Automatic Narrative Knowledge Base Generation

Robert Morain
Computer Science Department
Brigham Young University
Provo, UT 84602
rmorain2@byu.edu

Rafael Pérez y Pérez
Universidad Autónoma Metropolitana,
Cuajimalpa, México
Center for Digital Narrative,
University of Bergen, Norway
rperez@cua.uam.mx

Dan Ventura
Department of Computer Science
Brigham Young University
Provo, UT 84602
ventura@cs.byu.edu

Abstract

Traditional symbolic CC systems like MEXICA often require the creation of handcrafted knowledge bases. In order to advance the development of the MEXICA project, this paper introduces methods for the automatic creation of a knowledge base of short stories. The methods include a series of requests to Deepseek's R1 model to extract relevant structured data from a narrative, using the model to validate and correct the extracted data, and then parsing the structured data and formatting it for the required MEXICA artifacts. This process is validated by evaluating the quality of the extracted narrative data through a human survey. The results show that the process was effective at extracting conceptually accurate structured narrative data from a set of test stories. This work unblocks a significant bottleneck for MEXICA which is necessary for the system to advance to the next level of understanding narrative generation and demonstrates a unique symbiosis between symbolic and generative AI systems.

Introduction

The rapid development of generative AI raises important questions about the future of traditional symbolic computational creativity (CC) systems; for example, how can generative and symbolic AI complement each other by mitigating each other's weaknesses and amplifying each other's strengths (Veale 2024). Although neither symbolic AI nor generative AI alone can yet be considered truly creative, integrating the two paradigms may offer a path toward more advanced CC systems. However, integrating symbolic and generative approaches presents significant challenges as it requires reconciling two fundamentally different modeling paradigms. At the same time, this challenge offers an opportunity to reflect on the core strengths and limitations of each paradigm and to explore new methods for their complementary integration (Pérez y Pérez and Sharples 2023).

As an initial step toward this goal, this paper focuses on the task of automatic knowledge base generation, a critical component of many traditional symbolic CC systems (Ventura 2017). However, building these knowledge bases is difficult and often requires manual effort from domain experts, especially in creative domains.

MEXICA is a symbolic computational model of the creative writing process capable of automatic narrative generation (Pérez y Pérez 1999; Pérez y Pérez and Sharples 2001). A critical area for improvement in MEXICA is its knowledge base, which stores structured information about story events, including the sequence of actions and their associated preconditions and postconditions. Currently, this knowledge base is created manually by experts, a process that is both challenging and time-consuming. As MEXICA has evolved, its increasingly ambitious goals for understanding narrative generation necessitate the development of more powerful knowledge generation methodologies.

In this paper, we introduce methods for automatically creating a knowledge base of structured narrative artifacts (Valls-Vargas, Zhu, and Ontañón 2017; Chambers and Jurafsky 2008) using a pipeline facilitated by Deepseek's R1 model, a state-of-the-art mixture-of-experts model employing chain-of-thought reasoning (DeepSeek-AI et al. 2025). This pipeline automatically parses a story and generates the necessary artifacts for MEXICA using human-engineered prompts and artifact verification. The pipeline extracts actions, preconditions, and postconditions from a story and compiles them into a structured JSON object. This object is then parsed to produce two outputs: a Definition of Previous Stories (DPS) file detailing the event sequence, and a Primitive Action Definition (PAD) file specifying the structure of each action.

The primary goals of the pipeline are to create artifacts that are syntactically correct, conceptually accurate to the plot of the story, and consistent with the MEXICA story description paradigm. The pipeline is evaluated in two ways. First, a human survey consisting of 72 participants evaluates the conceptual accuracy of extracted information from five short story summaries (primarily fairy tales). Second, the generated artifacts for each story are provided as input to MEXICA and corrected if necessary. The number of changes required to conform to MEXICA's rules is tracked to provide a fine-grained measure of how syntactically correct the artifacts are as well as the artifacts fidelity to the MEXICA story description paradigm.

Preliminary results indicate that the pipeline is moderately successful in achieving these goals. In general, this work contributes to the development of more powerful research tools for MEXICA by introducing methods for automatic narrative knowledge base generation and evaluation. It also illustrates how symbolic and generative AI systems can work together in a complementary fashion. All of the code, analysis, prompts, and generated artifacts for this paper can be found on GitHub[1].

Background

The MEXICA story paradigm decomposes narratives into a series of actions focused on the emotional relationships between characters and the development of tension. This sequence of actions is formalized and is stored in a DPS file. Each action is formally defined and is stored in a PAD file. The full details for both of these formal descriptions can be found in (MEXICA).

A DPS file encodes a story as a sequence of actions, each following a ‘subject‘ ‘action‘ ‘object‘ structure to describe interactions between characters. The PAD file defines each action's preconditions and postconditions, where preconditions specify requirements needed to perform the action and postconditions describe the effects of the action on the story world. MEXICA requires at least one postcondition for every action to build narrative tension. Preconditions and postconditions fall into two main categories: emotional links and tensions. Emotional links model directed relationships between characters with different types and magnitudes of affection, while tensions represent conditions such as life or health being at risk, imprisonment, or their resolution.

For this paper, the LLM used is Deepseek's R1 70b model, running locally via Ollama. R1 is a state-of-the-art mixture-of-experts model that uses chain-of-thought reasoning. The 70b variant was chosen to balance inference speed, memory efficiency, and model complexity. Although the larger R1 671b variant was tested, it proved too slow for local execution. Although not evaluated in this study, other models could have been used. All experiments were conducted on a single NVIDIA H200 GPU.

Pipeline Design

The primary challenge in processing a story to produce the DPS and PAD files is ensuring the LLM receives enough information to understand and correctly execute the prompt. This is particularly difficult because the MEXICA story description paradigm is nuanced and often challenging even for human annotators to apply consistently across diverse stories. To address this complexity, key concepts from the MEXICA paradigm must be conveyed clearly and in a way that the LLM can reliably interpret. Additionally, it is crucial to limit the amount of new information introduced at each stage to minimize model distraction.

To achieve these design goals, we developed a multi-stage pipeline that processes a story and produces the corresponding DPS and PAD artifacts. The stages are:

1. Identify story actions
2. Identify or infer emotional link preconditions
3. Identify tension preconditions
4. Identify or infer postconditions
5. Verify a JSON object containing the extracted story data
6. Parse JSON to create DPS file
7. Parse JSON to create PAD file

[1]https://github.com/rmorain/LLMEXICA

At each stage, carefully engineered prompts—providing background information and task-specific instructions—are used to guide the LLM through producing the required intermediate artifacts within a single conversational thread.

Identify story actions

The first stage identifies the primitive actions performed by the main characters, focusing on emotional relationships and story tensions. A prompt frames the LLM as a narrative analysis expert and provides necessary context about emotional links and tensions. The prompt also includes instructions for how to construct a JSON object containing keys for the ‘action‘ label, the ‘subject‘ character performing the action, and the ‘object‘ character receiving the action directly or indirectly. Also, because the ‘subject‘ and ‘object‘ keys are optional, there is also a key for the number of characters involved in the action. An example is also provided to specify the expected style of the ‘action‘ label. Then, the prompt includes instructions to verify the JSON object to make sure the selected actions accurately reflect the plot of the story. Finally, the story is appended to the end of the prompt.

Identify or infer emotional link preconditions

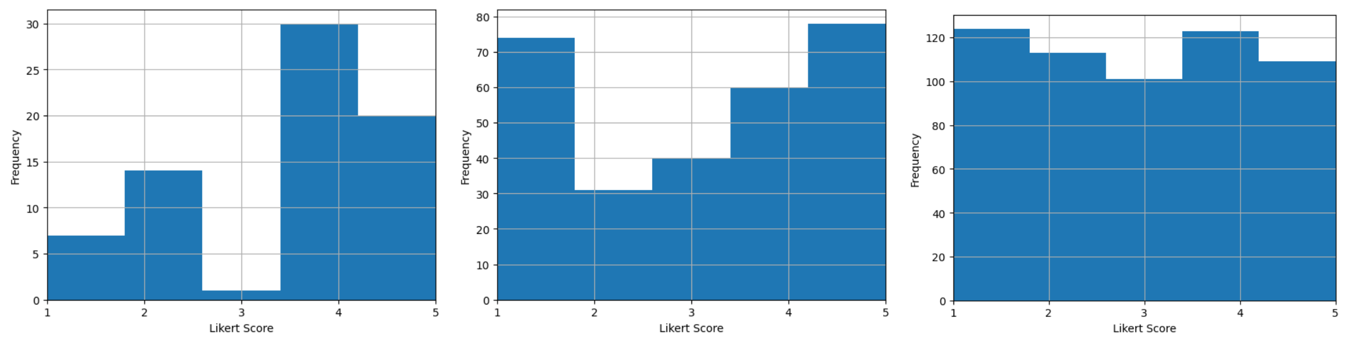
The second stage identifies or infers the emotional link preconditions associated with each previously extracted action. More details are provided to define what it means to be a precondition and specifying the types and magnitude range of emotional links. Specific instructions are given on how to update the JSON object to include keys for ‘preconditions‘, ‘emotional_links‘, ‘type‘, ‘from‘, and ‘to‘. MEXICA expects emotional links to reference variables ’a’ or ’b’ (depending on the number of characters involved in the action) to specify which character is the source or target of an emotional link. ’a’ refers to the ‘subject‘ character and ’b’ refers to the ‘object‘ character.

Identify tension preconditions

The third stage of the pipeline is responsible for identifying or inferring the tension preconditions for each action. Similar to the previous stage, more details on what tensions are are provided in the prompt along with specific instructions for how to modify the intermediate JSON object from the previous stage.

Identify or infer postconditions

The fourth stage of the pipeline is responsible for identifying or inferring postconditions. Because the conversation history already contains explanations about emotional links and tensions, it is not necessary to explain these concepts again. Instead, postconditions are defined, the complementary types of tensions that are specific to postconditions (life normal, health normal, and prisoner freed) are emphasized and specific instructions on how to update the JSON object are provided.



(a) Actions (b) Preconditions (c) Postconditions

Figure 1: Histograms of participant ratings for actions, preconditions, and postconditions across all stories using a five-point Likert scale (1 = strongly disagree, 5 = strongly agree). Actions received the highest ratings (M = 3.583, SD = 1.340; median = 4, mode = 4), followed by preconditions (M = 3.13, SD = 1.57; median = 3, mode = 5), and postconditions (M = 2.97, SD = 1.43; median = 3, mode = 1). The overall average across all items was 3.06.

Verify a JSON object containing the extracted structured story data

The verification stage takes place in two steps. First, the LLM is prompted to validate the intermediate JSON object stored within the conversation history. The prompt contains a wide variety of mistakes to avoid and as well as instruction to avoid any other kind of logical error. Next, the JSON is parsed from the LLM response. The parsing process provides additional checks such as for any action that lacks a postcondition, emotional links that are missing a source or a target, or tensions without an affected character. When these errors are detected, the LLM is prompted to regenerate the JSON object to address the error.

Parse JSON to create DPS and PAD files

Finally, the validated JSON object is parsed to generate the DPS and PAD files. This process is straightforward and can be referenced on Github. Notably, the structured data in the JSON could be transformed into a variety of formats for compatibility with other systems. It is important to note that early experiments prompting the LLM to directly generate DPS and PAD files were unreliable, as the outputs often failed to accurately reflect the structured information.

Evaluation

The pipeline is evaluated using a human survey and by directly analyzing the generated artifacts and correcting them to be usable by MEXICA, when necessary.

Survey Methodology

Five stories were selected to test the narrative knowledge base pipeline. Four classic fairy tales, “Goldilocks and the Three Bears" (Steel 1922), “Hansel and Gretel" (Grimm and Grimm 1812), “Jack and the Beanstalk" (Steel 1918), and “Little Red Riding Hood" (Perrault 1923), and one story generated by MEXICA, “Jaguar Knight". The primary goal of the survey is to evaluate how well the identified actions, along with their respective preconditions and postconditions, make sense within the context of the story. To do this, survey participants are asked to do the following:

1. Read a randomly selected story summary
2. Rate how well the main characters’ actions reflect the story plot
3. Review 5–6 actions with their preconditions and postconditions
4. Rate agreement with precondition statements (e.g., “Before Princess heals Jaguar Knight, it is likely that Princess likes Jaguar Knight”)
5. Rate postcondition statements similarly, replacing “Before” with “After”

Questions about the preconditions and postconditions are generated automatically by parsing the JSON object of story actions. The survey was distributed online to non-experts via social media.

Artifact Analysis and Correction

To evaluate the usability of the generated DPS and PAD files with MEXICA, it is necessary to attempt to provide the files as input to the system. If a file has errors it is corrected and then the number of unchanged lines can be used to measure the quality of the original artifact in the following manner:

$$\text{Artifact quality} = \frac{\text{unchanged lines}}{\text{total lines in generated artifact}} \tag{1}$$

This preliminary metric suffers from several limitations. For example, it does not take into account insertions into the corrected file. To account for this, counts for unchanged, inserted, deleted, and moved lines are included in the table.

Results

Survey results Seventy-two participants completed the survey (see Figure 1). Responses were collected using a five-point Likert scale (1 = strongly disagree, 5 = strongly

Story	% Unchanged	Unchanged	Inserted	Deleted	Moved
Jaguar Knight	100.00%	11	0	0	0
Goldilocks	100.00%	10	0	0	0
Hansel and Gretel	41.18%	7	7	10	0
Jack and the Beanstalk	31.25%	5	11	11	0
Little Red Riding Hood	61.71%	11	7	7	0
Mean unchanged generated lines	61.11%				

Table 1: Results from evaluating generated DPS artifacts using an analysis and correction procedure. The table reports the percentage of unchanged lines as well as the counts of each type of correction made to produce the final artifact. Overall, 61.11% of lines in the generated DPS artifacts were correct.

Story	% Unchanged	Unchanged	Inserted	Deleted	Moved
Jaguar Knight	58.62%	51	15	32	4
Goldilocks	52.08%	25	6	23	0
Hansel and Gretel	26.61%	33	20	86	5
Jack and the Beanstalk	93.75%	75	31	3	2
Little Red Riding Hood	44.71%	38	14	47	0
Mean unchanged generated lines	52.36%				

Table 2: Results from evaluating generated PAD artifacts using an analysis and correction procedure. The table reports the percentage of unchanged lines as well as the counts of each type of correction made to produce the final artifact. Overall, 52.36% of lines in the generated PAD artifacts were correct.

agree). Actions received the highest ratings (M = 3.583, SD = 1.340; median = 4, mode = 4), followed by preconditions (M = 3.13, SD = 1.57; median = 3, mode = 5), and postconditions (M = 2.97, SD = 1.43; median = 3, mode = 1). The overall average across all items was 3.06.

Some stories received higher ratings than others. “Jack and the Beanstalk" got the highest overall score across actions, preconditions, and postconditions (3.47), followed by “Jaguar Knight" (3.41), “Hansel and Gretel" (3.04), “Little Red Riding Hood" (2.64), and “Goldilocks and the Three Bears" (2.39). Note that the pipeline did not identify any preconditions for any actions for “Goldilocks and the Three Bears", “Jack and the Beanstalk", and “Little Red Riding Hood".

Artifact Analysis and Correction Results Table 1 and Table 2 show that across all stories 61.11% of the lines in DPS files are correct and 52.36% of lines in PAD files are correct. “Jaguar Knight" was the story with the most correct lines for both the DPS and PAD files.

Discussion

These preliminary results suggest that the methods presented in this paper provide a foundation for automatic narrative knowledge base creation. While survey results show that participants generally agreed that the identified actions accurately reflect the plot of the story, the extraction of sensible preconditions and postconditions remains an area for further improvement. In particular, there are some instances where the model fails to identify the correct direction of emotional links and tensions. The artifact analysis also reveals other areas for improvement to syntax and consistency with the MEXICA story description framework. For example, PAD files should not have repeated action definitions and DPS files should include relevant character movements throughout the story world.

Currently, these methods require the LLM to generate all preconditions and postconditions for each action in a single step, with verification occurring only at the end. In future work, it may be beneficial to better leverage the strengths of the LLM by breaking down this process on an action-by-action basis. By the same logic, the verification process may be improved by converting the structured information back into natural language and asking the LLM to verify it. It will also be important to include information about the movement of the characters throughout the story world in the DPS and PAD files.

This work marks a first step toward merging the strengths of symbolic and generative AI systems and should spark broader discussions about their associated advantages and limitations. It raises important questions, such as: “To what extent do LLMs truly understand human emotions?” and “Why is it so challenging (for both humans and LLMs) to translate descriptions of emotional relationships into the simple rules defined by the MEXICA paradigm?” A central challenge in the domain of narrative understanding is that stories operate not only through language, but also through layers of emotion and conflict. Because symbolic and generative systems follow different paradigms of story understanding, integrating them in a complementary way offers a promising path forward for deeper narrative understanding and symbolic CC systems.

Acknowledgements

This work was partially funded by the National Council of Humanities, Science and Technology in México, project CF-2023-I-312 and by the Research Council of Norway through its Centers of Excellence Scheme, Project No. 332643.

Prompts

This section presents a selection of the prompts used in the knowledge base generation pipeline. The full set of prompts is available on GitHub.

Listing 1: Prompt to extract story actions

```
You are a narrative analysis expert that
systematically identifies and interprets
actions, preconditions, and effects (called
postconditions) within stories, contributing
 to a structured understanding of a
narrative. You are primarily focused on
actions that relate to the emotional
relationships between characters (called
emotional links) and actions that build
tension within the narrative (called
tensions).

You are focused on specific types of
emotional links. By default, consider two
types of emotional links: `non-romantic` and
 `romantic`. `non-romantic` refers to how
much one character likes another character
in a platonic (non-romatnic) sense. Most
emotional links between characters that are
not in a relationship will be of this type.
The `romantic` emotional link type refers to
 a type of romantic love between the
characters.

You are also focused on specific types of
tensions. By default, consider the following
 tension types:

1. `character_dead`
2. `life_at_risk`
3. `health_at_risk`
4. `prisoner`

`character_dead` means a character has died.
 `life_at_risk` means a character's life is
at risk. `health_at_risk` means a character'
s health is at risk. `prisoner` means a
character is in prison or detained in some
way.
Analyze the given story and extract the
essential actions from the main characters.
Focus on actions that relate to the
emotional links between characters and the
tensions in the narrative.
Keep the `action` values simple and focused
on the abstract action being performed
rather than a specific character performing
the action. For example, if Goldilocks eats
Baby Bear's porridge, label the action as '
eats_porridge' with 'Goldilocks' as the
subject and 'Baby Bear' as the indirect
object rather than labeling the action as '
goldilocks_eats_baby_bear_porridge'.
Organize the actions in chronological order
and in JSON format . The JSON should have an
 `action` key for each action. The value for
 each action should be as simple and general
 as possible so that it can be reused in
other stories, avoid character names, and be
 in Snake_case. Each `action` should have a
key for the number of characters involved in
 the action called `n_characters`, a key
called `subject` for the character
performing the action, and a key called `
object` for the character receiving the
action. If the action lacks a `subject` or `
object` store a value of `-` in the key.
Only include actions where the `subject` and
 `object` refer to characters and not
inanimate objects. For now, only include
these specified keys in the JSON object.
Make sure that `n_characters` is consistent
with the presence of the `subject` and `
object` characters. The same character may
be both the `subject` and the `object`
character if the character is performing an
action on themself.

Once, the JSON object is created verify that
 the actions accurately reflect events
described in the story.

Here is the story:
<STORY>
```

Listing 2: Prompt to extract tension preconditions

```
For each action, identify or infer the
preconditions related to tensions. A
precondition is a requirement that needs to
be satisfied in order for a character to
perform a specific action. These
requirements take the form of either an
emotional link or a tension. A tension that
is a precondition is a tension that should
exist in order for taking the action to make
 sense.

Identify or infer the preconditions that
relate to tensions by following the
instructions below.

Instructions for each action:
1. In the `preconditions` object create a
key called `tensions`
2. Assign an array to the `tensions` key
containing the identified or inferred
tensions.
A. If there are no required `tensions`,
leave the array empty
4. Each `tension` is an object with keys `
type`, `from`, and `to`
A. `type` contains the type of the `tension`
 as defined previously.
B. `from` contains either values 'a', 'b',
'-', '' to indicate the character that is
the source of the tension. 'a' refers to the
 `subject` performing the action, 'b' refers
 to the `object` receiving the action, '-'
refers to no character, and '*' refers to
any character
C. `to` contains either values 'a' or 'b' to
 indicate the character that is the
recipient of the tension. 'a' refers to the
`subject` performing the action, 'b' refers
to the `object` receiving the action, '-'
refers to no character, and '*' refers to
any character

Return this new JSON object.
```

References

[Chambers and Jurafsky 2008] Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Anniversary Meeting of the Association for Computational Linguistics*, 789–797. Association for Computational Linguistics.

[DeepSeek-AI et al. 2025] DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; Zhang, X.; Yu, X.; Wu, Y.; Wu, Z. F.; Gou, Z.; Shao, Z.; Li, Z.; Gao, Z.; Liu, A.; Xue, B.; Wang, B.; Wu, B.; Feng, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; Dai, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Dai, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Bao, H.; Xu, H.; Wang, H.; Ding, H.; Xin, H.; Gao, H.; Qu, H.; Li, H.; Guo, J.; Li, J.; Wang, J.; Chen, J.; Yuan, J.; Qiu, J.; Li, J.; Cai, J. L.; Ni, J.; Liang, J.; Chen, J.; Dong, K.; Hu, K.; Gao, K.; Guan, K.; Huang, K.; Yu, K.; Wang, L.; Zhang, L.; Zhao, L.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Wang, M.; Li, M.; Tian, N.; Huang, P.; Zhang, P.; Wang, Q.; Chen, Q.; Du, Q.; Ge, R.; Zhang, R.; Pan, R.; Wang, R.; Chen, R. J.; Jin, R. L.; Chen, R.; Lu, S.; Zhou, S.; Chen, S.; Ye, S.; Wang, S.; Yu, S.; Zhou, S.; Pan, S.; Li, S. S.; Zhou, S.; Wu, S.; Ye, S.; Yun, T.; Pei, T.; Sun, T.; Wang, T.; Zeng, W.; Zhao, W.; Liu, W.; Liang, W.; Gao, W.; Yu, W.; Zhang, W.; Xiao, W. L.; An, W.; Liu, X.; Wang, X.; Chen, X.; Nie, X.; Cheng, X.; Liu, X.; Xie, X.; Liu, X.; Yang, X.; Li, X.; Su, X.; Lin, X.; Li, X. Q.; Jin, X.; Shen, X.; Chen, X.; Sun, X.; Wang, X.; Song, X.; Zhou, X.; Wang, X.; Shan, X.; Li, Y. K.; Wang, Y. Q.; Wei, Y. X.; Zhang, Y.; Xu, Y.; Li, Y.; Zhao, Y.; Sun, Y.; Wang, Y.; Yu, Y.; Zhang, Y.; Shi, Y.; Xiong, Y.; He, Y.; Piao, Y.; Wang, Y.; Tan, Y.; Ma, Y.; Liu, Y.; Guo, Y.; Ou, Y.; Wang, Y.; Gong, Y.; Zou, Y.; He, Y.; Xiong, Y.; Luo, Y.; You, Y.; Liu, Y.; Zhou, Y.; Zhu, Y. X.; Xu, Y.; Huang, Y.; Li, Y.; Zheng, Y.; Zhu, Y.; Ma, Y.; Tang, Y.; Zha, Y.; Yan, Y.; Ren, Z. Z.; Ren, Z.; Sha, Z.; Fu, Z.; Xu, Z.; Xie, Z.; Zhang, Z.; Hao, Z.; Ma, Z.; Yan, Z.; Wu, Z.; Gu, Z.; Zhu, Z.; Liu, Z.; Li, Z.; Xie, Z.; Song, Z.; Pan, Z.; Huang, Z.; Xu, Z.; Zhang, Z.; and Zhang, Z. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. https://arxiv.org/abs/2501.12948.

[Grimm and Grimm 1812] Grimm, J., and Grimm, W. 1812. Hänsel und gretel. In *Kinder- und Hausmärchen*, volume 1. Berlin: Realschulbuchhandlung. KHM 15.

[Pérez y Pérez and Sharples 2023] Pérez y Pérez, R., and Sharples, M. 2023. *An introduction to narrative generators: how computers create works of fiction*. Oxford University Press.

[Pérez y Pérez 1999] Pérez y Pérez, R. 1999. *MEXICA: a computer model of creativity in writing.* Ph.D. Dissertation, University of Sussex.

[Perrault 1923] Perrault, C. 1923. Little red riding-hood. In *Tales of Past Times Written for Children*. New York: E.P. Dutton and Co. 6–8.

[Pérez y Pérez and Sharples 2001] Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

[Steel 1918] Steel, F. A. 1918. Jack and the beanstalk. In *English Fairy Tales*. London: Macmillan and Co., Limited. 136–153.

[Steel 1922] Steel, F. A. 1922. The story of the three bears. In *English Fairy Tales*. Macmillan and Co.

[Valls-Vargas, Zhu, and Ontañón 2017] Valls-Vargas, J.; Zhu, J.; and Ontañón, S. 2017. Towards automatically extracting story graphs from natural language stories. In *AAAI Workshops*.

[Veale 2024] Veale, T. 2024. From symbolic caterpillars to stochastic butterflies: Case studies in re-implementing creative systems with LLMs. In *Proceedings of the International Conference on Computational Creativity*, 236–244. Association for Computational Creativity.

[Ventura 2017] Ventura, D. 2017. How to build a CC system. In *Proceedings of the International Conference on Computational Creativity*, 253–260. Association for Computational Creativity.