1.00	0	4	0.00	24	9	6	0.73	0.80	0.76
053	7	8	2	0.58	2	3	0	2	0.57	2	6	12	0.18	2	0	0	0	1.00	0	2	0.00	38	68	29	0.36	0.57	0.44
054	6	1	4	0.71	1	0	2	3	0.50	8	2	2	0.80	0	2	0	6	0.00	6	2	0.75	44	17	28	0.72	0.61	0.66
055	3	2	2	0.60	2	1	0	0	0.80	2	0	2	0.67	0	0	2	0	0.00	0	2	0.00	22	21	16	0.51	0.58	0.54
056	3	1	2	0.67	2	1	0	0	0.80	4	2	0	0.80	0	0	0	4	0.00	2	2	0.50	24	9	11	0.73	0.69	0.71

Table A6. *Cont.*

					Character										Interaction										Network		
		Existence					Main					Existence				Strength				Count					Score		
	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
057	7	1	0	0.93	1	1	2	3	0.40	6	8	4	0.50	2	0	0	4	1.00	4	2	0.67	46	23	14	0.67	0.77	0.71
058	3	0	1	0.86	2	0	0	1	1.00	6	0	0	1.00	2	0	0	4	1.00	2	4	0.33	26	0	5	1.00	0.84	0.91
059	6	1	0	0.92	1	0	0	5	1.00	8	2	0	0.89	2	0	0	6	1.00	2	6	0.25	36	7	0	0.84	1.00	0.91
061	8	2	1	0.84	1	0	1	6	0.67	14	2	4	0.82	2	2	2	8	0.50	8	6	0.57	54	20	15	0.73	0.78	0.76
063	4	0	0	1.00	2	2	0	0	0.67	4	0	0	1.00	0	0	2	2	0.00	0	4	0.00	28	5	5	0.85	0.85	0.85
064	8	2	2	0.80	1	0	1	6	0.67	10	0	2	0.91	0	0	2	8	0.00	8	2	0.80	50	16	15	0.76	0.77	0.76
065	6	0	1	0.92	1	0	0	5	1.00	14	0	0	1.00	2	0	0	12	1.00	4	10	0.29	42	0	5	1.00	0.89	0.94
069	3	0	0	1.00	2	1	0	0	0.80	0	0	4	0.00	0	0	0	0	0.00	0	1	0.00	20	2	6	0.91	0.77	0.83
070	2	0	4	0.50	0	2	0	0	0.00	0	0	0	0.00	0	0	0	0	0.00	0	1	0.00	8	4	29	0.67	0.22	0.33

Table A7. Detailed results for ChatGPT with gold mentions.

					Character										Interaction										Network		
		Existence					Main					Existence				Strength				Count					Score		
	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
001	4	1	0	0.89	2	0	0	2	1.00	8	0	0	1.00	2	0	0	6	1.00	4	4	0.50	32	5	0	0.86	1.00	0.93
002	2	0	0	1.00	2	0	0	0	1.00	2	0	0	1.00	2	0	0	0	1.00	0	2	0.00	18	0	0	1.00	1.00	1.00
003	4	1	1	0.80	2	0	0	2	1.00	2	0	4	0.50	2	0	0	0	1.00	0	2	0.00	26	11	8	0.70	0.76	0.73
004	14	1	1	0.93	1	0	0	13	1.00	32	14	0	0.82	2	2	0	28	0.67	28	4	0.88	92	21	6	0.81	0.94	0.87
005	3	0	2	0.75	2	0	0	1	1.00	4	2	0	0.80	2	0	0	2	1.00	2	2	0.50	24	2	10	0.92	0.71	0.80
006	8	1	1	0.89	1	0	0	7	1.00	10	6	4	0.67	2	0	0	8	1.00	8	2	0.80	46	13	10	0.78	0.82	0.80
007	7	1	0	0.93	1	0	1	5	0.67	14	4	0	0.88	4	0	0	10	1.00	6	8	0.43	50	16	4	0.76	0.93	0.83
008	5	0	0	1.00	1	0	3	1	0.40	8	2	0	0.89	0	0	0	8	0.00	4	4	0.50	44	14	12	0.76	0.79	0.77
009	7	1	1	0.88	1	2	1	3	0.40	10	2	2	0.83	0	0	0	10	0.00	6	4	0.60	46	16	15	0.74	0.75	0.75
010	4	0	1	0.89	1	0	1	2	0.67	4	2	0	0.80	0	0	0	4	0.00	4	0	1.00	28	6	9	0.82	0.76	0.79
011	8	3	1	0.80	1	0	2	5	0.50	12	4	6	0.71	2	0	2	8	0.67	8	4	0.67	56	30	21	0.65	0.73	0.69
012	5	0	0	1.00	2	1	0	2	0.80	10	0	0	1.00	2	0	2	6	0.67	6	4	0.60	38	3	3	0.93	0.93	0.93
013	10	0	0	1.00	1	0	2	7	0.50	32	16	0	0.80	2	2	0	28	0.67	24	8	0.75	84	25	9	0.77	0.90	0.83
014	8	0	0	1.00	1	0	0	7	1.00	14	12	0	0.70	0	2	2	10	0.00	10	4	0.71	50	14	2	0.78	0.96	0.86
015	6	1	0	0.92	4	2	0	0	0.80	18	8	0	0.82	8	2	0	8	0.89	6	12	0.33	58	19	5	0.75	0.92	0.83
016	3	1	0	0.86	2	1	0	0	0.80	6	0	0	1.00	0	4	0	2	0.00	2	4	0.33	26	8	4	0.76	0.87	0.81
017	7	1	0	0.93	1	0	1	5	0.67	12	6	0	0.80	0	0	0	12	0.00	8	4	0.67	48	15	4	0.76	0.92	0.83
018	3	0	0	1.00	3	0	0	0	1.00	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	30	0	0	1.00	1.00	1.00
020	7	1	2	0.82	1	0	1	5	0.67	8	8	2	0.62	2	0	0	6	1.00	6	2	0.75	44	18	15	0.71	0.75	0.73
021	6	5	1	0.67	1	0	2	3	0.50	10	2	6	0.71	2	2	0	6	0.67	4	6	0.40	46	36	19	0.56	0.71	0.63
022	6	1	0	0.92	1	0	1	4	0.67	8	4	0	0.80	0	0	0	8	0.00	8	0	1.00	40	14	4	0.74	0.91	0.82
024	6	1	1	0.86	1	0	2	3	0.50	12	2	2	0.86	0	0	2	10	0.00	10	2	0.83	48	16	15	0.75	0.76	0.76
025	5	0	0	1.00	1	0	0	4	1.00	4	8	0	0.50	0	0	0	4	0.00	4	0	1.00	28	8	0	0.78	1.00	0.88

Table A7. *Cont.*

	Character									Interaction												Network					
	Existence				Main					Existence				Strength					Count			Score					
	TP	FP	FN	F1	TP	FP	FN	TN	F1	TP	FP	FN	F1	TP	FP	FN	TN	F1	corr	inc	Acc	TP	FP	FN	P	R	F1
026	6	0	0	1.00	1	0	1	4	0.67	12	2	0	0.92	2	2	0	8	0.67	8	4	0.67	44	7	5	0.86	0.90	0.88
027	4	0	1	0.89	1	0	3	0	0.40	6	2	0	0.86	0	0	0	6	0.00	6	0	1.00	38	14	17	0.73	0.69	0.71
028	4	0	1	0.89	0	1	1	2	0.00	0	2	0	0.00	0	0	0	0	0.00	0	1	0.00	20	8	17	0.71	0.54	0.62
029	8	2	2	0.80	2	3	0	3	0.57	14	8	0	0.78	2	0	0	12	1.00	6	8	0.43	54	31	16	0.64	0.77	0.70
030	6	0	2	0.86	0	6	0	0	0.00	10	20	0	0.50	0	0	0	10	0.00	10	0	1.00	34	32	22	0.52	0.61	0.56
031	10	4	0	0.83	2	5	0	3	0.44	22	2	4	0.88	0	0	2	20	0.00	18	4	0.82	70	45	15	0.61	0.82	0.70
032	3	0	0	1.00	1	0	0	2	1.00	4	2	0	0.80	0	0	4	0	0.00	0	4	0.00	20	4	2	0.83	0.91	0.87
033	3	0	0	1.00	1	0	0	2	1.00	4	0	0	1.00	0	0	2	2	0.00	2	2	0.50	20	1	1	0.95	0.95	0.95
034	6	0	0	1.00	1	0	0	5	1.00	18	2	0	0.95	2	0	2	14	0.67	12	6	0.67	46	3	1	0.94	0.98	0.96
035	3	0	0	1.00	1	0	0	2	1.00	6	0	0	1.00	0	0	0	6	0.00	6	0	1.00	22	0	0	1.00	1.00	1.00
036	11	0	0	1.00	4	1	1	5	0.80	30	4	0	0.94	4	0	2	24	0.80	24	6	0.80	94	11	7	0.90	0.93	0.91
037	9	2	1	0.86	1	0	0	8	1.00	18	6	2	0.82	2	0	0	16	1.00	6	12	0.33	58	17	6	0.77	0.91	0.83
038-1	3	0	0	1.00	1	1	0	1	0.67	4	0	0	1.00	0	0	0	4	0.00	4	0	1.00	20	2	2	0.91	0.91	0.91
038-2	3	0	0	1.00	1	2	0	0	0.50	4	2	0	0.80	0	0	0	4	0.00	4	0	1.00	20	6	4	0.77	0.83	0.80
039-1	3	0	0	1.00	1	0	0	2	1.00	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	18	0	0	1.00	1.00	1.00
041	2	0	0	1.00	1	1	0	0	0.67	2	0	0	1.00	0	0	0	2	0.00	2	0	1.00	14	2	2	0.88	0.88	0.88
042	4	1	0	0.89	1	0	0	3	1.00	4	0	2	0.80	0	0	0	4	0.00	2	2	0.50	24	5	2	0.83	0.92	0.87
043	3	0	0	1.00	1	0	1	1	0.67	4	0	0	1.00	0	0	0	4	0.00	4	0	1.00	24	4	4	0.86	0.86	0.86
044	5	0	0	1.00	2	1	0	2	0.80	8	2	0	0.89	2	0	0	6	1.00	6	2	0.75	36	4	2	0.90	0.95	0.92
046	2	1	2	0.57	2	0	0	0	1.00	2	0	0	1.00	2	0	0	0	1.00	2	0	1.00	18	6	10	0.75	0.64	0.69
048	5	0	0	1.00	1	0	1	3	0.67	8	4	0	0.80	2	0	0	6	1.00	6	2	0.75	36	8	4	0.82	0.90	0.86
049	7	0	0	1.00	1	0	1	5	0.67	12	2	2	0.86	0	0	0	12	0.00	10	2	0.83	48	6	6	0.89	0.89	0.89
050	5	1	0	0.91	1	0	1	3	0.67	4	8	0	0.50	0	0	0	4	0.00	4	0	1.00	32	16	4	0.67	0.89	0.76
051	5	0	0	1.00	3	1	0	1	0.86	6	6	0	0.67	0	0	2	4	0.00	0	6	0.00	38	9	3	0.81	0.93	0.86
052	3	1	0	0.86	1	0	1	1	0.67	4	0	2	0.80	2	0	0	2	1.00	0	4	0.00	24	9	6	0.73	0.80	0.76
053	9	2	0	0.90	2	1	0	6	0.80	16	6	4	0.76	6	0	0	10	1.00	14	2	0.88	60	21	6	0.74	0.91	0.82
054	6	1	4	0.71	1	0	2	3	0.50	8	2	2	0.80	0	0	0	8	0.00	6	2	0.75	44	16	27	0.73	0.62	0.67
055	5	0	0	1.00	2	0	0	3	1.00	10	0	0	1.00	2	0	0	8	1.00	6	4	0.60	38	0	0	1.00	1.00	1.00
056	5	2	0	0.83	2	1	0	2	0.80	6	4	0	0.75	0	0	0	6	0.00	2	4	0.33	34	17	2	0.67	0.94	0.78
057	7	1	0	0.93	1	1	2	3	0.40	6	10	4	0.46	2	0	0	4	1.00	4	2	0.67	46	25	14	0.65	0.77	0.70
058	4	0	0	1.00	2	0	0	2	1.00	8	0	0	1.00	2	0	2	4	0.67	4	4	0.50	32	1	1	0.97	0.97	0.97
059	4	0	2	0.80	1	0	0	3	1.00	6	2	0	0.86	2	2	0	2	0.67	0	6	0.00	26	3	10	0.90	0.72	0.80
061	9	3	0	0.86	2	4	0	3	0.50	12	2	8	0.71	0	0	4	8	0.00	6	6	0.50	56	41	18	0.58	0.76	0.65
063	4	1	0	0.89	2	2	0	0	0.67	4	0	0	1.00	0	0	2	2	0.00	0	4	0.00	28	11	5	0.72	0.85	0.78
064	9	2	1	0.86	1	0	1	7	0.67	10	2	0	0.91	0	0	2	8	0.00	8	2	0.80	54	18	10	0.75	0.84	0.79
065	6	0	1	0.92	1	0	0	5	1.00	14	0	0	1.00	2	0	0	12	1.00	6	8	0.43	42	0	5	1.00	0.89	0.94
069	3	1	0	0.86	0	1	2	0	0.00	0	0	4	0.00	0	0	0	0	0.00	0	1	0.00	20	15	14	0.57	0.59	0.58
070	6	0	0	1.00	1	3	1	1	0.33	2	2	0	0.67	0	0	0	2	0.00	0	2	0.00	34	12	10	0.74	0.77	0.76

References

1. Propp, V. Morphologie des Märchens [1928]. In *Hrsg. Karl Eimermacher. Übs. Christel Wendt*; Carl Hanser Verlag: München, Germany, 1972.
2. Waumans, M.C.; Nicodème, T.; Bersini, H. Topology Analysis of Social Networks Extracted from Literature. *PLoS ONE* **2015**, *10*, e0126470. [CrossRef] [PubMed]
3. Labatut, V.; Bost, X. Extraction and Analysis of Fictional Character Networks: A Survey. *ACM Comput. Surv.* **2019**, *52*, 89. [CrossRef]
4. Elson, D.; Dames, N.; Mckeown, K. Extracting Social Networks from Literary Fiction. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 138–147.
5. Agarwal, A.; Kotalwar, A.; Rambow, O. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan, 14–19 October 2013; pp. 1202–1208.
6. Ardanuy, M.C.; Sporleder, C. Structure-based clustering of novels. In Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL), Gothenburg, Sweden, 26–27 April 2014; pp. 31–39.
7. Trovati, M.; Brady, J. Towards an automated approach to extract and compare fictional networks: An initial evaluation. In Proceedings of the 2014 25th International Workshop on Database and Expert Systems Applications, Munich, Germany, 1–5 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 246–250.
8. Dekker, N.; Kuhn, T.; van Erp, M. Evaluating named entity recognition tools for extracting social networks from novels. *PeerJ Comput. Sci.* **2019**, *5*, e189. [CrossRef] [PubMed]
9. Edwards, M.; Tuke, J.; Roughan, M.; Mitchell, L. The one comparing narrative social network extraction techniques. In Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), The Hague, The Netherlands, 7–10 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 905–913.
10. Krug, M. *Techniques for the Automatic Extraction of Character Networks in German Historic Novels*; Bayerische Julius-Maximilians-Universitaet Wuerzburg (Germany): Würzburg, Germany, 2020.
11. Agarwal, D.; Vijay, D. Genre classification using character networks. In Proceedings of the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 6–8 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 216–222.
12. Marienberg-Milikowsky, I.; Vilenchik, D.; Krohn, N.; Kenzi, K.; Portnikh, R. *An Experimental Undogmatic Modelling of (Hebrew) Literature: Philology, Literary Theory, and Computational Methods*; Graduate School Practices of Literature: Münster, Germany, 2022. [CrossRef]
13. Perri, V.; Qarkaxhija, L.; Zehe, A.; Hotho, A.; Scholtes, I. One Graph to Rule them All: Using NLP and Graph Neural Networks to analyse Tolkien's Legendarium. *arXiv* **2022**, arXiv:2210.07871.
14. Zhong, L.; Wu, J.; Li, Q.; Peng, H.; Wu, X. A Comprehensive Survey on Automatic Knowledge Graph Construction. *ACM Comput. Surv.* **2023**, *56*, 94. [CrossRef]
15. Schmidt, D.; Zehe, A.; Lorenzen, J.; Sergel, L.; Düker, S.; Krug, M.; Puppe, F. The FairyNet Corpus—Character Networks for German Fairy Tales. In Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, Online, Punta Cana, Dominican Republic, 11 November 2021; pp. 49–56. [CrossRef]
16. Krug, M.; Puppe, F.; Reger, I.; Weimer, L.; Macharowsky, L.; Feldhaus, S.; Jannidis, F. Description of a Corpus of Character References in German Novels—DROC [Deutsches ROman Corpus]. In Proceedings of the DARIAH-DE Working Papers, DARIAH-DE, Göttingen, Germany, 2018.
17. Bamman, D.; Lewke, O.; Mansoor, A. An Annotated Dataset of Coreference in English Literature. In Proceedings of the Twelfth Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 44–54.
18. Weischedel, R.; Pradhan, S.; Ramshaw, L.; Palmer, M.; Xue, N.; Marcus, M.; Taylor, A.; Greenberg, C.; Hovy, E.; Belvin, R.; et al. Ontonotes release 4.0. In *LDC2011T03*; Linguistic Data Consortium: Philadelphia, PA, USA, 2011.
19. Lee, H.; Chang, A.; Peirsman, Y.; Chambers, N.; Surdeanu, M.; Jurafsky, D. Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. *Comput. Linguist.* **2013**, *39*, 885–916. [CrossRef]
20. Schmidt, D.; Krug, M.; Puppe, F. Adapting Coreference Algorithms to German Fairy Tales. In Proceedings of the DHd 2022, Potsdam, Germany, 7–11 March 2022; Geierhos, M., Trilcke, P., Börner, I., Seifert, S., Busch, A., Helling, P., Eds.; Zenodo: Potsdam, Germany, 2022. [CrossRef]
21. Lee, K.; He, L.; Zettlemoyer, L. 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), New Orleans, LA, USA, 1–6 June 2018; pp. 687–692. [CrossRef]
22. Toshniwal, S.; Wiseman, S.; Ettinger, A.; Livescu, K.; Gimpel, K. Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 8519–8526. [CrossRef]
23. Liu, T.; Jiang, Y.E.; Monath, N.; Cotterell, R.; Sachan, M. Autoregressive Structured Prediction with Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 993–1005. [CrossRef]

24. Jannidis, F.; Reger, I.; Weimer, L.; Krug, M.; Toepfer, M.; Puppe, F. Automatische Erkennung von Figuren in deutschsprachigen Romanen. In Proceedings of the DhD, Graz, Austria, 2015. Available online: http://www.jannidis.de/publikationen.html (accessed on 22 October 2024).
25. Lee, K.; He, L.; Lewis, M.; Zettlemoyer, L. End-to-end Neural Coreference Resolution. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 188–197. [CrossRef]
26. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237. [CrossRef]
27. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
28. Xu, L.; Choi, J.D. Revealing the Myth of Higher-Order Inference in Coreference Resolution. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 8527–8533. [CrossRef]
29. Needleman, S.B.; Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **1970**, *48*, 443–453. [CrossRef] [PubMed]
30. Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; ANUBHAI, R.; dos Santos, C.N.; Xiang, B.; Soatto, S. Structured Prediction as Translation between Augmented Natural Languages. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
31. Le, N.T.; Ritter, A. Are Large Language Models Robust Zero-shot Coreference Resolvers? *arXiv* **2023**, arXiv:2305.14489.
32. Krug, M.; Jannidis, F.; Reger, I.; Macharowsky, L.; Weimer, L.; Puppe, F. Attribuierung direkter Reden in deutschen Romanen des 18.-20. Jahrhunderts. Methoden zur Bestimmung des Sprechers und des Angesprochenen. In Proceedings of the DHd, Leipzig, Germany, 7–12 March 2016.
33. Gansner, E.R.; North, S.C. An open graph visualization system and its applications to software engineering. *Softw. Pract. Exp.* **2000**, *30*, 1203–1233. [CrossRef]
34. Vilain, M.; Burger, J.; Aberdeen, J.; Connolly, D.; Hirschman, L. A model-theoretic coreference scoring scheme. In Proceedings of the 6th conference on Message understanding, Columbia, MD, USA, 6–8 November 1995; Association for Computational Linguistics: Stroudsburg, PA, USA, 1995; pp. 45–52.
35. Bagga, A.; Baldwin, B. Algorithms for scoring coreference chains. In Proceedings of the The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, Granada, Spain, 28–30 May 1998; Volume 1, pp. 563–566.
36. Luo, X. On coreference resolution performance metrics. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 25–32.
37. Denis, P.; Baldridge, J. Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. In Proceedings of the Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, NY, USA, 22–27 April 2007; Proceedings of the Main Conference; pp. 236–243.
38. Recasens, M.; Hovy, E. BLANC: Implementing the Rand index for coreference evaluation. *Nat. Lang. Eng.* **2011**, *17*, 485–510. [CrossRef]
39. Moosavi, N.S.; Strube, M. 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: Long Papers), Berlin, Germany, 7–12 August 2016; Volume 1, pp. 632–642.
40. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
41. Schmid, H.; Laws, F. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK, 18–22 August 2008; pp. 777–784.
42. Zehe, A.; Konle, L.; Dümpelmann, L.K.; Gius, E.; Hotho, A.; Jannidis, F.; Kaufmann, L.; Krug, M.; Puppe, F.; Reiter, N.; et al. Detecting scenes in fiction: A new segmentation task. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 3167–3177.
43. Reiter, N.; Sieker, J.; Guhr, S.; Gius, E.; Zarrieß, S. Exploring text recombination for automatic narrative level detection. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; pp. 3346–3353.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.