Narrative Text Generation from Abductive Interpretations using Axiom-specific Templates*

Andrew S. Gordon and Timothy S. Wang

University of Southern California, Los Angeles, CA USA
gordon@ict.usc.edu, wangtimo@usc.edu

Abstract. Structured story graphs have proven to be useful for representing content in pipelines for automated interpretation and narration. Recent progress on interpretation using logical abduction has made it possible to construct these representations automatically, and several methods for converting these structures into narrative text have been proposed. In this paper, we describe a technical approach to narrative text generation from structured story graphs that prioritizes simplicity and ease-of-use, employing full-sentence templates associated with the specific axioms used to construct graphs during the interpretation process. We evaluate our approach using the TriangleCOPA benchmark for narrative interpretation and text generation, comparing our results to human-authored narratives and to the results of previous work.

Keywords: Automated interpretation · Text Generation · Logical Abduction.

1 Introduction

A popular approach in research on narrative text generation is to first represent story content formally as symbolic structures, which are then converted into natural language text using a variety of approaches. Elson [2] proposed the *Story Intention Graph* as a formalism for encoding the interpretation of stories as symbolic causal structures, reminiscent of the *Causal Network Model* of psychologists Trabasso and van den Broek [18]. Using a software tool for hand-authoring these representations [3], different research teams have succeeded in authoring sizable corpora of story representations, and devising novel algorithms for converting these representations into fluent natural-language texts [2][12]. Although these text-generation systems are typically quite sophisticated in their use of numerous grammatical subsystems and lexical resources, the overall lesson from this line of research is that fluent narratives can be automatically generated if a rich structured representation of the story can be provided by some upstream interpretation process.

* The project or effort depicted was or is sponsored by the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, and that the content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

Current systems that attempt to automate the interpretation process have their roots in the work of Hobbs et al. [10], who proposed that language interpretation could be cast as a problem of logical abduction. In logic-based reasoning systems, abduction is viewed as a search for the optimal set of assumptions that, if true, would logically entail a set of observations, given a knowledge base of axioms. Hobbs et al. [10] proposed a cost-based method for finding and ranking solutions, where the initial costs assigned into input observations are transferred to antecedents by back-chaining on definite clauses in the knowledge base, i.e., *weighted abduction*. Gordon [4] devised a probabilistic alternative to weighted abduction, *etcetera abduction*, where the conditional probability of the consequent in a definite clause, given the antecedent, is reified as *etcetera literals* [9] included in the antecedents of every knowledge base axiom. Back-chaining from observations, solutions consisting entirely of etcetera literals can be ranked by their joint probability. Although finding an optimal solution via logical abduction requires an intractable combinatorial search process, Gordon [5] devised an incremental algorithm for etcetera abduction capable of handling large interpretation problems.

While the idea of interpretation as logical abduction grew out of computational linguistics, its applicability to narrative interpretation, more broadly, has been demonstrated in several previous efforts. Gordon [4] applied etcetera abduction to answer commonsense interpretation problems in the TriangleCOPA benchmark [13], consisting of 100 micro-narratives involving three characters. Gordon [5] applied incremental etcetera abduction to observable events in the interpretation of the Heider-Simmel film [7], a narrative that is ubiquitously used as a stimulus in social science research. Gordon and Spierling [6] demonstrated how etcetera abduction could be used for creative narrative interpretation, in the context of a storytelling party game. In each of these efforts, etcetera abduction produces structured story graphs than can serve as input to downstream narrative text generation systems.

In systems that convert structured representations into natural language text, much of the linguistic complexity arises when composing sentences from multiple nodes in the story graph. For any single node in the graph, a trivial template system is sufficient to produce a fluent sentence or clause. When composing sentences from content across connected nodes, however, the text generation system must be sensitive to a myriad of linguistic concerns, from lexical choice in subordinate clauses, unambiguous use of pronouns, and conjoining noun phrases that share a semantic role in the output sentence. Ahn et al. [1] showed previously how these complexities could be mitigated using the approach of over-generating and ranking. In their method, many possible grammatical rules for combining clauses from connected nodes are exhaustively applied to generate candidate sentences, which are then ranked for fluency using a probabilistic syntactic parser.

In this paper, we explore an alternative approach to this problem that avoids the grammatical complexity of combining phrases altogether, by associating textual templates directly with the axioms used to assemble the story graph in the first place. Using axiom-specific templates, we show that a trivial template sys-

tem with simple manipulations for noun phrases is sufficient to generate text from formal story graphs that is as fluent as those produced by previous approaches.

2 Axiom-specific Templates

The basic idea in our narrative text generation approach is to utilize sentence-length templates that are specific to individual knowledge base axioms, rather than trying to assemble grammatical sentences from groups of connected nodes in the structured interpretation graph. The rationale is that the knowledge base axioms used to construct the interpretation graph already identify a coherent set of interrelated nodes (logical literals) during the search process. When an axiom participates in building the interpretation, its constituent literals (and their variable bindings) provide all the necessary information to express the inference as a fluent natural-language sentence.

To illustrate this idea, consider the following knowledge base axiom, used by Gordon [4] to correctly answer question 83 of the TriangleCOPA benchmark.

```
(if (and (attack' ?e1 ?y ?z)
     (like' ?e2 ?x ?z)
     (etc3_angryAt 0.9 ?e1 ?e2 ?e ?x ?y ?z))
(angryAt' ?e ?x ?y))
```

This axiom captures the commonsense idea that if somebody attacks someone that you like, then you are likely to be angry at the attacker. During the interpretation process, this axiom would be used to replace an assumption that unifies with the consequent with the three assumptions in the antecedent, along with the necessary variable substitutions. In a subsequent narration process, a text template can be used to express this inference as a single sentence.

```
Due to a fondness for ?z, ?x was angry at ?y for the attack.
```

In order to correctly substitute the variables in an arbitrary template during narration, the interpretation system must record variable bindings for all uses of each axiom in a given interpretation. Conveniently, etcetera abduction encodes these substitutions in a unique etcetera literal that appears in an axiom's antecedent, e.g., the literal with the predicate `etc3_angryAt` in the example above. Given the etcetera literals that constitute a solution to an interpretation problem, the narration system can select templates and make the necessary variable substitutions by matching the antecedents in a knowledge base of textual templates, such as this one:

```
(if (etc3_angryAt 0.9 ?e1 ?e2 ?e ?x ?y ?z)
    (text "Due to a fondness for" ?z ","
          ?x "was angry at" ?y "for the attack."))
```

If an etcetera literal in the interpretation matches the antecedent of this template with substitutions {?x/BOB, ?y/CARL, ?z/DAVID}, the instantiated consequence of the text template is inferred:

```
(text "Due to a fondness for" DAVID "," BOB "was angry at"
      CARL "for the attack.")
```

The particular characteristics of etcetera abduction, where solutions uniquely identify all axioms that participated in the selected interpretation, afford a simple method for instantiating templates via logical inference. However, the approach is equally applicable to other abductive reasoning methods [11][14], which may require additional bookkeeping to identify the knowledge base axioms that participated in the construction of the selected interpretation. In each case, the basic idea is that the knowledge base axioms are a convenient target for sentence-level templates that express the important interrelationships between nodes in the structured interpretation.

3 Proper Nouns, Common Nouns, and Pronouns

After the substitution of bound variables, a text template in our approach will consist of a list of constants. String constants are the linguistic expressions included by the author of the template, e.g., `"was angry at"`. Symbolic constants identify entities that were either identified as arguments in the original observations provided to the interpretation engine, or introduced in the consequence of a knowledge base axiom, e.g., `CARL`. Skolem constants identify entities whose existence is assumed as a result of the interpretation process, introduced when an existentially quantified variable only appears in the antecedent of a knowledge base axiom, e.g. `$4`.

Our narrative text generation implementation provides a simple mechanism for replacing symbolic constants with strings for either the proper noun or a common noun of the entity, if they are known. Nouns of these types are provided as logical literals alongside the template axioms, as follows:

```
(proper_noun CARL "Carl")
(common_noun CB1 "city bus")
(common_noun GROUP7 "management team")
```

When converting the text literal into an output string, our implementation will swap symbolic constants for any string constants that have been provided, favoring proper nouns over common nouns.

For common nouns, our system precedes the reference with an indefinite article for its first use in a narrative ("a" or "an") and a definite article on subsequent uses ("the"). When a given common noun has previously been used to reference a different entity in the narrative, an additive determine is used ("another"). A default common noun of `"unknown entity"` is used for all Skolem constants.

To enable the use of English pronouns, the pronoun class of any entity can be provided as additional information.

```
(pronouns CARL Masculine)
(pronouns CB1 Neuter)
(pronouns GROUP7 Plural)
```

When the pronoun class of an entity has been provided, our implementation will favor referencing it using a pronoun rather than a proper or common noun, guided by specific directives provided by the template author. Our system supports pronoun substitution for subjects (he), objects (her), dependent possessives (their), independent possessives (hers), and reflexive pronouns (herself), as in the following example:

```
(if (etc3_angryAt 0.9 ?e1 ?e2 ?e ?x ?y ?z)
    (text "Due to" DependentPossessive ?x
          "fondness for" Object ?z ","
          Subject ?x "was angry at" Object ?y
          "for" DependentPossessive ?y "attack."))
```

Our implementation attempts to avoid the introduction of ambiguous pronouns into a narrative, guided by the heuristic pronoun resolution approach of Hobbs [8]. Specifically, we inhibit the introduction of subject and object pronouns when the entity has not yet been mentioned in the current or previous sentence, when its pronoun class is the same as another entity in the current or previous sentence, or when its pronoun class has not yet been revealed to the reader of the narrative via a possessive or reflexive pronoun substitution.

We provide an open-source C# implementation of our text generation approach alongside one of the existing distributions of the etcetera abduction algorithm[1].

4 Evaluation

We evaluate our approach by directly comparing it to the previous work of Ahn et al. [1], where over-generating and ranking is used to assemble content from connected nodes in the story graph into fluent sentences. As in their previous work, we apply our approach to 100 formal interpretations of problems in the TriangleCOPA benchmark.

Modelled after the *Choice of Plausible Alternatives* (COPA) benchmark [17] that is widely used in computational linguistics research, TriangleCOPA was conceived as an end-to-end evaluation for systems that jointly perform the tasks perception, interpretation, and narration. Each of its 100 questions consist of a sentence describing a situation involving three characters and a common setting, a question about the commonsense interpretation of the situation, and two plausible answers, where one was uniformly preferred by human raters. Unlike the original COPA evaluation, each TriangleCOPA question includes an animated video clip of the situation to support computer vision research on action recognition, a formal representation of the question and each alternative to support

[1] https://github.com/asgordon/EtcAbductionCS

a.	Q 5.	*Insults and yelling flew back and forth as the circle and triangle argued loudly in the house. Finally, the triangle had had enough and walked out, slamming the door behind it. As angry as the circle was at the triangle, it was very sad and knew that this may be the end of their relationship.*
	Q 83.	*The circle is trying to get away from the cops and pushes the small triangle to get out of its way. The big triangle feels attacked that the circle pushed its friend and chases after the circle too.*
b.	Q 5.	*Big Triangle was inside. Circle was inside. Big Triangle argued with Circle because Big Triangle was angry at Circle. Big Triangle exited. He closed a door. Circle moved to the corner because Circle was feeling sad.*
	Q 83.	*Circle approached Little Triangle in order to attack Little Triangle. Circle pushed on Little Triangle to attack Little Triangle. Big Triangle chased Circle because Big Triangle was angry at Circle.*
c.	Q 5.	*The big triangle argues with the circle. He is inside the box because he is asleep. She is inside it because she is asleep. He exits it and closes the door. She goes to the corner because he argues with her.*
	Q 83.	*The circle approaches the little triangle and pushes him in order to attack him. The big triangle chases her because the big triangle likes the little triangle and she attacks the little triangle.*

Fig. 1. Examples of textual narratives for TriangleCOPA questions, (a) authored by Maslan et al. [13], (b) generated by our system, and (c) generated by Ahn et al. [1]

research on automated interpretation, and a human-authored textual narrative of the depicted situation to support research on narrative text generation.

We are aware of no end-to-end system that is capable of answering TriangleCOPA questions given only the video clip as input, but Gordon [4] applied etcetera abduction to correctly answer 91 of the 100 questions using a knowledge base of 279 commonsense axioms. These 91 automatically-generated interpretations were used by Ahn et al. [1] as input story graphs for their narrative text generation method, yielding a short narrative for each correctly-answered question.

To use our narrative text generation system for this benchmark, we first generated the most probable interpretations for each TriangleCOPA question using Gordon's original knowledge base of 279 commonsense axioms. Then, we hand-authored text templates for each of the 279 axioms, which required approximately 1.5 person-workdays of effort. Finally, we generated textual narratives for each interpretation using our approach, and compared the results to the human-authored narratives in the TriangleCOPA benchmark and to those generated in the work of Ahn et al.

Example human-authored and system-authored narratives for TriangleCOPA questions 5 and 83 are shown in Figure 1.

We concede that the human-authored narratives for TriangleCOPA would be preferred by human readers for most task contexts, as they exhibit creativity in their interpretation and a fluency that is unmatched by either of the two systems. As well, we see only minor differences in the quality of text generated by either of the automated approaches.

In an effort to quantify the relative performance of each system on this benchmark, we explored the use of language model perplexity as a metric of fluency. Typically, perplexity is used in computational linguistics research to quantify the accuracy of a given language model, where lower scores indicate that the model finds the input language less perplexing. Here we use a single high-quality language model to see which system generates text that is closer in perplexity to that of the human-authored narratives. Specifically, we utilize the transformer-based GPT-2 language model [16] to compute the perplexity of a given text, as e^{loss} given the *loss* of the output tensor. Perplexity scores for narratives were computed using a PyTorch script employing pre-trained models provided in the HuggingFace transformers package.

For each TriangleCOPA question, we computed the perplexity of narratives generated by each of the two systems, and compared them to the the perplexity of the corresponding human-authored narrative. To assess whether observed differences were significant, we computed statistical p-values using stratified shuffling, a compute-intensive significance test that is popular in computational linguistics research when comparing different systems on the same test set [19]. In this context, p-values answer the question, What is the likelihood that we would see a difference in mean scores this large if there was actually no difference between the systems that generated these results?

Table 1 shows the results of this comparison of perplexity. The language model finds the narratives produced by our approach to be slightly more perplexing than human-authored texts, and those produced by Ahn et al. to be somewhat less perplexing. These differences are statistically significant only for the Ahn et al. results.

There are many pitfalls in this use of automatic evaluation metrics such as perplexity in research on natural language generation, as they are often shown to have poor correlation with human judgements of language quality [15]. Although we are encouraged that our approach to narrative text generation produces text that more closely matches the perplexity of human-authored narratives, we view these results with caution. To our eyes, the human-authored narratives in this study are superior in quality to the system-generated texts in all cases. We see several improvements that could be made to our approach, which may not

Table 1. Mean perplexity of narratives of TriangleCOPA questions

version	$p(version)$	$\lvert p(gold) - p(version)\rvert$	p-value
Maslan et al. [13] (gold)	4.686	0	n/a
Our system	4.790	0.104	0.444
Ahn et al. [1]	4.041	0.652	< 0.001

be easily assessed using only the metric of perplexity. Instead, we are most encouraged by the finding that our approach generates text that is at least as good as Ahn et al., using a much simpler method.

5 Conclusions

The use of structured graphs to represent story content has aided progress in automated interpretation and narration by allowing researchers to focus their efforts on either of the two different parts of the problem, namely graph construction and natural language generation. However, a downside of this separation is that certain opportunities to exploit synergies across these two processes are not immediately evident. The problem addressed in this paper is one such example, where the assembly of sentences from connected nodes in the graph is greatly simplified by attaching templates directly to the axioms used to make these connections during the interpretation process. Here we exploit a particular feature of interpretations constructed using etcetera abduction, namely that the etcetera literals present in a solution indicate exactly which axioms were used in its construction, along with the variable bindings for each universally quantified variable. Using text templates and straightforward methods for including proper nouns, common nouns, and pronouns, the difficult grammatical problems of sentence construction can be largely avoided. While the resulting narrative text is similar in quality to that of more sophisticated approaches, our hope is that the simplicity of our method encourages researchers to shift their development efforts toward more interesting aspects of the narrative text generation problem, such as content selection and discourse planning.

References

1. Ahn, E., Morbini, F., Gordon, A.S.: Improving fluency in narrative text generation with grammatical transformations and probabilistic parsing. In: Proceedings of the 9th International Natural Language Generation Conference. pp. 70–73. Association for Computational Linguistics, Stroudsburg, PA (Sep 2016)
2. Elson, D.K.: Modeling narrative discourse. Columbia University (2012)
3. Elson, D.K., McKeown, K.R.: A platform for symbolically encoding human narratives. In: AAAI Fall Symposium: Intelligent Narrative Technologies. pp. 29–36 (2007)
4. Gordon, A.S.: Commonsense interpretation of triangle behavior. In: Thirtieth AAAI Conference on Artificial Intelligence. pp. 3719–3725. AAAI Press, Palo Alto, CA (2016)
5. Gordon, A.S., EDU, U.: Interpretation of the heider-simmel film using incremental etcetera abduction. Advances in Cognitive Systems **6**, 1–16 (2018)
6. Gordon, A.S., Spierling, U.: Playing story creation games with logical abduction. In: International Conference on Interactive Digital Storytelling. pp. 478–482. Springer (2018)
7. Heider, F., Simmel, M.: An experimental study of apparent behavior. The American Journal of Psychology **57**(2), 243–259 (1944)

8. Hobbs, J.R.: Resolving pronoun references. Lingua **44**(4), 311–338 (1978)
9. Hobbs, J.R.: Ontological promiscuity. In: Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics. pp. 60–69. Association for Computational Linguistics (1985)
10. Hobbs, J.R., Stickel, M.E., Appelt, D.E., Martin, P.: Interpretation as abduction. Artificial Intelligence **63**(1-2), 69–142 (Oct 1993)
11. Inoue, N., Inui, K.: Ilp-based inference for cost-based abduction on first-order predicate logic. Journal of Natural Language Processing **20**(5), 629–656 (December 2013)
12. Lukin, S.M., Walker, M.A.: Narrative variations in a virtual storyteller. In: International Conference on Intelligent Virtual Agents. pp. 320–331. Springer (2015)
13. Maslan, N., Roemmele, M., Gordon, A.S.: One hundred challenge problems for logical formalizations of commonsense psychology. In: Proceedings of the Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning. pp. 107–113. AAAI Press, Palo Alto, CA (2015)
14. Meadows, B.L., Langley, P., Emery, M.J.: Seeing beyond shadows: Incremental abductive reasoning for plan understanding. In: Plan, Activity, and Intent Recognition: Papers from the AAAI 2013 Workshop. pp. 24–31. AAAI Press, Palo Alto, CA (2013)
15. Mir, R., Felbo, B., Obradovich, N., Rahwan, I.: Evaluating style transfer for text. 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). pp. 495–504. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
16. Radford, A., Wu, J., Child, R., Luan, David @and Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
17. Roemmele, M., Bejan, C., Gordon, A.: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In: Proceedings of the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning, Stanford University (2011)
18. Trabasso, T., Van Den Broek, P.: Causal thinking and the representation of narrative events. Journal of memory and language **24**(5), 612–630 (1985)
19. Yeh, A.: More accurate tests for the statistical significance of result differences. In: COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics (2000)