UNIVERSITY OF CALIFORNIA
SANTA CRUZ

GENERATING VARIATIONS IN A VIRTUAL STORYTELLER

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Doctor of Philosophy

in

COMPUTER SCIENCE

by

Stephanie M. Lukin

March 2017

The Dissertation of Stephanie M. Lukin
is approved:

Professor Marilyn A. Walker, Chair

Professor Jim Whitehead

David K. Elson, Ph.D.

Dean Tyrus Miller
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

Generating Variations in a Virtual Storyteller

by

Stephanie M. Lukin

This dissertation introduces the Expressive-Story Translator (EST) content planner and Fabula Tales sentence planner in a storytelling natural language generation framework. Both planners operate in a domain independent manner, abstractly modeling a variety of stories regardless of story vocabulary. The EST captures story semantics from a narrative representation and constructs text plans to maintain semantic content through rhetorical relations. Content planning is performed using these relations to enhance narrative effects, such as modeling emotions and temporal reordering. The EST transforms the story into semantic-syntactic structures interpreted by the parameterizable sentence planner, Fabula Tales. The semantic-syntactic integration allows Fabula Tales to employ narrative sentence planning devices to change narrator point of view, insert direct speech acts, and supplement character voice using operations for lexical selection, aggregation, and pragmatic marker insertion. The frameworks are evaluated using traditional machine translation metrics, narrative metrics, and overgenerate and rank to holistically test the effectiveness of each generated retelling. This work shows how different framings affect reader perception of stories and its characters, and uses statistical analysis of reader feedback to build story models tailored for specific narration preferences.

To Mom and Dad

Acknowledgments

I would like to thank everyone who has been a part of my Ph.D– what a journey this has been! First and foremost, thank you to my advisor Marilyn Walker, who took me under her wing and guided me through the Ph.D. path since the beginning.

To David Elson, for his time and dedication to Story Intention Graphs. I'm glad to be a part of the work he began in his own thesis. To Elena Rishes, with whom this project started out as a collaborative class project. People still remember our tag-team ICIDS presentation and the Elena-Stephanie Translator lives on. To my committee members, Jim Whitehead and Arnav Jhala, for your feedback to first help shape my work into a well-scoped cohesive body of work, and then to make the presentation of the work even stronger.

To the NLDS lab and other collaborators over the years, Elahe Rahimtoroghi, Amita Misra, Shereen Orbay, Lena Reed, Grace Lin, Rob Abbott, Reid Swanson, Kevin Bowden, James Ryan, Merley Conrado, Raquel Justo, and especially to Zhichao Hu, with whom I began graduate school and have had much fun playing BDO and DND. To fellow researchers met at conferences all over the world, for your impromptu conversation and insights that have opened my eyes to new ideas and directions, and especially to Michael Garber-Barron, for our many discussions about emotion and appraisal.

To my professors at Loyola University, Roberta Sabin, Dave Binkley, Dawn Lawrie, and Roger Eastman, who introduced me to the joys of computer science and encouraged me to pursue a doctorate. To the UCSC SURF-IT program and those involved, it was through this program that I first came to Santa Cruz.

To friends who have walked the Ph.D. path together, especially Lourdes Morales and Corinne Beier. Many cupcakes were consumed over the years. To my graduate school roommates, Julia Kelly, Leah Sicat, and Beth Mole, for pie making and entering the waters of graduate school together. To my oldest friend and sister, Michele Fujii, not only for your critical eye and editorial feedback, but for being my CBFE.

To my parents, for always loving and supporting me, especially when I embarked on a new journey in California. Skype has been our savior over the years. And finally, to Lincoln, for being my continual support. Now we can finally take that trip to France.

Chapter 1

Introduction

> *Stories forge connections with others allowing us to share similarities, affirm who we are, and cross the barriers of time and space.*
> – Andrew Stanton, Filmmaker, Toy Story, Finding Nemo, WALL-E [Stanton, 2013]

As humans, we are psychologically wired to understand the world through stories. People often structure observed events into a story [Bruner, 1991, Gerrig, 1993, McAdams et al., 2006]. An average day at work may later be described as a narrative and the events exaggerated to revolve around the individual, creating a story, rather than simply listing the events that happened to them. Our reaction to stories is not purely social; Speer et al. [2009] found that when reading stories, we mentally enact the events as if we are personally observing or performing the behavior. Specific regions of the brain activate, observed with an fMRI, when readers are following a character's physical location or tracking the character's goals and plans [Speer et al., 2009]. The active brain regions mirror activation when people perform, imagine, or observe similar everyday real-world activities. We not only enact events in our imaginations when reading a book or watching a movie, but constantly tell ourselves stories and make ourselves part of them.

This human capacity for storytelling has given rise to the development of interactive narrative systems to appeal to the immersive nature of stories and the strong emotional and psychological effect they have on readers [Bohanek et al., 2006]. In education and training simulations, researchers provide students with the opportunity to embark on an interactive learning adventure, instead of using standard textbook and lecturer tools [Aylett et al., 2005, Johnson et al., 2004, Ward et al., 2012]. Narrative Centered Environments, for instance, are immersive fantasy worlds through which the student can learn by co-constructing an unfolding narrative [Mott and Lester, 2006, Mott et al., 2006]. The student, now an active player, explores an open world. They are encouraged to form and test hypotheses inspired by their exploration and interaction with the world, and talk with non-player characters to discover information about a variety of topics.

In addition to interactive learning, stories can be used to share experiences through dialogue interactions such as intelligent virtual agents. Vassos et al. [2016] create a virtual museum art-bot that weaves a story of the life and works of the artist as the visitor explores the art exhibit. Visitors follow the story on their smartphones, asking questions of the agent to heighten the visitor experience.

Stories are rarely told once, and rarely told the same way twice. Ideally, an interactive storytelling system would emulate this variance in framing, mimicking how storytellers tailor their stories: repeatedly telling a story to get the desired effect and communicate effectively with the audience. A storyteller may explore different interpretations of the same incident from multiple points of view [Mateas, 2001], or use a richer style to address highly interactive and responsive addressees [Thorne, 1987]. In life-threatening narratives in young adults, different telling styles are used to convey different messages to the audience, such as empathy for others, preoccupation with one's own fear or sadness, or one's courage or bravery [Thorne and McLean, 2003]. Moreover, narratives are highly effective for persuasion [Green and Brock, 2000] and the degree of immersion in a story increases this effect [Green, 2004].

A perfect display of different framing is in Queneau's book Exercises in Style where a basic sequence of events (a man gets into an argument with another passenger on a bus and runs into him later) are told in 99 different ways [Queneau and Wright, 1981]. Framings explored include retrograde ("I met him in the middle of the Cour de Rome, after having left him rushing avidly towards a seat"), surprise ("Two hours after, guess, whom I met in front of the gare Saint-Lazara! The same fancy-pants!"), and hesitation ("I rather think that it was the same character I met, but where? In front of a church? in front of a charnel-house? in front of a dust-bin?"). Madden [2006] repeats the exercise in visual storytelling (a man being asked the time by his roommate then forgetting why he was looking in the refrigerator) and creates different visual depictions of the story including manga, humor comic, and political cartoon.

These events (running into someone, being asked the time, looking in the refrigerator) abstract each story into a set of building blocks from which more complex narrative forms can be built [Abbott, 2008]. These building blocks are the *fabula*, denoted in Russian formalism [Propp, 1969], or narrative discourse, pieces of the story world including the characters, their goals, and the actions that take place in the story world. The *sujet* is the telling and framing of a subset of these events [Propp, 1969], including the manipulation of the presentation of events, the emotional impact of which the story is narrated, a subjective or objective interpretation, and the perspective from which the story is told.

In their works, Queneau and Madden create 99 different framings, *sujet*, from a singular *fabula*. The storyteller has many devices at their disposal to frame stories, including changing the overall tone, mood and effect of the story to distinguish between “who sees?” and “who speaks?” [Genette and Lewin, 1983]. Theories of narratology provide a number of such narrative devices or parameters to produce diverse framing [Bal, 1997, Genette and Lewin, 1983, Lönneker, 2005, Prince, 1974]. From Lönneker [2005]:

- *Mood:point of view*: Spatial, temporal, and ideological point of view from which events are described. Events can be described from the third person point of view of the storyteller or a character, or from the first person perspective of characters in the action.

- *Mood:focalization*: Accessibility of knowledge needed to select story events for presentation in discourse. If a narrative instance disposes of unrestricted knowledge of the story world, it uses external focalization; if the knowledge is restricted to a character’s field of perception, focalization is internal.

- *Voice:person*: Narrator participation. A narrative instance is a character of the current narration (grammatical realization typically in the first person), while a heterodiegetic narrative instances is “absent” from the current narrative and not referred to.

Consider the personal narrative *Bug Out For Blood* in Table 1.1 from Burton et al. [2009]’s Spinn3r corpus. This framing is told in the first person (*Mood:point of view*) in the narrator’s own voice (*Voice:person*). The narrator tells about a time when she saw bugs in her house and exaggerates her fear as she hunts the bugs down. A different framing of this story could instead

> Bug out for blood the other night, I left the patio door open just long enough to let in a dozen bugs of various size. I didn't notice them until the middle of the night, when I saw them clinging to the ceiling. Since I'm such a bugaphobe, I grabbed the closest object within reach, and with a rolled-up comic book I smote mine enemies and smeared their greasy bug guts. All except for the biggest one. I don't know what it was; it was one of those things you see skimming the surfaces of lakes, with a legspan of a few inches. I only clipped that one, taking off one of its limbs. But it got away before I could finish the job. So now there's a five-limbed insect lurking in the apartment, no doubt looking for some vengeance against me. I'm looking around corners, checking the toilet before sitting down, checking the bowl before taking another scoop of cereal, wondering when it's going to jump out.

Table 1.1: *Bug Out For Blood* personal narrative

make moral judgments against the agents of her distress, the bugs, and discuss the ramifications, trying to evoke fear or distress in the audience. For example, the narrator may have a serious allergic reaction to this particular kind of bug and elaborate on what would happen if she is bitten. Yet another framing could instead evoke the audience empathy for the bugs by retelling *Bug Out For Blood* from the perspective of the bugs, who are cold and hungry and only seeking a warm place to live (*Mood:point of view, Mood:focalization*), and communicate despair as they are hunted down (*Voice:person*).

A computational treatment of storytelling gives the advantage of retelling stories, such as *Bug Out For Blood*, in these different styles, and harnesses them in interactive narrative environments. This story in particular could be incorporated into an experience to help students overcome a fear of bugs, or, through a different narrative, a fear of heights, through a dramatic telling or a dialogue between friends. To immerse the subject into the narrative experience, the system must learn to tailor each narrative to the experiencer.

A general and computational model of the *fabula* building blocks would allow for the procedural manipulation of the presentation, the *sujet*, to create different framings from one

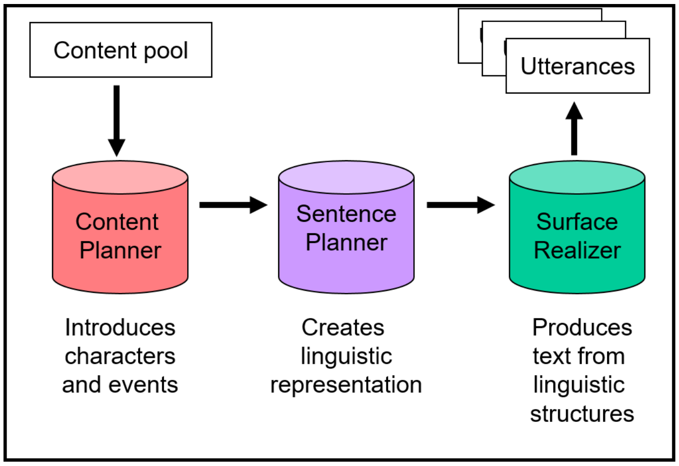

Content pool
Utterances
Content Planner
Sentence Planner
Surface Realizer
Introduces characters and events
Creates linguistic representation
Produces text from linguistic structures

Figure 1.1: Natural Language Generation Pipeline

fabula. Natural Language Generation (NLG) engines offer part of the solution to producing many *sujet* from a computational *fabula*. Figure 1.1 shows the traditional natural language generation pipeline introduced by Reiter et al. [2000] that is most commonly used today.

The content pool is the *fabula*, a representation of the characters (e.g. the narrator, the bugs), setting (the narrator's house), props (a rolled up comic book), and the events and the semantic relationships between them (e.g. a database or a temporal graph encoded with the fact that the narrator is hunting the bugs).

The rest of the NLG pipeline is modular and parameterizable, and pulls from the *fabula* to create *sujet*. The content planning module is responsible for selecting content from the content pool to include in the telling such as temporal ordering or inferring semantic relationships to manipulate. The sentence planning module creates linguistic representations for the selected events from the *fabula* and is responsible for stylistic variation such as elaboration and altering the narrative style. Finally, the surface realizer module lexicalizes the linguistic representation into a textual utterance.

Fabula	The narrator hunted the bugs.
Sujet	I ran down those bugs!
	Something big was chasing us!
	"Get back here!" she screamed as she chased the bugs.

Table 1.2: Possible *sujet* for one event in the *fabula*

Table 1.2 shows possible realizations of the same story event from the *fabula*. The first *sujet* is a straightforward interpretation of the story event from the narrator's perspective using lexical substitutions, invoking the sentence planning module of the NLG pipeline. The second *sujet* is a focalization of the story event from the bug's perspective, again utilizing sentence planning. The final *sujet* transforms the action into a speech act, invoking both content planning and sentence planning.

Utilizing an NLG engine for storytelling helps authors in two ways. First, it allows for reusability of stories and story paths, supplementing diverse story content. It is critical for a generation engine to reuse story content and not be limited in terms of the story domain. A collection of stories with insights into everyday life evoking emotion from different audiences, such as *Bug Out For Blood*, would be useful for learning how to tell stories in many different ways by adapting the story style or framing to particular audiences or to achieve specific goals.

Second, an NLG engine could help alleviate the authorial burden hindering authors from creating compelling characters in stories and games. Manually authoring dialogue in large games can be tedious and time-consuming, or even altogether infeasible. Researchers have attempted to overcome this burden by filling in templates or scripts from a planner for different characters [Cavazza and Charles, 2005, Mateas and Stern, 2002, McCoy et al., 2011], or modeling archetypes [Rowe et al., 2008]. Parameterizable NLG engines have also been used to generate possible character utterances which authors can then modify [Walker et al., 2013].

These advantages lead to learning an automatically created generation dictionary, ideally producing variety of style with easy-to-author content. However, it is not as easy to simply combine existing NLG systems into the aspirational multi-domain, adaptive storytelling system. There exists a gap in the NLG pipeline where the content pool or representation is not rich enough, or the sentence planning module is not flexible enough in a fully streamlined system. This disconnect is defined as the NLG story gap [Callaway and Lester, 2002, Lönneker, 2005]; an architectural disconnect between narrative representation, generation, and natural language generation.

Li et al. [2013]'s work creates new stories through content planning using a logical content representation. They construct possible *sujet* from a "bank robbery" *fabula* by planning sequences of events, for example: *John enters the bank. John hands Sally a note. The note demands money* or the sequence *John enters the bank. John pulls out a weapon. Sally screams*. However, there are two shortcomings in this approach with respect to the NLG gap. First, the generated language is simple and cannot be framed differently due to the inflexibility of the content representation. No syntactic information is available, only the fact that one event follows another. Second, it takes time to construct this representation for the bank robbery domain, and is not easily extendible to other story topics.

On the other hand, Callaway and Lester [2002]'s NLG system creates diverse tellings from a rich syntactic representation that allows for fluid and natural scenes from Little Red Riding Hood such as *The sun was shining brightly, but it was not too warm under the shade of the old trees. Little Red Riding Hood went on her way singing and gathering great bunches of wild* *flowers.* Despite the stylistic diversity, this system too suffers from lack of easily extending the domain, and can only support the Red Riding Hood story.

Bridging the NLG Gap and Content Planning

The first aim of this thesis is to bridge the NLG story gap by developing a framework that can generate diverse tellings and is reusable under different story domains. We treat a domain independent representation of content as a computational *fabula*, and create the Expressive-Story Translator (EST) translation and content planning framework to bridge the NLG gap between the narrative representation and a language representation.

We utilize the Story Intention Graph (SIG) narrative representation as *fabula* [Elson, 2012a]. The SIG representation is generalizable and does not require specific domain knowledge, modeling different topics such as *Bug Out For Blood* and many others introduced in the Spinn3r corpus. Figure 1.2 shows a subset of the SIG narrative representation corresponding to *Bug Out For Blood*. The SIG makes a distinction between the original telling of the story and the events in the story which can be extrapolated as *sujet* and *fabula* respectively. It also gives us access to the goals and emotions of the characters in the story in the interpretative and affectual layers. The Scheherazade annotation tool allows human annotators to create new SIG story encodings [Elson and McKeown, 2009].

The EST transforms SIG content into semantic-syntactic structures that are interpreted by a parameterizable sentence planner. The selected content is mapped onto text plans, which preserve semantic relationships derived from SIGs, and onto Deep Syntactic Structures (DSYNTS), [redacted] which allow for the manipulation of content at the syntactic level [Mel'čuk, 1988]. The EST-

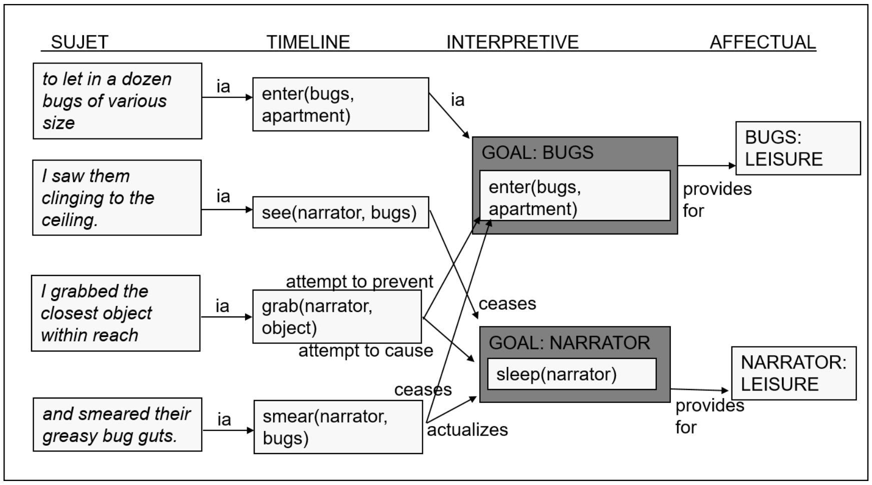


Figure 1.2: Part of the SIG for *Bug Out For Blood*

generated DSYNTS and text plans are passed to a surface realizer to generate a telling.

The EST acts not only as a one-to-many syntactic map. It is a new model that maintains semantic story and sentence information, in addition to syntax, to support content planning. The EST preserves rhetorical relations for temporal order, contrast, and causality, and can be used to manipulate the sequence in which events are told, in comparison with the sequence in which they “actually happened” [Biber, 1991].

Emotionally charged stories are potentially more memorable and impactful, and hypothesize that readers change their perceptions of characters based on how they speak. We envision a future storytelling system where an agent tells a story with animated faces, voice, and language and style to match the context of the story. Emotion derivation from a story is a part of content planning and selection, and the EST can derive from the SIG an emotional state of the characters based on the success or failure of their goals. We explore approaches for adapting the textual retelling to suit the appraisal.

Sentence Planning

The second objective of this thesis is to create a narrative parameterizable sentence planner called Fabula Tales to integrate the content planning and semantic-syntactic structures produced from the EST. Fabula Tales also introduces narrative parameters for sentence planning and tests them as to their ability to evoke difficult responses in readers: changing story narrator point of view (first or third), inserting direct speech acts, and supplement character voice using operations for lexical selection, aggregation, and pragmatic marker insertions.

Biber [1991] claims that first person pronouns are markers of ego-involvement with a text. Often the subject of cognitive verbs, first person pronouns indicate that the matter at hand is personal and an immediate mental interaction, compared to third person pronouns (see Table 1.3). By using different perspectives the narrative space is restricted to the eye of a particular character, allowing the audience limited perception, or focalization [Pizarro et al., 2003]. We hypothesize that stories told in this first person narrative space will be more engaging for the audience by activating mental involvement with the character, rather than the third because first person brings the reader into the immediacy of a story. Therefore, H_1: there is a significant correlation between first person point of view and engagement.

When storytellers tell stories, they know what their characters are feeling, and can express it in the telling. In order to reveal the depth of the characters and relate to them, we hypothesize that insight into character personality and emotions is enhanced via direct speech acts. Bal claims that "dialogue is a form in which the actors themselves, and not the primary narrator, utter language" [Bal, 1997] and that a narrative is more dramatic the more dialogue it contains. Because direct speech allows for characters to express themselves, we hypothesize

	Personal	Impersonal
Pronouns	I, you	he/she
Grammatical person	first and second person	third person
Tense	not all past tense	all past tense
Emotive	Oh!	(absent)
Conative words	please	(absent)
Modal, uncertainty	perhaps	(absent)

Table 1.3: Summary of personal and impersonal parameters from Biber [1991]

H_2: there is a significant correlation between direct speech and engagement.

The support of direct speech and the first person perspective allow the characters to speak in their own tone or style. Biber gives support for emotive words (*Oh!*), cognitive (*please*), modal and uncertainty (*perhaps*) as being indicators of personal stories, whereas these items are lacking in impersonal stories [Biber, 1991] (see Table 1.3). These voice and style features can be used in direct speech to act as character voice or narrator voice, and we hypothesize, H_3: there is a significant correlation between direct speech and certain styles and increased reader engagement.

Many other stylistic sentence planning manipulations increase the naturalness and fluidity of a story retelling. Aggregation operations help to avoid repetition and produce more coherent, concise and context aware output [Cahill et al., 2001, Paris and Scott, 1994, Scott and de Souza, 1990]. We introduce "deaggregation", the breaking apart of longer sentences into shorter sentences, which can be reconstructed to achieve different effects. This sentence planning operation is not content dependent, so all topics can take advantage of this. By making more natural sounding and concise stories, we hypothesize H_4: there is a significant correlation between domain independent deaggregation and engagement.

Evaluation

The final objective of this thesis is to define an evaluation for the new framework. Our goals are to 1) examine how the EST and Fabula Tales perform as an NLG system in terms of topic coverage and correctness, and 2) use our one-to-many-*sujet* framework to explore reader responses to different framings and test our hypotheses. It is difficult to measure the quality of utterances generated by a parameterizable NLG system because of the potential number of different stories that could be generated from these parameters. Evaluation metrics and perceptual effects of different combinations of these parameters are not yet well understood. Furthermore, personal preferences must also be taken into consideration before we can claim one variation is better than another.

First, we ensure the NLG system creates correct tellings, treating the EST and Fabula Tales generated output truly as a "translation" problem using traditional machine translation metrics to verify the essences of the story has been captured faithfully by the translation, as well as correct syntax. Then, we create powerful narratives by combining different parameters, evoking diverse framing. To explore reader perceptions, we perform voice substitution in dialogue to alter the perception of characters, showing how different parameters lead to different perceptions of the story. We conduct experiments to discover how different parameters change reader perceptions and which parameters readers prefer, focusing our evaluation metrics on narrative immediacy, correctness, preference, and naturalness.

Finally, we create and conduct a "Create Your Own Story" experimental paradigm for testing stories generated by combinations of narrative parameters. Collections of sentences generated by diverse EST and Fabula Tales parameters are aggregated, and subjects select the sentences they believe “fit” together to create a fluid story. This novel experimental setup tests many combinations at once and teases apart the parameters consistently preferred. We incorporate the results of this experiment in an overgenerate and rank methodology, enabling us to test many possible generations and see how they affect human perceptions by first generating many sentences from the EST and Fabula Tales, and then ranking them, providing training data for the rank function over preference using statistical learning to produce stories with the best parameters.

Overview of Contributions

The NLG gap prevents narrative systems from generating rich and diverse variations over a variety of topics due to an architectural disconnect in the representations of the traditional NLG pipeline. We create the Expressive Story Translator and content planner, and Fabula Tales sentence planner to support diverse story content and a variety of generations previously unavailable using SIG semantics or DSYNTS alone. The EST takes advantage of the affordances of both representations by preserving the SIG semantics through text plans and creating syntactic structures through DSYNTS. By integrating DSYNTS with story semantics, the EST constructs text plans to maintain semantic content such as rhetorical relations for temporal order, contrast, and causality. These semantics are utilized for content planning as we explore emotion modeling, temporal manipulation, and non-narrative parameters such as elaboration and repetition.

This semantic and syntactic integration allows for Fabula Tales to employ narrative sentence planning devices to change point of view, insert direct speech acts, and supplement character voice using operations for lexical selection, aggregation, and pragmatic marker insertions. We create the PersonaBank corpus of over 100 blogs encoded as SIGs, such as *Bug Out For Blood*, to showcase the domain independence of the EST methodology and Fabula Tales narrative parameters. The EST automatically constructs a generation dictionary per story; similar semantics from different stories could be merged into one entry, and associated syntactics learned and reused.

We create an evaluation methodology of narrative retellings of an NLG system with an overgenerate and rank "Construct Your Own Story" paradigm. This paradigm combines subject responses with statistical analysis while subjects take into consideration the entire generated story. We observe that that reader perceptions of characters and the overall correctness and engagement with a story changes as narratological parameters are varied. Furthermore, we find there are general reader preference trends for narratological parameters (H_2, H_3, H_4 significant), but that individual reader preferences exist (H_1 not significant).

The outline of this thesis is as follows. Chapter 2 discusses related work on the NLG pipeline and gap, and language and plot generation using different representations. Chapter 3 introduces in detail the SIG, DSYNTS, and text plans used in this pipeline. Then we discuss the first goal of the thesis; building a bridge between SIG and DSYNTS. We describe the EST translation and content planning framework, including the theoretical framework we assume for narrative representation, the translation methodology, and a walkthrough of a story through the translator. We enumerate over the potential content planning operations available to our framework and focus the discussion on emotion modeling and temporal manipulation.

Chapter 4 discusses the second goal of this thesis: customization and personalization of stories through the implementation of narratological sentence planning parameters in Fabula

Tales. We describe Scheherazade, the annotation tool used to create SIGs [Elson and McKeown, 2009], and the PersonaBank corpus of SIG personal narratives. In Chapter 5, we discuss how the EST and Fabula Tales were first developed and continually evaluated, including quantitative and qualitative evaluations. Then, the evaluation via an overgenerate and rank paradigm and examine H_1-H_4. We conclude in Chapter 6 and discuss limitations, future work, and applications.

Chapter 2

Previous Work

A primary goal of this thesis is to bridge the NLG story gap: an architectural disconnect between narrative generation and language generation. We offer a more general definition of the gap as any NLG system lacking a flexible semantic-to-syntactic mapping. We begin with a discussion of different approaches to content planning (Section 2.1.1) and sentence planning (Section 2.1.2) in NLG systems. A distinct difference between the various systems is that the approach and its capabilities differ based on the data representation of the semantics and syntactics.

We then survey several systems that advance narrative NLG systems for producing story variation using complex content planning or sentence planning (Section 2.2). However, each of these advancements are paired with a shortcoming. Systems that apply rich sentence planning variations are limited in domain coverage. Data-driven systems can overcome limited coverage, however, they tend to be inflexible in adapting the story text because, for instance, the text is retrieved from text-only based corpora. Data-driven approaches also often lack an interpretable semantic understanding between story points.

Narrative systems that prioritize content and story planning can temporally manipulate the presentation of the story, but are limited in sentence planning variations. It is often time consuming for authors to create new content for these systems. We compare these approaches against our own, and aim to address each of these shortcomings in the EST's content planning and Fabula Tales' sentence planning.

2.1 Natural Language Generation

2.1.1 Content Planning: Planning and Structuring

Content planning, or document planning, is the module in the NLG pipeline responsible for the selection and structure of content to be told in a story, utterance, or summary. The planner selects events and observations from the larger *fabula*, and narrows down what to include in the telling.

The work of Reiter et al. [2005] generates weather forecasts from numerical weather predication data. A data segmentation algorithm is tailored to domain-specific goals. Data points of interest for weather forecasting are recorded and prioritized, including wind speed, temperature, and cloud coverage. It is important not to overwhelm the produced summary with tiny, non-important details, and of course important details must not be excluded. One summary may mention that the wind direction changes, but not that the wind speed has changed.

The content planning algorithm of Reiter et al. [2005] is extended to new domains in the works of Sripada et al. [2003] and Yu et al. [2007], generating summaries for neonatal intensive care unit and gas turbines respectively. The planning algorithm is tailored to heart rate and blood pressure for neonatal summaries, and spikes, dips, and oscillations for gas turbine summaries.

The representation of input depends upon the application, and determines the kind of methodology employed for content selection. These data segmentation content planning approaches are useful in strictly observationally based domains where there are gold standard goals for selecting and presenting content. But in an application such as dialogue, where there are less expectations about what to say, but rather, how that content is structured. The PERSONAGE NLG engine offers a different approach to content planning [Mairesse and Walker, 2008]. The underlying syntactic-semantic representation for dictionary entries in PERSONAGE is called Deep Syntactic Structures or DSYNTS [Mel'čuk, 1988]; these are based on Meaning-Text theory and allow for sentences to be altered and combined to produce new sentences with similar meanings.

The content planning module of PERSONAGE is responsible for content size, polarity, and structure of arguments. Content size makes use of the verbosity, repetition, and restatement parameters, and content polarity compares a statement that focuses on the bad information, e.g. *X has mediocre food*, with a neutral statement of fact: *X is a Thai restaurant*. The order in which content is presented can affect the persuasiveness of an argument [Carenini and Moore, 2000]. PERSONAGE manipulates how negative and positive information is presented and if the final claim is more effective in a disagreement or agreement. Much of content ordering is used in aggregation with respect to the sentence planning, e.g. contrast, justify.

These content planning operations can be parameterized and combined into a voice model. Mairesse and Walker [2011] build models from the parameters to emulate the Big Five personality traits [Gosling et al., 2003]. They ensure the models are perceived as expected, showing that humans perceive the personality stylistic models in the way that PERSONAGE intended.

Using statistical analysis of character dialogue in movie scripts, Lin and Walker [2011] automatically learn parameters associated with characters, such as Indiana Jones and characters from the TV show Friends. Rather than use the Big Five personality traits as the basis of a model, PERSONAGE parameters are combined to produce good coverage and portray the intended style of the target character. These models are not limited to only content planning, as we discuss in the next section.

2.1.2 Sentence Planning: Stylistic Variation

NLG research on pragmatic and stylistic choices forms one of the building blocks of our approach. Many recent state of the art NLG systems offer stylistic enhancements to the sentence planning module of the NLG pipeline that are not exclusively in the storytelling domain or harnessed into a larger interactive experience. The open question in these works is, *given some representation of content, how do we generate very different versions?* Our coverage here focuses on the individual contributions to the NLG pipeline from the syntactic standpoint, and that the systems assume some fixed semantics as input. Many of these theories and approaches for restaurant recommendations, news story generation or direction giving could be applied to the storytelling domain.

The first system to explicitly explore narrative variations and their pragmatic effects was the PAULINE system [Hovy, 1987]. PAULINE's input is a set of pragmatic (communicative) goals, such as the opinion or style to be conveyed. To reduce the complexity of the rules involved, linguistic features were associated with a small set of intermediary *rhetorical goals*, which were combined to generate various pragmatic effects. For example, the rhetorical goals of low formality, high force and high partiality produce a "no-nonsense" effect. Other pragmatic effects include the topic's subjective connotation, the confusion induced in the hearer, or the distance between the hearer and the speaker.

Other work focuses on variation in journalistic writing or instruction manuals, where stylistic variations as well as journalistic slant or connotations have been explored. More recent work controls the subjectivity of its output by selecting the utterance minimizing a distance measure between the connotation of individual phrases and a scalar vector representing the attitude of the user towards the phrase's object [Fleischman and Hovy, 2002b]. For example, the system would choose the template utterance *'Y smashed into X'* over *'X was hit by Y's car'* if the user is known to dislike Y. An evaluation shows that human perceptions of the attitude of the speaker towards the utterance's object correlate significantly with the system's emotional target. These slant arguments could be utilized in a storytelling system to shift blame on different characters according to some focalizer. This approach would be especially useful to incorporate into an interactive narrative system when teaching young children how to respond to bullying, such as Aylett et al. [2005], by immersing them into a narrative environment where the child can experience the perspective of other individuals involved in the scenario.

The idea that lexical choice alone has a large influence on the construal of narrated events is also found in the works of DiMarco and Hirst [1993] and Inkpen and Hirst [2004]. This line of research uses stylistic grammars that associate linguistic markers with primitive stylistic elements, which in turn are mapped to stylistic goals over three dimensions: clarity/obscurity, concreteness/abstraction, and dynamism/staticness. For example, a “heteropoise” sentence, i.e. using multiple grammatical forms, results in a low clarity but a high concreteness [DiMarco and Hirst, 1993]. Stylistic grammars can constrain the NLG engine to produce a list of primitive stylistic elements based on the target stylistic goals, e.g. use of conjunctive clause for concreteness, which are consulted whenever decision are made about a stylistic element, e.g. clause aggregation [Green and DiMarco, 1996].

This mode of stylistic control is more grounded in linguistics than PAULINE, but it requires many handcrafted rules, and it has not been fully evaluated. Our system performs similar manipulations to a syntactic structure, but we have primarily focused on high level narrative parameters. However, it is not inconceivable to imagine how these variations could be integrated into the already parameterizable DSYNTS representation.

Power et al. [2003] built the ICONOCLAST system to generate patient information leaflets in different styles appropriate to the genre and style of author such as, paragraph length, frequency of passive voice, commas, and technical detail. One style may take a narrative tone, describing in detail how one takes a tablet, e.g. *To take a tablet, remove the tablet from the foil, and swallow it with water.* Another style is a bulleted list: *To take a tablet: 1. Remove it from the foil. 2. Swallow it with water.* The input structure is a graph with constraints, which is difficult to port to other domains, but this approach utilizes author level goals to create different framings, as many narrative systems currently do.

Previous research on NLG of linguistic style shows that dialogue systems are more effective if they can generate stylistic linguistic variations based on the user's emotional state, personality, style, confidence, or other factors [André et al., 2000, Dethlefs et al., 2014, Forbes-Riley and Litman, 2011, Mcquiggan et al., 2008, Piwek, 2003, Porayska-Pomsta and Mellish, 2004, Wang et al., 2008]. Such variations are important for expressive purposes, for user adaptation and personalization [Mcquiggan et al., 2008, Wang et al., 2008, Zukerman and Litman, 2001]. Furthermore, alignment or entrainment in text can be learned by examining and adapting the style of the system response to the user's style. For example, in a direction giving task, if the user asks the question "Do I hang a right at Pacific street?", the system can respond in turn "Yes, hang a right at Pacific street" instead of a non-aligned response "Yes, turn right at Pacific street". Alignment is shown to increasing dialogue task completion and cooperation [Hu et al., 2015, 2014, Kühne et al., 2013]. We posit that alignment or stylistic preference such as our H_{1-4} will be of similar influence on increasing engagement.

In addition to content planning, PERSONAGE implements parameterizable sentence planning decisions including aggregation and lexical choice. PERSONAGE also supports syntactic template selection, including syntactic complexity, simple or complex sentences. Pragmatic transformations insert various markers into the generation to produce certain effects. Inserting certain softener syntactic elements such as sort of, kind of, somewhat, mitigates the strength of a proposition. On the other hand, emphasizer hedges insert syntactic elements to strengthen a proposition, e.g. really, basically, actually. Stuttering duplicates part of a content word, e.g. *I ju-jumped because the bugs startled me*. Filled pauses insert syntactic elements expressing hesitancy, e.g. I mean, err, like, mmhmm, you know. PERSONAGE associates the softener hedges, stuttering, and filled pauses with the introverted voice, and emphasizer hedges, exclamation,

	Sparrow	Otter
General Traits	gregarious, social, impulsive, flighty	playful, child-like, eager, curious
NLG params	repetition, exclamations, short sentences	expletives, in-group address terms, tag questions, disfluencies
NLG sample	Oh I mean, you must thwart Cartmill. You need to stop Cartmill. No one is worse than Cartmill.	Well, mmhm... no one is worse than Cartmill, so Cartmill cannot be permitted to continue.

Table 2.1: Character dialogue in SpyFeet

and expletives with the extroverted voice.

Reed et al. [2011] introduce SpyFeet, a mobile game to encourage physical activity, that makes use of dynamic storytelling and interaction using a descendant of PERSONAGE, called SpyGen, as its NLG engine. Figure 2.1 shows general speech traits of two unique characters in the game world, a Sparrow and an Otter, the associated PERSONAGE content planning and sentence planning parameters, and generated utterances. The Sparrow makes use of the repetition and short sentences content planning, and exclamations in sentence planning. The Otter mostly uses sentence planning operations for tag questions, disfluencies and other pragmatic markers.

Another source of linguistic variation comes from splitting and aggregating sentences at the discourse level. Aggregation operations help to avoid repetition and produce more coherent, concise and context aware output [Cahill et al., 2001, Paris and Scott, 1994, Scott and de Souza, 1990]. Nakatsu and White [2010] use tree adjoining grammar (TAG) proposed by Joshi and Schabes [1991] for sentence planning because TAG can generate clauses with multiple connectives and dependencies not constrained to forming a tree. Mann and Thompson [1988] introduce Rhetorical Structure Theory, from which many works use for aggregation and sentence planning. Walker et al. [2007]'s SPaRKy from the Restaurant Corpus trains a sentence planner from RST trees and Howcroft et al. [2013] create enhancements to the trees. These systems use PERSONAGE as the underlying sentence planning framework.

We implement aggregation in our work, building on these previous theories of RST. However, we are not aware of previous NLG applications needing to first de-aggregate the content, before being able to apply aggregation operations. The semantics provided by the SIG representation are advantageous in the storytelling domain. Furthermore, the flexibility of the DSYNTS representation and parameterization that PERSONAGE and others make use of is appealing from a syntactic standpoint, and why we implement DSYNTS in our work.

2.1.3 Overgenerate and Rank

Another way to generate many variations is with a model that overgenerates first and then ranks the produced output based on some measure of goodness. Previous work has examined the use of a statistical or n-gram model as an objective scoring function. Langkilde and Knight [1998] (Nitrogen) and Langkilde-Geary [2002] (Halogen) measured the quality of all the sentences their systems generated using an overgenerate and rank method to generate candidate utterances based on some set of statistical rules, and then rank the utterances based on the highest probability according to an n-gram model. An advantage of this method is that unacceptable, unexpected, or incomplete utterances are generated and then ranked low so that their rank is not promoted.

Isard et al. [2006]'s CrAg-2 system generates dialogue between two agents trained to be both aligned and individualized by overgenerating utterances according to n-gram, personality, and alignment models. The CrAg-2 system has not been tested. Inkpen and Hirst [2004] use the overgenerate and rank method for learning near synonyms, words that have the same meaning but differ in lexical nuances. The system, Xenon, generates sentences with Langkilde and Knight [1998]'s Halogen and inserts near-synonyms into the sentence and evaluates if the sentence has correctly selected and inserted the synonym.

The overgenerate and rank method can also be used without rules or probabilities as a scoring method, but instead elicit human judge feedback. In Walker et al. [2002], a sentence planner is automatically trained, using feedback from human judges, to choose the best from among different options for realizing a set of communicative goals. Using the judge feedback, a learning component is created that selects sentence plans with a maximum of a 5% lower rating than human ranked sentence plans. Walker et al. [2007] believe the simplest way to deal with the inherent variability in possible generation outputs is to treat generation as a ranking problem. Possible realizations are generated according to n-gram, concept, and tree features. User feedback is collected to rank the utterances. Based on user preferences, they learn which features made which utterances well structured and natural, and attempt to duplicate the rankings in the training examples.

PERSONAGE was tested with an overgenerate and rank evaluation. After generating several utterances with different parameters, they ask users to select the "best" utterances, in their case, the most natural introverted or extroverted utterances [Mairesse and Walker, 2010]. After receiving user feedback, user preferences as well as their own personality are incorporated into a statistical learning to refine PERSONAGE models. Walker et al. [2013] generate multiple utterances using PERSONAGE and present them to users to select the most natural and fitting for their experimental context. The benefits to an overgenerate and rank approach are that human evaluations are used to build statistical models for learning features of an NLG system. We apply this approach in our own evaluations.

2.2 Narrative and Dialogue Systems

2.2.1 Narrative Prose Generation

In the fictional domain, Storybook is an end-to-end narrative prose generation system that utilizes a primitive content planner along with a complex sentence planner and surface realizer to produce multi-page stories in the Little Red Riding Hood fairy tale domain [Callaway and Lester, 2002]. The input to Storybook is a narrative stream that reflects the orderly progression of events, descriptions and states, produced by a hypothetical story generator. Storybook expects the input stream to contain narrative primitives that specify scene changes and other aspects of the narrative structure such as instructions about narrative person, focalization, and details about the dialogue realization. Storybook's main contribution is in the sentence planner that converts the narrative stream into different possible narrative plans to be realized by an off-the-shelf generator FUF-SURGE [Elhadad and Robin, 1996]. Storybook manipulates sentence planning parameters such as lexical choice and syntactic structure, as well as narratological parameters such as person, focalization, and the choice to realize dialogue as direct or indirect speech, similar to the sentence planning variations offered by Fabula Tales. For example:

> Her grandmother was a very old lady, who lived in the heart of a neighboring wood.
> She was delighted at being sent on this errand, for she liked to do kind things.
> ”Where are you going, Little Red Riding Hood,” said the woodman, ”all alone?”

The flow of the Storybook framework is similar to that of our approach. Where Storybook maps from a narrative stream to FUF-SURGE, we map the SIG representation to syntactic DSYNTS and use off-the-shelf tools to perform surface realization. However a limitation of Storybook is that its commonsense rules and the module that translates the story into the discourse are specific to the domain of Little Red Riding Hood, whereas our approach allows for easier annotation of input stories using the markup tool Scheherazade [Elson and McKeown, 2009] for creating SIGs. Furthermore, the EST is not simply a syntactic map, but a content planner.

Another significant contribution to narrative variation is Montfort [2007]'s planner for generating multiple variations of text in an interactive fiction (IF) environment. Inspired by theories of narratology, Montfort identifies and isolates several aspects of narrative variation, chiefly the selection and ordering of the events in the underlying story representation, and the style of the resulting prose. Montfort develops an authoring tool that allows IF authors to write enriched strings as templates that specify which parts of the discourse can be varied. The IF system, and its successor Curveship, can then recover the story from the world simulator for the interactive fiction domain and render narrative variations, such as different focalizations or temporal orders. One such example is the "speed" of the narrative, which hinges on the order in which the story is told. For example, the following excerpt explicitly uses ellipses in identifying an area of action not being explicitly narrated: *No mention shall be made of what happened in the southern part of the plaza.*

Curveship allows IF authors to define a minimal specification of the story characters, objects, and possible events, and the system can produce and alter the narration in each playthrough, focusing heavily on temporal manipulation. In our work, we explore how the

EST acts as a content planner and posit that it can achieve parameterize temporal manipulations similar to that of “speed” using semantic information from the SIG preserved by the semantic-syntactic structures produced by the EST. A syntactic constraint on Curveship is that rather than relying on a full NLG engine and off-the-shelf linguistic resources and ontologies, Curveship uses resources crafted for a particular narrative application, so the rich generation is only applicable to the IF interface.

2.2.2 Narrative Plot Generation

Tale-Spin [Meehan, 1977] is a classic story authoring tool that operates at the content planning level, allowing for characters’ goals to craft and motivate the telling of the story. Prior to story generation, Tale-Spin requires that authors assert facts about the world and specifies character knowledge about themselves and their relationships with other characters. A story is then generated from the autonomous actions of the characters’ wants and needs, and is influenced by character perceptions about the world. Tale-Spin’s complex planning gives it fine control over the story world, but produces a simple generation style:

> Once upon a time George Ant lived near a patch of ground. There was a nest in an ash tree. Wilma bird lived in the nest. There was some water in a river. Wilma knew that the water was in the river. George knew that the water was in the river. One day Wilma was very thirsty. Wilma wanted to get near some water. Wilma flew from her nest across a meadow through a valley to the river. Wilma drank the water. Wilma wasn’t very thirsty anymore.

This generated story focuses primarily on Wilma’s goals and actions, but still mentions George as a supporting character. The implementation of a focalization or point of view parameter could remove George from the action, truly telling the story about Wilma.

Previous work on providing content for narrative generation for fictional domains has typically combined story and discourse, focusing on the generation of story events and then simply reporting those events with this direct realization strategy [Lebowitz, 1983, Meehan, 1976, Riedl and Young, 2006, Turner, 1994]. In contrast to the Storybook and Curveship narrative systems, Tale-Spin prioritizes planning mechanisms in order to automatically generate complex story event structure, yet no work has been done to automatically map the semantic representation to syntactic structures that allow the story to be told and varied in natural language. The TalesSpin text realization pulls from a series of hardcoded story points, which are time consuming to craft, and these realizations have no underlying syntactics that can be modified once selected.

Bae et al. [2011] use a computational model of focalization in generating narratives. A planning-based generation engine identifies which events or inner thoughts characters are aware at the moment, including character's belief, desire, hidden intent, or inner state of mind. These effect the selection of the events the planner selects to tell. For example, in Bae and Young [2009], stories with surprise endings are generated by exploiting the disparity of knowledge between a story's reader and its characters. Ware and Young [2011, 2012], Ware et al. [2014] describe many narrative systems that avoid conflict or make assumptions about its structure, or rely on humans to author it, creating a system based on character worlds, plans, their intentionality, and goals. In comparison to our own work, these works purposefully do not prioritize diversity in language generation.

2.2.3 Interactive Narrative Systems

The introduction of automatically authored dialogues using expressive NLG engines is explored in recent work [Cavazza and Charles, 2005, Lin and Walker, 2011, Montfort et al., 2014, Rowe et al., 2008]. Walker et al. [2013] use the dynamic and customizable PERSONAGE system to generate a variety of character styles and realizations as one way to help authors to reduce the authorial burden of writing dialogue instead of relying on scriptwriters. Yet the authorial burden is still high, as authors now have to manually produce detailed syntactic structures to provide content.

The input to SpyFeet, as we previously discussed, is a text plan from the Inform7 IF platform, which acts as the content planner and manager. Reed et al. [2011] show that this architecture allows diverse character personalities to be used in any game situation. SpyGen, as well as PERSONAGE, requires manual construction of content and text plans, which is time consuming and does not easily yield content reusability for different domains.

Another interactive narrative system is Crystal Island, an implementation of the narrative centered learning environment [Rowe et al., 2009]. The player is charged with discovering the cause of a mysterious disease is plaguing a research team on an island. The game's contents are based upon an 8th grade microbiology curriculum. Through a graphical environment, the player experiences the game through the first person perspective and is instructed to speak with the non-player characters of the research team to learn more about the illness.

Crystal Island is completely customized to the player's decisions throughout the game and will step in to help guide the player along. If it seems the player is not making progress, the game will give a hint, planning and replanning the story using a hierarchical task network (HTN) to generate sequences that can be constructed into coherent and engaging stories.

The construction of these HTNs is similar to that of Universe [Lebowitz, 1985], a dynamic story generator that utilizes story level goals in which the author first specifies broad narrative goals which execute subgoals. At the bottom level, a plot fragment is eventually reached and if all the preconditions are met, it is executed. In Universe, execution prints out a concrete piece of preauthored and inflexible story text. Instead, in Crystal Island, the narrative HTN will send the physical logistics of the event to the world model so it can update the model and graphics.

When creating this replayable and adaptable system, it is important to have believable non-player characters (NPCs). However the dialogue generation in Crystal Island is limited to a partially hand authored library that is not as flexible as the adaptable aspects of the narrative. One student may prefer to be guided by NPCs that are helpful and kind, whereas another might prefer NPCs to be more demanding, or limit the dialogue from the NPCs. In terms of believability, if the player replays the game and gets a different story, the dialogue should be different from other stories, but the current framework suggests this is not the case. If two stories have the same cause of the mysterious disease, the dialogue does not change as the player interacts with the world. We posit the Crystal Island architecture would benefit from a union with a sentence planner such as Fabula Tales.

2.2.4 Data-Driven Narrative and Dialogue Generation

Rather than carefully curate content or syntactic representations as some predefined and assumed input to a narrative system, some works use natural language processing techniques to plan and generate stories and dialogue. Swanson and Gordon [2008]'s Say Anything takes turns with the user to co-construct a story using stories from the Spinn3r blog corpus [Burton et al., 2009], which we use in our own work. After the user takes a turn to enter text to the system, the algorithm mines for similar sentences in stories in the corpus using TF-IDF. Once a sentence from an existing blog story in this corpus is retrieved, the system returns the next sentence from that story to the user, progressing forward the co-constructed narrative.

Munishkina et al. [2013]'s work automatically builds an interactive game by mining from movie scripts, such as Indiana Jones. Key information is preprocessed and extracted from the script, including characters, the setting, events and dialogues. The player is shown part of the extracted movie script and they have the opportunity to select the next action which is a set of retrieved actions from the script, also obtained using TF-IFD. The selected action is narrated and the player is presented with a new set of actions from which to select, similar to a "choose your own adventure" experience.

These textual learning approaches are advantageous for domain independence and have coverage over different prose styles. However, they make no attempt to vary the resulting text, as it is simply retrieved from the original story or the movie script. These systems lack an interpretable understanding of the story or the semantic relationships between concepts, as compared to systems that do model semantics, including the EST.

A more advanced approach to data-driven story generation is the introduction of neural networks for language generation [Ritter et al., 2011, Sordoni et al., 2015, Vinyals and Le, 2015]. A model, typically an RNN or an LSTM, learns how to respond in a dialogue by observing common patterns in large amounts of data. System responses are generated word by word, predicting the next most likely word in the sequence conditioned on the context of the conversation so far and the words selected in the response so far. These models can be manipulated to learn express personality models from the data [Li et al., 2016].

The Scheherazade story generation system [Li, 2015], not to be confused with Elson's Scheherazade tool annotation tool for creating SIGs, prioritizes both content planning and stylistic variation using a data-driven approach. The *fabula* is defined by a set of tuples representing events, relationships between the events, and a set of required and optional paths from and between story points. The algorithm examines the graph and generates a story following the paths, such as the robbery scenario previous discussed. A benefit of this data-driven approach is that it is domain independent. Any story can be represented along these dimensions. However, it is time consuming and tedious to construct such a graph in a new domain.

An early version of the Scheherazade story system could generate only the baseline text associated with each event tuple [Li et al., 2013]. However, improvements allow it to produce text by mining through large amounts of data [Li, 2015]. A set of descriptive sentences are clustered around a single event and are retrieved by computing probabilities of similarity between the most likely sentences in the cluster. The event "John covers face" in the bank robbery domain, returns a likely realization *John put on a fake mustache* as a possible retrieved sentence of the story event, and a less likely realization *John kept his head down as he pulled* *open the outer door and slipped his Obama mask over his face.* In addition, the most fictional response is also retrieved, providing more subjective emotions and intentions: *John looked at his reflection in the glass of the door, gave himself a little smirk and covered his face.*

These data-driven approaches require an enormous amount of training data, and even so, sometimes have difficulty with personalization beyond generic responses. To overcome this, researchers often fine-tune their models and change their objective functions. However, this requires intimate knowledge of the algorithm, and may not be easily compatible for integration with an interactive narrative environment, a long term vision of our work. These approaches tend to lack an interpretable semantic understanding of the relationship between story points due to the fact that the story facts are encoded and optimized for statistical, machine learning, or text retrieval approaches.

Chapter 3

The Expressive Story Translator and Content Planning

The first goal of this thesis is to create an NLG storytelling system with a semantic-to-syntactic mapping to bridge the NLG gap. By surveying previous narrative systems discussed in Chapter 2, and working towards the second goal of this thesis to generate diverse sentence planning variations, we require that our system is domain independent, is semantically interpretable, can support content planning manipulations, and will produce stylistic variations through sentence planning based on a syntactic representation.

We propose to render the SIG, acting as the *fabula* and content pool, to deep syntactic structures (DSYNTS) by creating a semantic-syntactic representation. We review these existing SIG and DSYNTS frameworks in Section 3.1.1 and Section 3.1.2 respectively. The SIG provides a deep representation of the structure of the story and character intentions that are encoded as propositions with WordNet [Fellbaum, 2010] and VerbNet [Kipper et al., 2006]. Furthermore, the annotation tool, Scheherazade, is available with the SIG framework to create new SIGs from any genre of story. The DSYNTS representation, which is compatible with the PERSONAGE generation engine, is optimized for parameterization of different sentence planning styles or voice models.

However, even using these rich representations, there is still a disconnect in the model proposed thus far, and each existing framework by itself is insufficient to bridge it. The SIG features a built-in generation module that paraphrases the story using templates and lexical realizations, called the what-you-see-is-what-you-mean paradigm (WYSIWYM). This WYSIWYM realization offers no flexibility by design, which makes it difficult to repurpose the modeling. No content planning takes place in the WYSIWYM realization. On the other hand, DSYNTS are used in PERSONAGE to support rich variation, yet the representation is computationally difficult to manually produce and is thus an authorial burden. The time and skill burden of creating DSYNTS slows the content creation of applying PERSONAGE to exploring different domains.

We present the Expressive Story Translator (EST) to bridge the NLG gap and offer domain independent, narratologically informed content planning. The motivation of our modeling is twofold: first, the EST uses a semantic-syntactic representation to preserve semantically encoded facts from the lexical choice and temporal order of the SIG representation. Second, the EST automatically generates DSYNTS linguistic structures from these semantic-syntactic structures, creating a generation dictionary for realization (Section 3.2).

The EST is not only a direct mapping from SIG to DSYNTS; the EST also operates as a content planner. The EST can use the encoded semantics in addition to unexplored usage of the interpretation layer of the SIG for content planning. One such modeling is of character

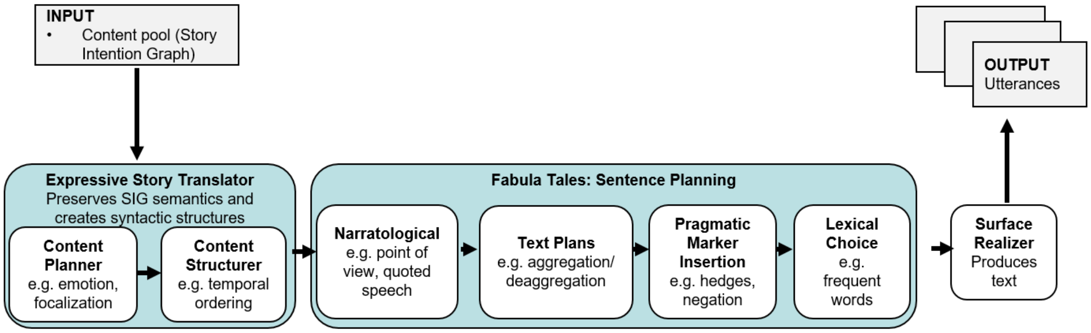

INPUT
• Content pool (Story Intention Graph)
Expressive Story Translator
Preserves SIG semantics and creates syntactic structures
Content Planner
e.g. emotion, focalization
Content Structurer
e.g. temporal ordering
Fabula Tales: Sentence Planning
Narratological
e.g. point of view, quoted speech
Text Plans
e.g. aggregation/ deaggregation
Pragmatic Marker Insertion
e.g. hedges, negation
Lexical Choice
e.g. frequent words
Surface Realizer
Produces text
OUTPUT
Utterances

Figure 3.1: The Expressive-Story Translator and Fabula Tales NLG pipeline

appraisal (Section 3.3.1). We explore other content planning operations within the EST framework, including temporal reordering (Section 3.3.2), and briefly enumerate over other content operations, such as verbosity (Section 3.3.3). Despite the advances the EST affords us, there are some limitations of the model thus far that we describe and allude to in future work.

Figure 3.1 shows our proposed NLG pipeline. From the SIG model of a story, content is translated and selected through the EST (Section 3.2). Section 3.4 gives a walkthrough of the semantic-syntactic construction. Content planning decisions can be made and enhanced for narrative effects, such as modeling emotions and focalization, as we discuss in Section 3.3. With the resulting DSYNTS and text plans from the EST, our Fabula Tales sentence planner operates over the semantic and syntactic structures to generate different points of view, direct speech, and character voice (Section 4.1).

3.1 Existing Framework

3.1.1 Story Intention Graphs

The STORY INTENTION GRAPH (SIG) is a representation of a story along a variety of dimensions, including plans, goals, and actions of characters. The SIG formalism creates a computational model of narrative that goes beyond the surface form of a text to compare and contrast stories based on content, as opposed to style; "what the story *is* [*fabula*] and how the story is *told* [*sujet*]" [Elson, 2012a]. This formalism emphasizes key elements of a narrative rather than attempting to model the entire semantic world of the story. The creation of these models invites untrained human subjects to use the Scheherazade annotation tool to create an open-domain corpus of SIGs [Elson and McKeown, 2009]. The annotation process and how we repurpose the guidelines are further discussed in Section 4.2.1.

The SIG offers the inverse strategy of transforming a single *fabula* into many *sujet*; instead, the SIG breaks down one *sujet* to derive the underlying *fabula*. A caveat is that, as is the nature of the *sujet*, the telling is only one interpretation of a larger narrative discourse. Some events may not be made explicit in the *sujet* from which we transform into a SIG, so they may be excluded from the derived *fabula*. For our storytelling purposes, we treat the derived *fabula* as one possible diegetic interpretation and leave the discussion of constructing a larger *fabula* from many *sujet* to future work.

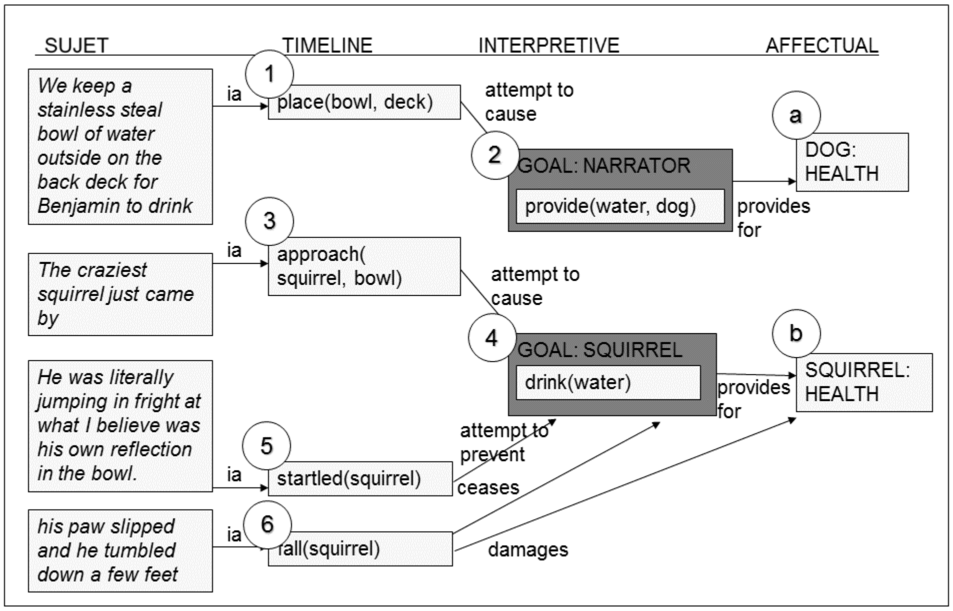

SUJET
TIMELINE
INTERPRETIVE
AFFECTUAL
We keep a stainless steal bowl of water outside on the back deck for Benjamin to drink
The craziest squirrel just came by
He was literally jumping in fright at what I believe was his own reflection in the bowl.
his paw slipped and he tumbled down a few feet
ia
ia
ia
ia
1
place(bowl, deck)
3
approach(squirrel, bowl)
5
startled(squirrel)
6
fall(squirrel)
attempt to cause
2
GOAL: NARRATOR
provide(water, dog)
attempt to cause
4
GOAL: SQUIRREL
drink(water)
attempt to prevent
ceases
damages
provides for
provides for
a
DOG: HEALTH
b
SQUIRREL: HEALTH

Figure 3.2: A Story Intention Graph for *Startled Squirrel*

3.1.1.1 Story Representation

A story in the SIG formalism is represented by four layers, *sujet*, timeline, interpretive, and affectual, connected by arcs signifying discourse relationships between the nodes in each layer. The original story (the *sujet*), is the first layer represented in the first column in Figure 3.2 (derived from the *Startled Squirrel* in Table 3.1). The *sujet* is divided into textual segments, each segment representing a point in the story.

The next three layers of the SIG comprise different elements of the *fabula*. The timeline layer provides a link between the sujet and interpretation layer. The interpretation layer has character goals and plans linking to the outcome in the affectual layer. Each layer has nodes derived from VerbNet frames with WordNet story elements. There are a fixed set of arcs that can be used to connect the nodes of the *fabula* layers.

> We keep a large stainless steel bowl of water outside on the back deck for Benjamin to drink out of when he's playing outside. The craziest squirrel just came by- he was literally jumping in fright at what I believe was his own reflection in the bowl. He was startled so much at one point that he leap in the air and fell off the deck. But not quite, I saw his one little paw hanging on! After a moment or two his paw slipped and he tumbled down a few feet. But oh, if you could have seen the look on his startled face and how he jumped back each time he caught his reflection in the bowl!

Table 3.1: *Startled Squirrel*

The second dimension of the SIG (second column in Figure 3.1) is called the "timeline layer", in which the story segments are encoded as predicate-argument structures (propositions) and temporally ordered on a timeline paralleling the surface form of the story. Each point in the timeline layer is associated with a span of text from the *sujet* to maintain the association by an interpreted as (*ia*) relation between the *sujet* and the timeline layer. Arcs between the *sujet* layer and timeline layer are always *interpreted as* (see *ia* arcs in Figure 3.2).

The third dimension (third column in Figure 3.2) is called the "interpretative layer". This layer is intended to capture the annotators' interpretation of *why* characters were motivated to take the actions they did, adopting a "theory of mind" approach to modeling narratives [Palmer, 2007]. Unlike the timeline which summarizes the actions and events that occur, the interpretation layer attempts to capture story meaning derived from agent-specific plans, goals, attempts, outcomes and affectual impacts.

The "actualization" arcs include *implies*, *interpreted as* link the same sentiment across two layers. *Actualizes* and *ceases* support or deny the fulfillment of a goal in the interpretation layer by some action in the timeline layer. "Plan" arcs allow the SIG to model strategies by characters to achieve their goals. When modeling a plan, *precondition for* and *precondition against* arcs must be first be satisfied. *Would cause* and *would prevent* arcs indicate nodes in the

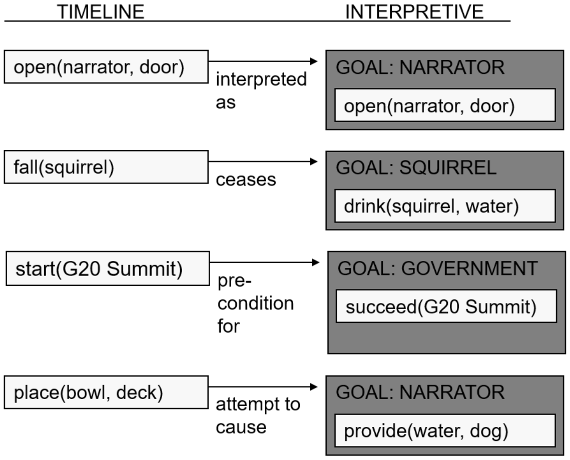


Figure 3.3: Sample discourse relationships (arcs) in SIGs

interpretation layer that the agents believes would affect another node. “Attempts” arcs connect actions from the timeline layer that *attempt to cause* or *attempt to prevent* a goal. “Affects” arcs connect the interpretation layer to the affectual layer. The node in the interpretation layer either *provides for* or *damages* the affect node (arcs and nodes in Figure 3.4).

The final dimension in the SIG is the affect layer (fourth column in Figure 3.2). This layer represents the deeper motivations that underlies character goals and the effect these goals have on the characters. The affect nodes themselves are devised from Maslow’s hierarchy of needs [Maslow, 1943] and Max-Neef’s classification of needs [Max-Neef et al., 1992] and include life, health, and wealth. The affect nodes are connected to interpretation nodes via the *provides for* and *damages* arcs. We observe in our personal narratives three common affect nodes: life (continuation of basic life functions), health (freedom from pain, disease, malnutrition, and other physical/mental ailments), and leisure (entertainment and enjoyment). Figure 3.4 shows examples of these affect nodes in excerpts from *Bug Out For Blood* and *Startled Squirrel*.

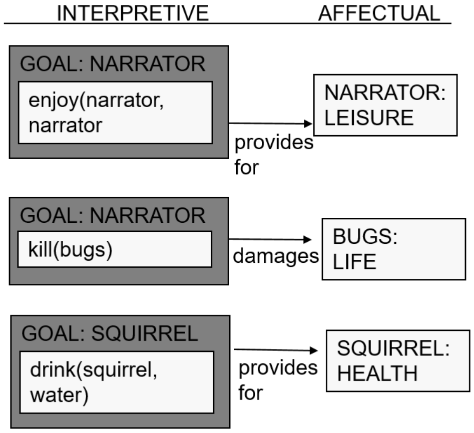


Figure 3.4: Most common Affect nodes in personal stories

We provide a brief walkthrough of how to interpret SIGs: numbers indicate timeline or interpretation events, and letters are affect nodes in the *Startled Squirrel* SIG (Figure 3.2). The narrator places a bowl on the deck (#1) as an *attempts to cause* the goal of the narrator to give the dog some water (#2) which would *provide for* the dogs' health (a). Then the squirrel approaches the bowl (#3) as an *attempt to cause* the squirrel's goal to drink the water (#4) which would *provide for* the squirrel's health (b). When the squirrel is startled (#5), this *attempts to prevent* the goal of drinking the water, and when the squirrel falls (#6) this both *ceases* the goal (#4) and *damages* the squirrel's health (b).

3.1.1.2 Assigned Predicates and WYSIWYM realization

The Scheherazade annotation tool features a built-in generation module that paraphrases the story as the annotator performs annotations, using templates and lexical realizations from WordNet nouns and VerbNet frames. This realization is called the what-you-see-is-what-you-mean paradigm (WYSIWYM) and is a direct realization of the underlying SIG semantics

Original text	WYSIWYM
Today was a very eventful work day. Today was the start of the G20 summit.	
It happens every year and it is where 20 of the leaders of the world come together to talk about how to run their governments effectively and what not.	The group of leaders was meeting in order to talk about running a group of countries and near a workplace.
Since there are so many leaders coming together their are going to be a lot of people who have different views on how to run the government they follow so they protest.	A group of peoples protested because the group of leaders was meeting and began to be peaceful.
There was a protest that happened along the street where I work and at first it looked peaceful until a bunch of people started rebelling and creating a riot. Police cars were burned and things were thrown at cops. Police were in full riot gear to alleviate the violence.	A group of peoples stopped being peaceful, began to be riotous, burned a group of police cars and a group of peoples pelted a group of police officers.

Table 3.2: *Protest Story* and WYSIWYM realization

[Bouayad-Agha et al., 1998]. Table 3.2 shows excerpts from the *Protest Story*. Complete sentences or partial phrases are segmented by the annotator in the original story and linked with an *interpreted as* arc to the corresponding Scheherazade AssignedPredicate and generated WYSIWYM realization.

Every span of text in the original story can be associated with a story point object, a SIG data structure called an AssignedPredicate. This object included information about the timeline in which the actions and its arguments. Figure 3.5 shows an example of the semantic AssignedPredicate representation. The realization strategy from AssignedPredicates is direct and simple. WYSIWYM is not associated with theories of syntax and does not attempt to produce narrative variation; instead only produces the text for the selected encoding. This is a shortcoming of using the SIG in our modeling framework: we don't have access to changing

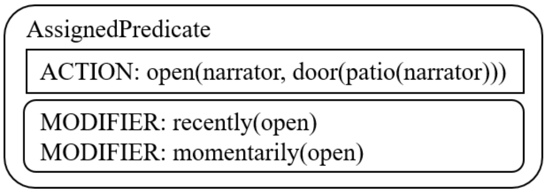

AssignedPredicate
ACTION: open(narrator, door(patio(narrator)))
MODIFIER: recently(open)
MODIFIER: momentarily(open)

Figure 3.5: SIG AssignedPredicate semantics

style. This motivates our choice of DSYNTS as a supplement to the SIG semantics.

3.1.2 Deep Syntactic Structures and Text Plans

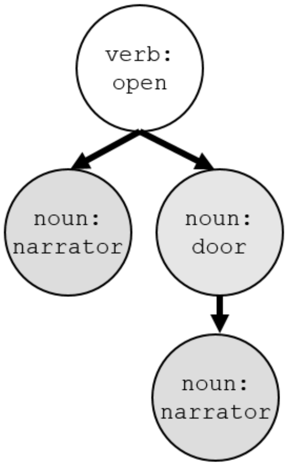
In order to manipulate stories on the syntactic level while preserving semantic information, we use deep syntactic structures (DSYNTS) [Mel'čuk, 1988], which we briefly introduced as the input to PERSONAGE in Chapter 2. We discuss DSYNTS and their semantic companion, text plans, in more detail with PERSONAGE as the frame of reference.

PERSONAGE manipulates a sentence plan tree whose internal representations are DSYNTS. DSYNTS provides a flexible dependency tree representation of an utterance which can be altered by the PERSONAGE parameter settings. The nodes of the DSYNTS syntactic trees are labeled with lexemes and the arcs of the tree are labeled with syntactic relations. The DSYNTS formalism distinguishes between arguments, modifiers, and between different types of arguments (subject, direct and indirect object etc). Lexicalized nodes also contain a range of grammatical features used in generation. DSYNTS are passed to the real-time surface realizer RealPro, and realized as natural text [Lavoie and Rambow, 1997]. RealPro handles morphology, agreement and function words to produce an output string.

Table 3.6a shows the DSYNTS representation for the sentence "The narrator opened

```
<dsyntnode class="verb" lexeme="open" rel="II"
    tense="past">
  <dsyntnode article="def" class="common_noun"
      gender="fem" lexeme="narrator" number="sg"
      rel="I"/>
    <dsyntnode article="no-art" class="common_noun"
        gender="neut" lexeme="door" number="sg"
        rel="II">
      <dsyntnode article="no-art" class="common_noun"
          gender="fem" lexeme="narrator" number="sg"
          pro="poss" rel="I"/>
    </dsyntnode>
  </dsyntnode>
</dsyntnode>
```

(a) DSYNTS in XML format



(b) DSYNTS in tree format

Figure 3.6: DSYNTS for *The narrator opened her door*

her door". The nested structure of the DSYNTS can be visually interpreted as a tree (Figure 3.6b). DSYNTS are ordered in structure; the root is the main verb with required properties in the XML format, including lexeme and tense. The *rel* argument indicates the relationship of the argument with respect to its parent. Nouns require an *article* argument, indicating a definite or indefinite article. Additionally, they can have a *gender* and *number*. Possession is represented structurally, so "her door" is structured with door as the parent, and "the narrator" as the child, with the child also being possessive (*pro*). Gender, tense, coreferences, and articles are automatically handled by RealPro at generation time.

Table 3.3 shows generated utterances from the PERSONAGE introvert and extrovert personality models making use of various sentence planning operations. We select a few and dissect the representation below. The order and placement of the nodes can be restructured, and nodes can be added or subtracted to achieve a desired effect. Pragmatic marker insertion inserts

Model	**Parameter**	**Example**
Shy	Softener hedges	The somewhat crazy squirrel desperately held the deck's railing.
	Stuttering	The cr-crazy squirrel fell.
	Filled pauses	The squirrel cautiously approached the err... steely bowl.
Laid-back	Emphasizer hedges	The narrator grabbed the obviously rolled comic book.
	Exclamation	The bugs scared the narrator!!
	Expletives	The narrator killed the damn bugs.

Table 3.3: Examples of pragmatic marker insertion parameters from PERSONAGE.

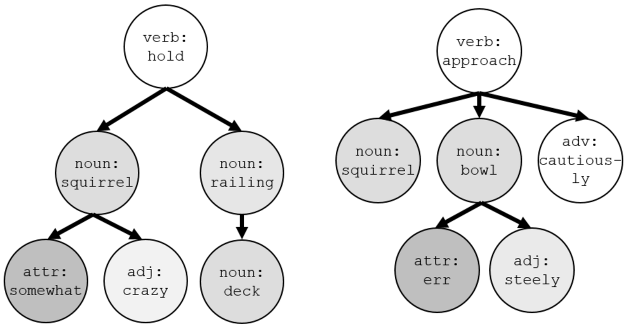
```
<dsyntnode class="verb" lexeme="fall" mode="" mood="ind" rel="II" tense="past">
  <dsyntnode article="def" class="common_noun" gender="neut" lexeme="squirrel"
      number="sg" person="" rel="I">
    <dsyntnode class="adjective" lexeme="cr-crazy" rel="ATTR"/>
  </dsyntnode>
</dsyntnode>
```

Table 3.4: Stuttering: "The cr-crazy squirrel fell."

various items into the existing DSYNTS. From Table 3.3, the softener hedges are inserted into DSYNTS by adding a new node. Figure 3.7a show the dsynts in textual and tree form for the sentence *The somewhat crazy squirrel desperately held the deck's railing.* Adding filled pauses (Figure 3.7b) and emphasizer hedges follow a similar pattern as softener hedges.

To support stuttering, PERSONAGE randomly selects a node from the DSYNTS tree, and repeats the first syllable in the lexeme. Table 3.4 show the DSYNTS for the sentence *The cr-crazy squirrel fell.* To support exclamations, no changes are made to the structure, but to the XML attributes. PERSONAGE can set a parameters at the root of the DSYNTS tree indicating the punctuation to use.

Syntactic template selection controls the complexity of the statement, "Le Marais is



(a) Softener: "The somewhat crazy squirrel desperately held the deck's railing."

(b) Filled pauses: " The squirrel cautiously approached the, err, steely bowl."

Figure 3.7: DSYNTS examples in tree format

the best" is rated as less complex than "Le Marais is one of my favorite restaurants". Lexical choice performs a substitution of a random lexeme in a sentence, if one exists. From Mairesse and Walker [2011], "it has decent service" is transformed to "it features friendly service".

In addition to manipulation of a single DSYNT, we can draw relationships between two DSYNTS with a text plan. Text plans are inspired by Rhetorical Structure Theory (RST) Mann and Thompson [1988] and model the semantic relationship between syntactic clauses. PERSONAGE supports several relationships including the JUSTIFY relation, offering a justification of the nucleus clause via the satellite clause.

A subset of sample content facts within a content pool are shown in Table 3.5, with justify relationships between facts. Table 3.6 shows the text plan with the justify relationship to call fact 1 the nucleus and 4 the satellite, and the associated textual realization is in Table 3.5. Other content planning operations include leaving off the satellite completely, or joining to-

Content	1: assert(best (*Le Marais*)) 2: assert(is (*Le Marais*, cuisine (*French*))) 3: assert(has (*Le Marais*, food-quality (*good*))) 4: assert(has (*Le Marais*, service (*good*))) 5: assert(has (*Le Marais*, decor (*decent*))) 6: assert(has (*Le Marais*, price (*44 dollars*)))

Relation	Realization
nuc:1	LeMarais is the best.
JUSTIFY (nuc:1, sat:4)	LeMarais is the best because it has good service.
JUSTIFY (nuc:1, sat:3, sat:5)	LeMarais is the best because the food-quality is good and the decor is decent.

Table 3.5: Content plans and realizations for justify

```
<rstplan>
    <relation name="justify">
      <proposition id="1" ns="nucleus"/>
      <proposition id="2" ns="satellite"/>
    </relation>
  </rstplan>
  <proposition dialogue_act="1" id="1"/>
  <proposition dialogue_act="4" id="2"/>
</speechplan>
```

Table 3.6: A justify text plan

gether more than one satellite. We discuss in Section 3.2 how we use these text plans to preserve the *fabula* from the SIG.

A more recent version of PERSONAGE has been developed to make it easier to adjust parameters and implement new ones. PyPersonage performs similar sentence planning functionality to PERSONAGE, and is compatible with DSYNTS [Bowden et al., 2016]. PyPersonage is aimed at generating dialogue and breaks down a set of DSYNTS to be rendered into two sets of DSYNTS to simulate a dialogue. To this end, PyPersonage takes the entire discourse into consideration in order to correctly allocate speaker turns and support dialogic features such as entrainment and speaker repetition in co-telling a story.

PERSONAGE and PyPersonage already have the affordances of DSYNTS and text plans but are unable to serve as-is as the sentence planner in our storytelling pipeline. Both systems require as input DSYNTS, which up until now has been manually crafted and time consuming. The EST introduces the first end-to-end pipeline making use of DSYNTS with automatic tools for DSYNTS creation and a generation dictionary.

While PERSONAGE and PyPersonage are able to control many aspects of the dialogue generation, they do not store information about the narrative or consider narrative content or sentence planning and its effects on storytelling. The EST is a new model that maintains both semantic and syntactic story information providing for new content planning and manipulation of the *fabula*, as well as introducing the sentence planning stylistic variations.

3.2 Expressive-Story Translator

The Expressive Story Translator[1] (EST) is motivated by a simple observation: a flexible semantic-to-syntactic mapping in NLG systems can be achieved by automatically transforming the semantic representation of the SIG to the joint semantic-syntactic representation of DSYNTS and text plans [Rishes et al., 2013]. The SIG affords us a rich modeling of story at the *fabula* level, but is not suited for expressive generation on its own because the WYSIWYM generation does not offer the flexibility to induce content or sentence planning.

Our new lexico-syntactic representation is designed to be compatible in theory with DSYNTS, thus allowing any SIG transformed by the EST to take advantage of the technologies

[1] https://bitbucket.org/smcl/es-translator/wiki/Home

compatible with DSYNTS. This approach overcomes the handcrafting of DSYNTS that dialogue authors using the PERSONAGE framework have had to face. The advantage of handcrafting is that authors can finely tune utterances to ensure no morphologically odd sentences. However, this technique for DSYNTS generation is time consuming and only skilled annotators and linguists are capable of such a feat in a short amount of time. If authors wanted a new domain or set of utterances supported, each one must be hand crafted. This automatic translation produces a generation dictionary of the story.

The EST translation methodology provides a trade-off in annotation time and difficulty: instead of requiring skilled experts to create DSYNTS, we require minimally trained annotators to create a SIG. This trade off is worthwhile and justifiable because SIGs can be annotated in under an hour (time trade-off), and it makes use of the existing lexical ontologies provided by WordNet and VerbNet, to allow for morphological diversity in retelling.

The EST transformation allows for the SIG content to remain salient, captured within text plans created by the EST. We use these relations to generate narrative content planning such as appraisal modeling, temporal order, and causality, and other non-narrative specific content planning operations, such as repetition and verbosity.

This semantic and syntactic integration of the EST allows for the Fabula Tales sentence planner to employ narrative devices to change point of view, insert direct speech acts, and supplement character voice using operations for lexical selection, aggregation, and pragmatic marker insertions, as we introduce in Chapter 4. The rest of this section describes the creation of semantic plans and a generation dictionary by the EST.

3.2.1 Semantic Modeling

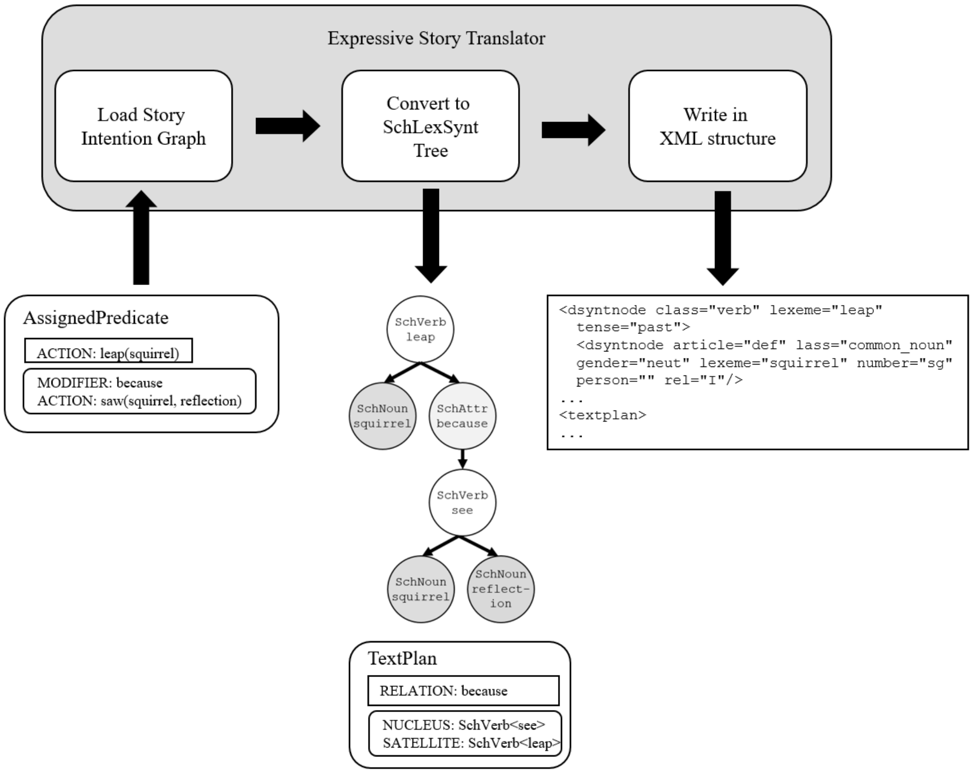
The EST constructs text plans to preserve semantic relationships from the SIG in order to utilize in content planning and sentence planning. Figure 3.8 shows the high level EST framework, which Section A describes in greater implementation detail. The EST reads the AssignedPredicate Scheherazade data structures, and semantic information about the narrative is preserved, using salient information from the SIG AssignedPredicates in the SIG timeline layer. Text plan classes are created, following the Rhetorical Structure Theory nucleus-satellite predicate assignments.

Because the EST framework iteratively examines each AssignedPredicate, and the AssignedPredicates themselves are in temporal order in the telling, the straightforward timeline of events is captured by the order in which the text plans are created. This open doors for temporal reordering of events in the telling, as we discuss in Section 3.3.2.

Deriving text plans from the SIG also allows us to capture relationships, such as if utterances are spoken in the direct speech (Section 4.1.2) or if clauses contain a contingency relationship and can be aggregated (Section 4.1.5). More content relationships can be easily extended in this general framework, including using the relationships in the interpretation layer as we explain in Section 3.3.1.

3.2.2 Syntactic Modeling and a Generation Dictionary

We define a new set of lexico-syntactic classes to map the semantic information from the SIG directly onto the syntactic model of DSYNTS. The EST constructs lexico-syntactic classes from the content received from the content planner. The lexico-syntactic classes cor-

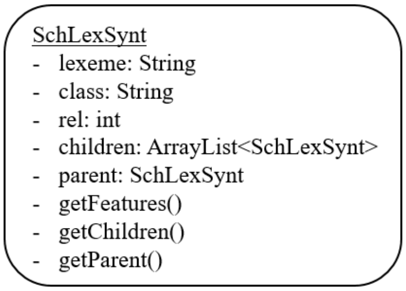


```
<dsyntnode class="verb" lexeme="leap"
  tense="past">
  <dsyntnode article="def" lass="common_noun"
  gender="neut" lexeme="squirrel" number="sg"
  person="" rel="I"/>
...
<textplan>
...
```

Figure 3.8: ES-Translator overview

respond roughly to parts of speech: verbs, nouns, adjectives, and prepositional phrases. To populate our new Scheherazade Lexico-Syntactic classes (**SchLexSynt**), we extract relevant information from each AssignedPredicate story point. This section focuses on defining each SchLexSynt class and describing the information required by DSYNTS, and Section A describes how to build the entire SchLexSynt tree.

SchLexSynt. SchLexSynt (Figure 3.9) is a generic Scheherazade Lexico-Syntactic node that every new class we create implements. The lexico-syntactic class aggregates all of the information necessary for generation of a lexico-syntactic unit in DSYNTS. According to the DSYNTS formalism described in Section 3.1.2, every DSYNTS node must have a class, lex-



```
SchLexSynt
-  lexeme: String
-  class: String
-  rel: int
-  children: ArrayList<SchLexSynt>
-  parent: SchLexSynt
-  getFeatures()
-  getChildren()
-  getParent()
```

Figure 3.9: Scheherazade SchLexSynt class

eme, and relation; these are required fields in the SchLexSynt class. Each class implementing SchLexSynt has a list of properties such as morphology or relation type that are required by the DSYNTS notation for a correct rendering of a category. These properties are mapped to associated DSYNTS features.

SchLexSynt objects have built in methods to reference the elements each SchLexSynt node governs and the element that governs it. SchLexSynt nodes are iteratively constructed and result in a tree structure. The class properties are based on the modeling of DSYNTS and the accessor methods allow for manipulation of the resulting tree structure. The SchLexSynt tree is modeled similarly to DSYNTS with a direct mapping between SchLexSynts structure and properties and formatted XMLs.

SchVerb is the lexico-syntactic class for verbs. In addition to attributes it inherits from SchLexSynt, SchVerb requires the following: tense, mood, and polarity. These key attributes are highlighted in Figure 3.10, but there are many more fields maintained by SchVerb and the other SchLexSynt subclasses to determine the best way to derive these import attributes from corresponding semantics from the AssignedPredicate. *Tense* is set to past by default, but in conjunction with the *mood* field, it can be changed to support to infinitive, gerunds or imper-

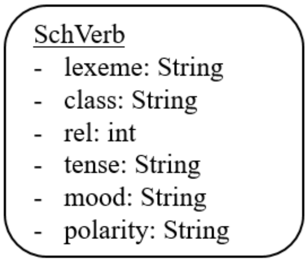


Figure 3.10: SchVerb class

ative. *Polarity* can be inferred from the AssignedPredicate, potentially negating the verb.

SchNoun is the lexico-syntactic class for nouns and are appended as subjects and objects of SchVerbs. The information that DSYNTS need is pronoun (for coreference), article, person, and gender (Figure 3.11). *Article* is defaulted to definite unless otherwise specified, *person* to third, and *gender* derived from the annotated noun. *Number* defaults to singular unless it is a group of nouns, which then becomes plural. *Pronoun* is set by examining if this is a possessive noun or not, which can be inferred from the arguments.

SchAdv is the lexico-syntactic class for adverbs derived from modifiers attached to an AssignedPredicate. In addition to the required fields, it requires a *position* indicating pre or post verbal. This is fixed to preverbal, rendering the adverb before the verb (e.g. "The narrator recently opened the door") but can be set to postverbal ("The narrator opened the door recently"). SchAdvs are appended as the children of SchVerbs.

SchAdj is the lexico-syntactic class for adjectives and do not require additional attributes. SchAdjs are appended as children to SchNouns. **SchFunc** acts as a catch all for prepositions and does not require additional fields. SchFunc's can be used for coordination, where the lexeme is "and", or for prepositional phrases. **SchAttr** acts as a class for "extra"

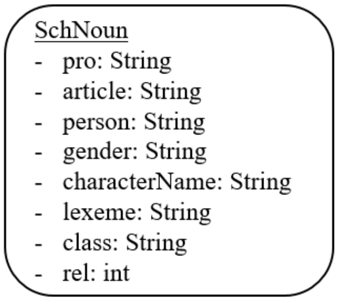


Figure 3.11: SchNoun class

lexico-syntactic nodes and is not directly related to Scheherazade data types. The *lexeme* can be anything we define, used for special cases such as “if-then” statements where the lexeme is “if” or “then” and then the node is strategically positioned in the SchLexSynt tree.

These Lexico-Syntactic classes give us a syntactic representation of a set of semantics. Because they were designed with DSYNTS in mind, it is easy to map from these classes to the XML format required by DSYNTS formalism. The conversion algorithm of the EST traverses the syntactic trees in-order and creates an XML node for each lexico-syntactic unit. Class properties are then written to disk, and the resulting file is processed by the surface realizer to generate text. Section 3.4 gives a complete walkthrough of the transformation process for a subset of story points in the *Bug Out For Blood* story. The algorithm behind the translator process is detailed in Section A.

3.3 Content Planning and Structuring

The EST is more than a one-to-many syntactic mapping. The extracted semantics from discourse relationships derived from the timeline layer of the SIG are used for content

Desirability	Certainty	Emotion
Desirable	Certain	Joy
Desirable	Uncertain	Hope
Undesirable	Certain	Distress
Undesirable	Uncertain	Fear

Table 3.7: Emotions defined by Appraisal Theory

planning. The interpretation layer in particular is of great use for in-depth story modeling. We explore additional semantic relationships the EST captures in this section.

3.3.1 Emotion Modeling

Insights into the emotions of characters could create contextually appropriate character reactions to events in the story, changing the style of a narrative as events happen to the protagonist or other characters. The SIG offers rich information about character goals and plans we use to derive character emotions from this layer according to appraisal theory [Ortony et al., 1990].

Appraisal theory describes how agents evaluate a situation based on the desirability and certainty of the scenario. Previous approaches to computationally modeling appraisal theory, include models that appraise as a dynamic scenario unfolds, [Gratch and Marsella, 2001, 2003, 2004], models that appraise cultural behavior [Bulitko et al., 2008], and models that appraise in emergent narratives for bullying scenarios [Aylett et al., 2005, Dias et al., 2011]. These models are useful for engaging users in tutoring applications and teaching cooperation by supplying emotional states to the agents in the simulations [Conati and Maclare, 2004].

Unlike previous work, we do not generate or calculate these appraisals from proce-

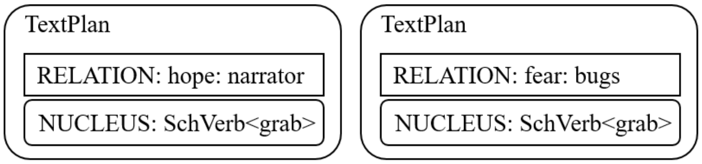


Figure 3.12: Emotional trajectory for narrator and bug in *Bug Out For Blood*

durally generated content, but extract them from the SIG encoding. We define a set of rules for each emotion following the appraisal rules in Table 3.7 by identifying similar patterns (and their inverses) in the SIG interpretation layer, for example hope can be defined by an attempt to cause, which is an uncertain outcome, some desirable goal, such as wealth or health.

We write rules in Prolog to be compatible with Prolog rules from Elson [2012b] for defining higher level goals and plans in the interpretation layer (see Appendix B). These rules are a part of the EST's content planner and associates these character appraisals with plot points in the story. Text plans are created (Figure 3.12) and the emotion is inserted as a new attribute into the DSYNTS.

By examining part of the SIG for *Bug Out For Blood*, we trace the emotional state of the narrator: The narrator is joyful when she opens the patio door because the action actualizes the goal of providing for the narrator's leisure. The narrator happiness continues until she is woken up, which ceases her leisure, causing distress (Figure 3.13). When the narrator sees the bugs, she attempts to squash the bugs which would provide for her leisure. Because this "attempt to squash" is an attempt to cause arc, it is hopeful, not certain, but desirable (Figure 3.14). When she finally squashes the bug, she is joyful because this event is certain (it happened) and desirable (provides for leisure).

In stories with multiple characters we find interesting trade-offs between characters

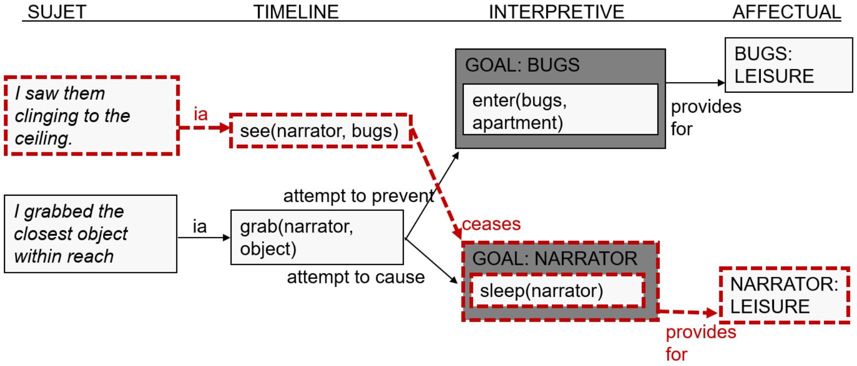


Figure 3.13: Distress trace in *Bug Out For Blood* SIG

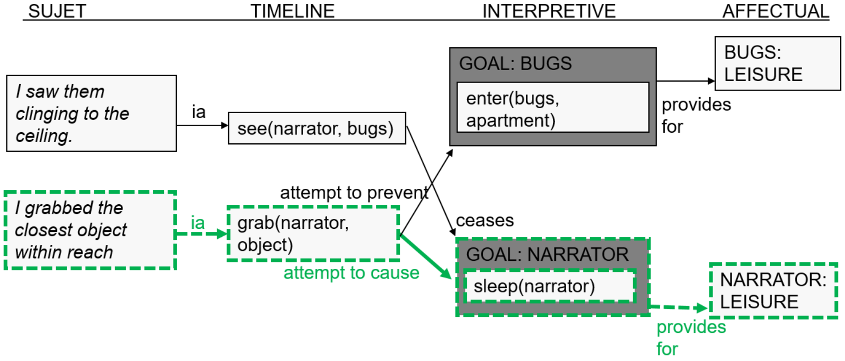


Figure 3.14: Hope trace in *Bug Out For Blood* SIG

with conflicting goals. Figure 3.15 shows how the emotions of the narrator and the bug oppose each other as the narrator's trying and actualization of squashing the bugs leads to her hope and joy, and the bugs to fear and distress.

Of course, this SIG represents one interpretation of the derived *fabula* from a particular *sujet*. Different responses could be modeled according to culture, personality, and situation. For example, in a new story, if the narrator finds themselves in a snake pit, a tourist probably didn't want that to happen, thus the event is appraised as fear or distress. On the other hand, an explorer might have sought out the snakes, and would result in a hope or joy appraisal.

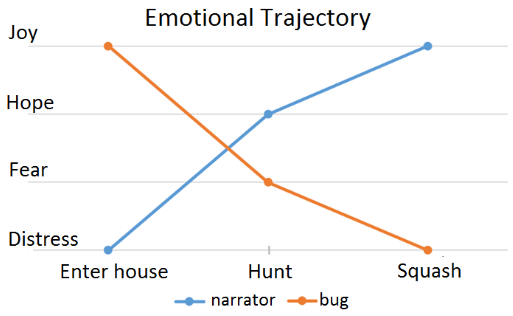


Figure 3.15: Emotional trajectory for narrator and bug in *Bug Out For Blood*

We test our rules on 21 stories that are annotated with an interpretation layer in the PersonaBank corpus, described in detail in Section 4.2, by applying the rules to get emotion at each story point for each character. We compare the output of the rules by manually examining the SIG to ensure that our rules have coverage of what we expect with respect to the appraisal rules. We have 100% recall and precision of the rules on the 194 nodes in all the 21 SIGs that have a link to an affect node. Although the SIG was not meant to model appraisal, and although appraisal is a complex mode, these general rules have excellent coverage with the existing computational and theoretical framework.

We are able to take the emotions appraised and selected by the EST and generate simple expression of the character's appraisal. Table 6.3 shows a basic surface realization of the appraisals by filling in templates for the character and the emotion associated with that character from the text plan.

This is a very simplistic textual generation approach with a complex underlying modeling, similar to the goal insertion from Pérez ý Pérez and Sharples [2001], or even story generation from Tale-Spin. This phenomena is dubbed the Tale-Spin Effect: "works that fail to represent their internal system richness on their surfaces" [Wardrip-Fruin, 2009]. We do not

The bugs entered the narrator's apartment.
The bugs were joyful.
The narrator despaired.

The narrator grabbed the comicbook.
The narrator was hopeful.
The bugs were fearful.

Table 3.8: Simple emotive observations in text

perform further evaluations because it is unclear yet the most effective way of expressing emotional states of characters. Is it through observations, such as this? Or through varied character language invoking sentence planning? Section 6.3.1 discusses future work on modeling and experimentation.

3.3.2 Temporal Manipulation

Empirical studies have shown that the temporal manipulation of discourse structure can produce different cognitive and emotional responses (e.g., suspense, curiosity, and surprise) by influencing the reader's inferences and anticipation [Bae and Young, 2009]. According to Structural Affect Theory [Brewer and Lichtenstein, 1980, 1982], these responses can be aroused by manipulating the temporal characteristics in narrative structure. For suspense, an outcome event is delayed until the last moment so that the reader is uncertain about the important story outcome. To elicit surprise, some significant expository information is hidden from the reader until a surprising event occurs, which makes a knowledge gap between the reader and some characters in the story [Bae and Young, 2009].

Prolepsis, or flashforwards, is a discrepancy between the order of events in the *sujet* and the order in which they occur in the *fabula* [Prince, 2003]. Conversely, analepsis or flashbacks, go back to the past with respect to the "present" moment, evoking one or more events that occurred before the "present" moment [Prince, 2003]. Analepsis can serve as a narrative device for revealing the backstory of a narrative setting or key character, as well as a tool for repeating and reinforcing prior narrative events [Rowe et al., 2010]. Foreshadowing is an event that indicates another event to come [Morson, 1996]. In contrast to analepsis, foreshadowing is a narrative device that hints about a future event in advance of its occurrence. While foreshadowing does not directly resequence events, it does foster increased awareness of future events in a non-chronological fashion [Rowe et al., 2010].

By default, the EST performs greedy, linear temporal ordering, and assumes all AssignedPredicates will be selected. However, we propose that the *fabula* of the SIG can be manipulated to tell stories out of order by finding a causal chain. By accessing the events in the SIG, which are encoded in temporal order, we can restructure the event frames to achieve a different telling order. For example, in Aesop's Fable *The Fox and the Crow*, the Fox could narrate his perspective of the events by telling the end of the story, then backtracking:

> I got the crow's cheese by making her sing, and told her she was dumb that she had let it go. It all started when I observed a crow in a tree with cheese in her beak.

The temporal selection method must be careful to select the correct information and not skip necessary events. The arcs in the interpretation layer give information like "precondition for", "provides for", "would cause", etc. that are captured to create chains of cause and effect. Patterns and subgroups of goals could be identified and then local reordering performed. Looking at the interpretation layer in Figure 3.16, each goal box is it's own subgroup; it can stand alone as a piece of the narrative:

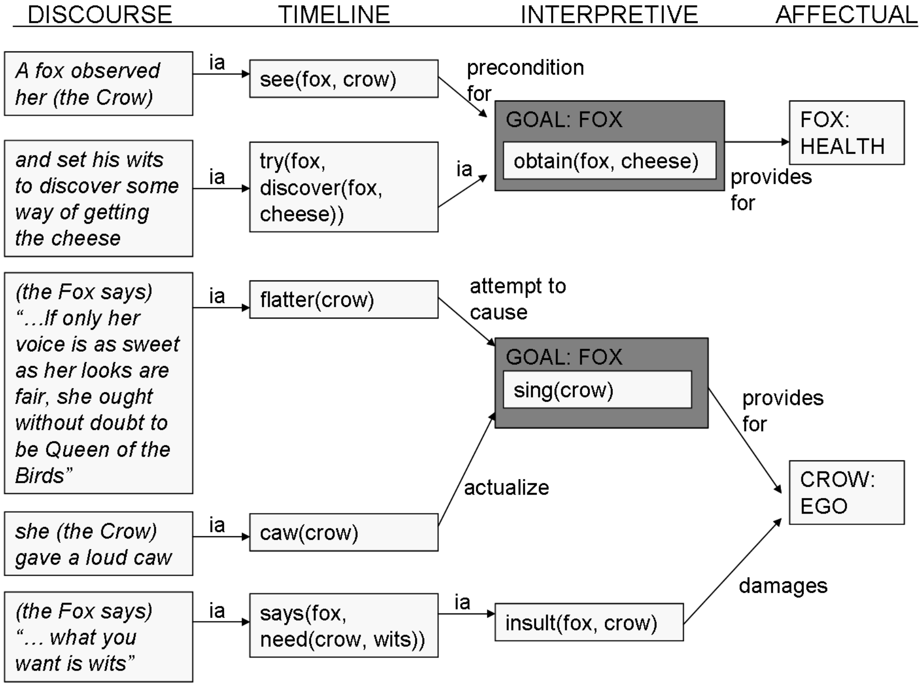

DISCOURSE
TIMELINE
INTERPRETIVE
AFFECTUAL
A fox observed her (the Crow)
ia
see(fox, crow)
precondition for
GOAL: FOX
obtain(fox, cheese)
FOX: HEALTH
provides for
and set his wits to discover some way of getting the cheese
ia
try(fox, discover(fox, cheese))
ia
(the Fox says) "...If only her voice is as sweet as her looks are fair, she ought without doubt to be Queen of the Birds"
ia
flatter(crow)
attempt to cause
GOAL: FOX
sing(crow)
provides for
CROW: EGO
actualize
she (the Crow) gave a loud caw
ia
caw(crow)
damages
(the Fox says) "... what you want is wits"
ia
says(fox, need(crow, wits))
ia
insult(fox, crow)

Figure 3.16: Fox and Crow goals in *The Fox and the Crow*

1. The fox wants the crow to drop her cheese so he can get it. He succeeds and provides for his health.

2. The fox flatters the crow in order for her to sing. He succeeds and provides for the Crow's ego. Then he damages the Crow's ego.

To find the causal chain, we start at an affect state, which are by nature and design, sinks in the SIG network, and work backwards to find a timeline or interpretation node at the start of the character goal: The fox wants the crow to drop her cheese so he can get it. The chain of events in the SIG is "the fox's goal" is "the crow sings" *would cause* "the crow opens her beak" *would cause* "the crow drops the cheese" *would cause* "the fox obtains the cheese from the beak of the crow" *actualizes* "Fox: Health". We start at the affect node "Fox: Health" and work backwards

	The Fox has a plan to boost the crow's ego.
At the beginning, the Crow had some cheese.	At the beginning, the Crow has some cheese.
The Fox made the Crow sing.	The Fox made the crow sing.
The Crow dropped the cheese.	The Crow dropped the cheese.

Table 3.9: "The Fox and the Crow" in different orders

until we find the last event in the goal box. At this point we can realize the goal: "the fox obtains the cheese from the beak of the crow".

Table 3.9 shows two different versions of "The Fox and the Crow". By manipulating the presentation of the story by telling the final goal first, we can aim to affect the reader's perceptions. The difference between these two story tellings is that in the story on the right hand side in Table 3.9, we hint at the Fox's motives. As the story is told, the reader is constantly anticipating the moment when the Fox's plan will be realized and he gets the cheese. Perhaps the reader invests less emotional identification with the Fox in this telling because they already know what will happen. On the other hand, readers may instead root for the fox, waiting for the plan to be revealed. In the telling on the left hand side, readers do not know what will happen to the Fox; at the beginning there are no expectations as to what emotions will be evoked.

We point out that, despite the strength and availability of the relationships found in the SIG and further captured in the EST, the temporal ordering methodology required a deeper understanding than what the EST already provides in order to understand narrative effects, such as suspense. For example, if the goal path from the flattery event is realized, the EST might inadvertently tell the audience "the Fox flatters the crow to attempt to cause his goal of the crow singing to drop the cheese", ruining the effect of surprise.

The EST requires additional modeling that we discuss in future work (Section 6.3.2) along with other works that look in this space. We look at attention models and elaborate more on the methodology and planning methods other works use, and theorize what new mechanisms could be adapted to integrate suspense into the EST.

3.3.3 Non-Narrative Specific Content Planning

In addition to modeling affective states and temporal manipulation, we discuss briefly how non-narrative specific content planning operations, such as those employed by PERSONAGE, can be easily integrated into the framework. In a low verbosity setting, the EST can remove the modifiers from SchLexSynt trees. The sentence *I grabbed the comicbook because the bugs scared me* can become *I grabbed the comicbook* (Point of view introduced in Section 4.1.1). Verbosity makes use of the existing deaggregation and aggregation operations introduced in Section 4.1.5. A different low verbosity transformation could remove adjective or adverbial modifiers by excluding SchAdv or SchAdj nodes.

High verbosity realizations might invoke repetition. The EST reselect a root SchVerb tree and realizes it twice. The benefit of the Fabula Tales sentence planner is that the second realization may differ from the first, making the repetition sound more realistic, for example, *I grabbed the comicbook because the bugs scared me. Oh, I grabbed it!* (Character voice introduced in Section 4.1.3 and 4.1.4).

The EST could also transform an action story point into a direct speech act, e.g. *I chased the bugs* becomes *Get back here! she yelled.* However, this operation may require knowledge accumulated and learned from the generation dictionary, as we explore further in Section 6.1.

id	Fabula Tales baseline	Features
1	The narrator recently momentarily opened the narrator's patio's door.	adverb modifiers, nested possession
2	The slimy bugs quietly entered the narrator's apartment.	group, adjective, possession
3	The narrator did not initially notice for the bugs quietly to enter the narrator's apartment.	negation, nested verb phrase
4	The narrator grabbed the comicbook because the bugs scared the narrator.	"because" text plan creation

Table 3.10: Special features of sentences in *Bug Out For Blood*

3.4 Selected Translation Examples

This section describes the translation from semantic story points in the SIG to EST lexico-syntactic and text plan data structures. We assume that these AssignedPredicates were selected by the content planner and passed through to the transformation process. Table 3.10 shows the actual output of a subset of sentences from *Bug Out For Blood* from our EST translation method using no narrative variations in sentence planning, thus we refer to these sentences as "Fabula Tales baseline". We provide detail for the transformation of these selected sentences, focusing on the key features of the semantics.

Fabula Tales baseline sentence 1: *The narrator recently momentarily opened the narrator's patio's door*

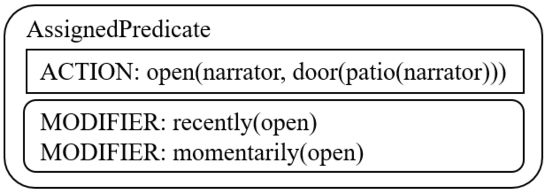
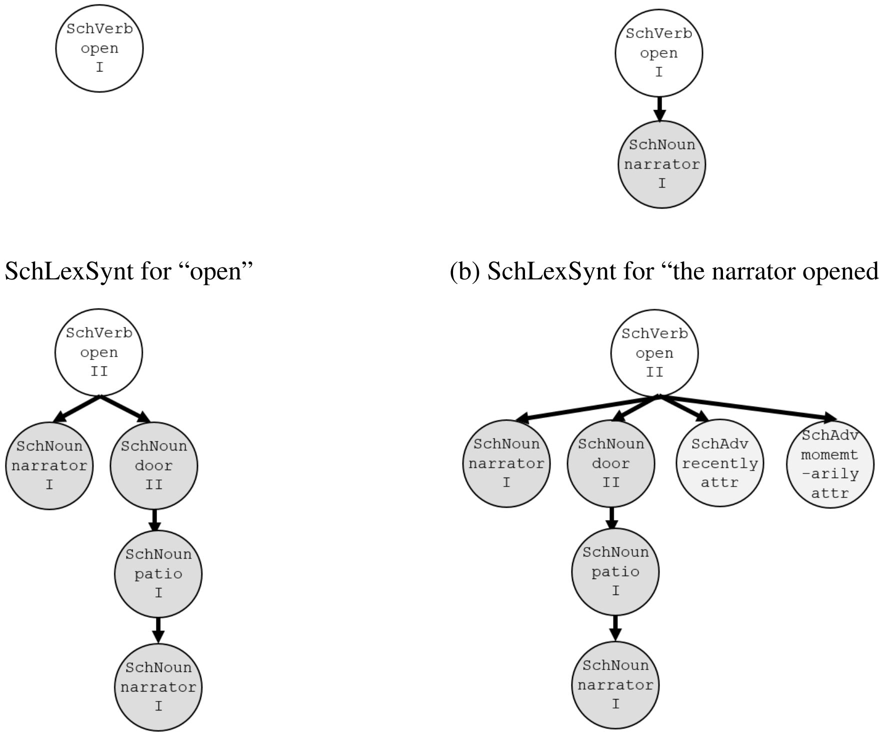

AssignedPredicate
ACTION: open(narrator, door(patio(narrator)))
MODIFIER: recently(open)
MODIFIER: momentarily(open)

Figure 3.17: SIG semantics for sentence 1



(a) SchLexSynt for "open"

(b) SchLexSynt for "the narrator opened"

(c) SchLexSynt for "the narrator opened the narrator's patio's door"

(d) SchLexSynt for "the narrator recently momentarily opened the narrator's patio's door"

Figure 3.18: SchLexSynt for sentence 1

The first sentence we transform is represented in Figure 3.17. To extract the lexical information and structure needed for the SIG transformation to DSYNTS, we retrieve the AssignedPredicate object from the API that represents the root of this sentence. The EST extracts the lexical information from the AssignedPredicate, where the lexeme is "open". A new SchVerb is initialized with "open", denoted in the following discussion as SchVerb<open> (Figure 3.18a). The first argument of the AssignedPredicate has the lexeme "narrator". A new SchNoun is initialized with "narrator" (SchNoun<narrator>) and set as the first child of the SchVerb<open> (Figure 3.18b).

```
<dsyntnode class="verb" lexeme="open" rel="II" tense="past">
 <dsyntnode article="no-art" class="common_noun" gender="fem" lexeme="narrator"
      number="sg" person="" rel="I"/>
  <dsyntnode article="no-art" class="common_noun" gender="neut" lexeme="door"
       number="sg" person="" rel="II">
    <dsyntnode article="no-art" class="common_noun" gender="neut" lexeme="patio"
         number="sg" person="" rel="I">
     <dsyntnode article="no-art" class="common_noun" gender="fem"
          lexeme="narrator" number="sg" person="" pro="poss" rel="attr"/>
    </dsyntnode>
   </dsyntnode>
   <dsyntnode class="adverb" lexeme="recently" position="pre-verbal" rel="ATTR"/>
   <dsyntnode class="adverb" lexeme="momentarily" position="pre-verbal"
        rel="ATTR"/>
 </dsyntnode>
</dsyntnode>
```

Table 3.11: DSYNTS for sentence 1 realized as *The narrator recently momentarily openned the narrator's patio's door*

The second child of the AssignedPredicate for open is a series of nested arguments representing “the door of the patio of the narrator”. Nested SchNouns, each one its own entity, are created: a new SchNoun for door is set as the child of a SchNoun for patio, and finally set as a child of a SchNoun for the narrator (Figure 3.18c).

There are no more children of the AssignedPredicate for open, completing the ACTION box in Figure 3.17. The next step is to assign modifiers to entities in this proposition. A modifier of the adverbial type with the lexeme “recently” is attached to the AssignedPredicate open. A SchAdverb is created (SchAdverb<recently>) and set as the third child of SchVerb<open>. There is a second adverbial modifier for the lexeme ”momentarily”. The SchAdverb creation process is repeated and SchAdverb<momentarily> is appended as the fourth child of SchVerb<open> (Figure 3.18d).

Having completed processing the semantics in Figure 3.17, the XML representation of the SchLexSynt tree features are written. A final post processing step transforms the possessive articles in the nested set of SchNouns representing “the narrator’s patio’s door”. Without this modification, the SchLexSynt would realize as “the door of the patio of the narrator”. The EST opts for a more natural realization. The articles of the nested SchNouns are set to none to enforce possession; if the articles were maintained, the output would produce “the narrator’s the patio’s the door”. The resulting DSYNTS are shown in Table 3.11 and realized as *The narrator recently momentarily opened the narrator’s patio’s door*.

Fabula Tales baseline sentence 2: *The slimy bugs quietly entered the narrator’s apartment.*

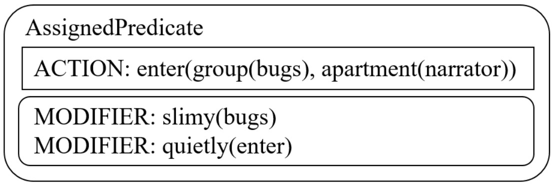
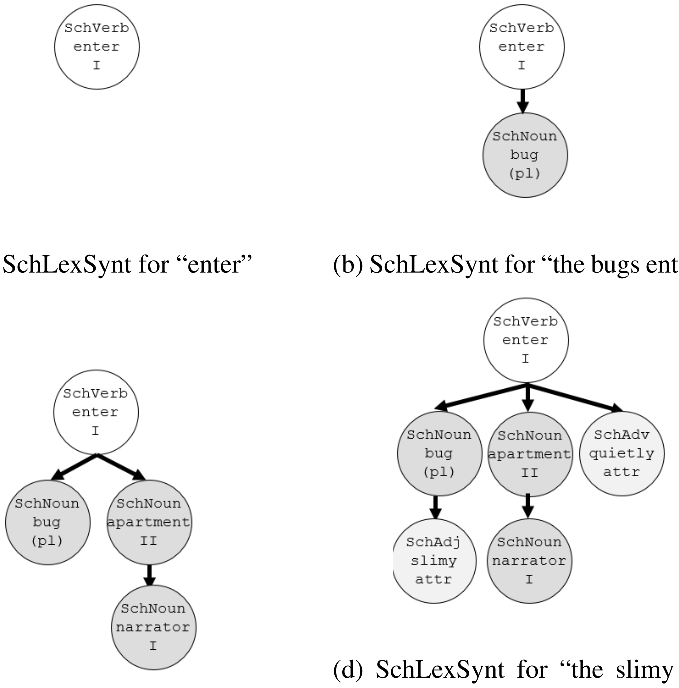


Figure 3.19: SIG semantics for sentence 2

In the semantics in Figure 3.19, we focus the discussion on the plurality of the bugs and the modifier of the bugs. A new SchVerb is initialized with “enter” from the AssignedPredicate object (Figure 3.20a), as in the previous example. The first argument is of a group type. This noun alone does not indicate what the group consists of, so we look at the arguments of the “group” and find another noun for “bug”. This nested noun indicates this is a “group of bugs”, which the EST handles by creating a SchNoun for bugs with the plural attribute set to true and appending this SchNoun<bugs> as the first child of SchVerb<enter> (Figure 3.20b). The second argu-



(a) SchLexSynt for “enter”

(b) SchLexSynt for “the bugs entered”

(c) SchLexSynt for “the bugs entered the narrator’s apartment”

(d) SchLexSynt for “the slimy bugs quietly entered the narrator’s apartment”

Figure 3.20: SchLexSynt for sentence 2

ment of the AssignedPredicate “enter” is a series of nested nouns representing “the narrator’s apartment”, which are handled in the same way as the example in sentence 1 (Figure 3.20c).

After the construction of the ACTION semantics as a SchLexSynt tree, there is a “slimy” modifier associated with the noun “group of bugs”. A SchAdj<slimy> is created and set as a child of SchNoun<bug>. There is also a “quietly” modifier associated with the AssignedPredicate for “enter” which we process into SchAdverb<quietly> and append similarly as in sentence id 1. The final SchLexSynt tree is in Figure 3.20d and Table 3.12 shows the DSYNTS. When the DSYNTS are written to the file, the “number” is set to “pl” (line 2 of Ta-

```
<dsyntnode class="verb" lexeme="enter" mode="" mood="ind" rel="II" tense="past" wn_offset="2016523">
  <dsyntnode article="def" class="common_noun" gender="neut" lexeme="bug" number="pl" person="" rel="I" wn_offset="2236355">
    <dsyntnode class="adjective" lexeme="slimy" rel="ATTR"/>
  </dsyntnode>
  <dsyntnode article="no-art" class="common_noun" gender="neut" lexeme="apartment" number="sg" person="" rel="II" wn_offset="2726305">
    <dsyntnode article="no-art" class="common_noun" gender="fem" lexeme="narrator" number="sg" person="" pro="poss" rel="attr"/>
  </dsyntnode>
  <dsyntnode class="adverb" lexeme="quietly" position="pre-verbal" rel="ATTR"/>
</dsyntnode>
```

Table 3.12: DSYNTS for sentence 2 realized as *The slimy bugs quietly entered the narrator's apartment*

ble 3.12). The final realization produced from the EST DSYNTS is *The slimy bugs quietly entered the narrator's apartment*.

Fabula Tales baseline sentence 3: *The narrator did not initially notice for the bugs quietly to enter the narrator's apartment*

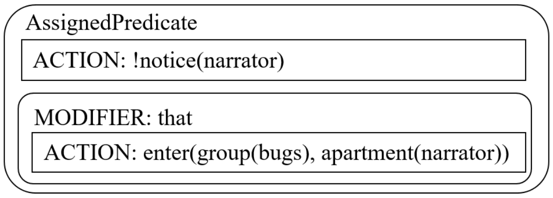
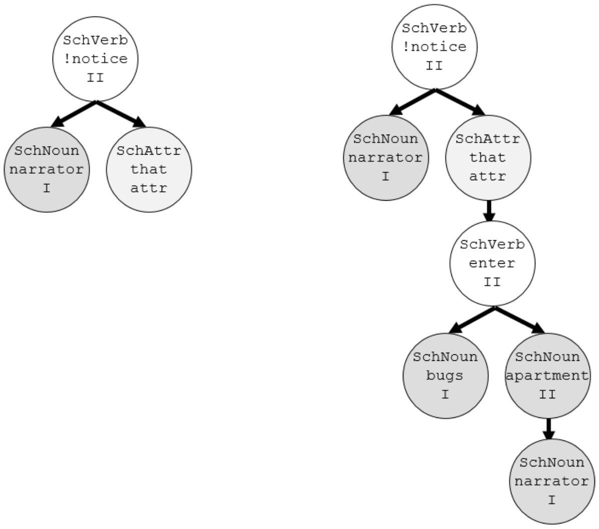

AssignedPredicate
ACTION: !notice(narrator)
MODIFIER: that
ACTION: enter(group(bugs), apartment(narrator))

Figure 3.21: SIG semantics for sentence 3

The discussion points highlighted in the semantics of sentence 3 are the negation and the nested action (Figure 3.21). The AssignedPredicate for "notice" has a negative truth value. The EST accordingly sets the polarity of the new SchVerb to false (denoted as SchVerb<!notice>). The



(a) SchLexSynt for "The narrator did not notice that"

(b) SchLexSynt for "The narrator did not notice that the bugs entered her apartment."

Figure 3.22: SchLexSynt for sentence 3

first argument of this verb we create a SchNoun<narrator>.

The second argument is an AssignedPredicate with "enter". A SchAttr<that> and a SchVerb<enter> are created, and SchVerb<enter> is set to the child of SchAttr<that> (Figure 3.22a). Then, SchAttr<that> is set as a child of SchVerb<!notice>. The "bugs" and the "narrator's apartment" are handled in the same manner as in the previous sentences (Figure 3.22b). When the DSYNTS are written from SchLexSynts, the polarity of SchVerb<notice> is set to "neg", reflected in line 1 in Table 3.22. These DSYNTS produced by the EST are realized as *The narrator did not initially notice that the slimy bugs quietly entered the narrator's apartment.*

```
<dsyntnode class="verb" lexeme="notice" mode="" mood="ind" polarity="neg"
    rel="II" tense="past" wn_offset="2154508">
  <dsyntnode article="no-art" class="common_noun" gender="fem" lexeme="narrator"
      number="sg" person="" pro="pro" rel="I"/>
  <dsyntnode class="" lexeme="that" rel="ATTR">
    <dsyntnode class="verb" lexeme="enter" mode="" mood="ind" rel="II"
        tense="past" wn_offset="2016523">
      <dsyntnode article="def" class="common_noun" gender="neut" lexeme="bug"
          number="pl" person="" rel="I" wn_offset="2236355"/>
        <dsyntnode article="no-art" class="common_noun" gender="neut"
            lexeme="apartment" number="sg" person="" rel="II" wn_offset="2726305">
        <dsyntnode article="no-art" class="common_noun" gender="fem"
            lexeme="narrator" number="sg" person="" pro="poss" rel="attr"/>
      </dsyntnode>
    </dsyntnode>
  </dsyntnode>
</dsyntnode>
```

Table 3.13: DSYNTS for sentence 3 realized as *The narrator did not notice that the bugs entered the narrator's apartment.*

Fabula Tales baseline sentence 4: *The narrator grabbed the comicbook because the bugs scared the narrator.*

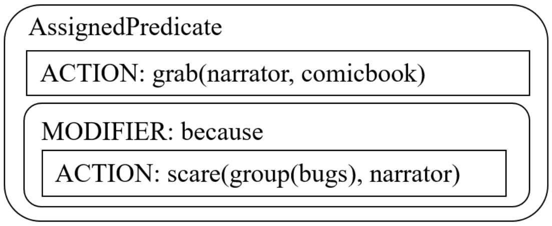
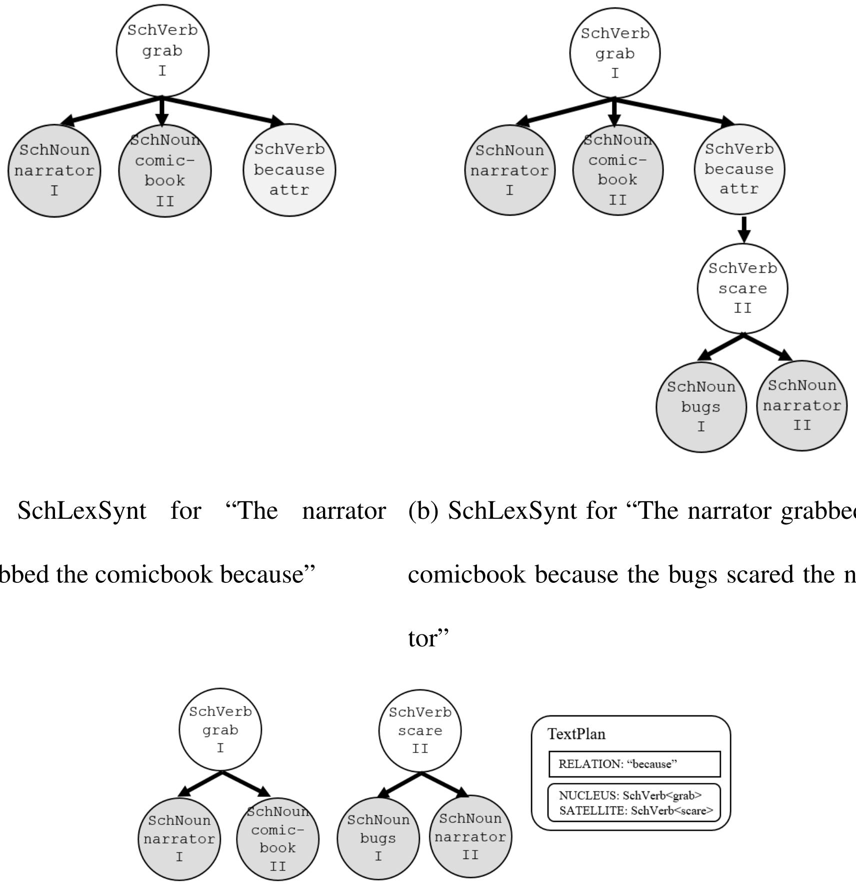

AssignedPredicate
ACTION: grab(narrator, comicbook)
MODIFIER: because
ACTION: scare(group(bugs), narrator)

Figure 3.23: SIG semantics for sentence 4

For the semantics in Figure 3.23, the SchVerb<grab> is created from an AssignedPredicate as usual with its children from argument as SchNoun<narrator> and SchNoun<comicbook>. When looking through modifiers, the EST finds a subordinate clause "because". A SchFunc with



(a) SchLexSynt for “The narrator grabbed the comicbook because”

(b) SchLexSynt for “The narrator grabbed the comicbook because the bugs scared the narrator”

(c) SchLexSynt and text plans for sentence 4

Figure 3.24: SchLexSynt for sentence 4

lexeme “because” is created and appended to the root verb SchVerb<grab> (Figure 3.24a). The first modifier of the subordinate clause is another ACTION for “scare”. A SchVerb<scare> is created appended as the child of SchFunc<because>. Similarly, a SchNoun<bugs> and SchNoun<narrator> are appended to SchVerb<scare> as in previous sentences. The written DSYNTS from the EST are realized as: *The narrator grabbed the comicbook because the bugs scared the narrator.*

```
<dsynts id=1>
 <dsyntnode class="verb" lexeme="grab" rel="II" tense="past">
  <dsyntnode article="def" class="common_noun" lexeme="narrator" rel="I"/>
  <dsyntnode article="def" class="common_noun" lexeme="comicbook" rel="II"/>
 </dsyntnode></dsynts>
<dsynts id=2>
 <dsyntnode class="verb" lexeme="scare" mode="" tense="past">
  <dsyntnode article="def" class="common_noun" lexeme="bug" number="pl" rel="I"/>
  <dsyntnode article="def" class="common_noun" lexeme="narrator" rel="II"/>
 </dsyntnode></dsynts>
<textplan><rstplan>
    <relation name="because">
      <proposition id="1" ns="nucleus"/>
      <proposition id="2" ns="satellite"/>
    </relation></rstplan>
  <proposition dialogue_act="1" id="1"/>
  <proposition dialogue_act="2" id="2"/>
</textplan>
```

Table 3.14: DSYNTS for sentence 4 realized as *The narrator grabbed the comicbook* (1), *The bugs scared the narrator.* (2), and the text plan

The EST identifies a text plan in the semantics between “the narrator grabbed the comicbook” and “the bugs scared the narrator”, joined by a “because” relationship. The EST preserves the original DSYNTS and in addition, creates two new DSYNTS, splitting apart the content (Figure 3.24c). It assigns the nucleus and satellite to the first and second clause respectively, and creates a text plan relationship output in Figure 3.14.

3.5 Summary

The EST creates a bridge across the NLG gap that utilizes the strengths of both the SIG modeling and DSYNTS structure, overcoming the weaknesses if one were to harness each framework independently in an NLG capacity. The EST offers a domain independent solution to NLG, as opposed to the domain specificity found in Callaway's system and others introduced in Chapter 2, that were only able to generate variations, for instance, limited to the Little Red Riding Hood domain. The modeling strength of the EST is that any story that can be modeled as a SIG can be translated. Furthermore, the EST does not require manual DSYNTS creation or a formal specification of input, as required in the PERSONAGE, SpyGen, or Curveship systems. The EST's automatic derivation of DSYNTS alleviates the time and skill burden in content creation.

The EST is a also content planner. We explore narrative content planning operations, including appraisal modeling, temporal selection, and non-narrative specific parameters associated with voice models in PERSONAGE to add diversity to the generated structure. The EST further provides semantic interpretability, unlike some data-driven approaches, using text plans. The EST's transformation of DSYNTS and text plans positions the framework to perform sentence planning with Fabula Tales from the rich content preserved by the EST (Chapter 4), that previous works, including SIG's WYSIWYM were unable to perform.

Chapter 4

Fabula Tales and Narratological Sentence Planning

This Chapter introduces the narrative sentence planner, Fabula Tales, into our larger NLG storytelling framework. The EST, as a bridge and content planner, preserves and converts semantic content into a parameterizable syntactic representation. The modeling of the SIG through the EST allows for changing point of view and inserting and representing direct speech, as well as the exploration of the following sentence planning parameters in the narrative scope: lexical choice, pragmatic marker insertion, and aggregation and deaggregation.

Fabula Tales introduces the capacity to retell any story from the perspective of any character in it, such as the narrator seen in A1 in Table 4.1, (*point of view*, Section 4.1.1). Section 4.1.2 shows how Fabula Tales can transform an indirect speech act (B1 in Table 4.1) into a *direct speech act* (B2), giving the characters the opportunity to speak in their own voice. This motivates designing Fabula Tales to apply a voice model to utterances (C1 in Table 4.1), utilizing

ID	Narrative Parameter	Example
A1	Point of View	The protesters started to protest at the Capitol Building. The protesters moved toward the downtown. I stood at Stout and 16th.
B1	Indirect Speech	Anne said she didn't receive the new schedule
B2	Direct Speech	"Yeah, well, I didn't receive the new schedule!", Anne said.
C1	Character Voice (pragmatic insertion, lexical choice)	I stood at the classroom's front. I no-noticed my ankle to be somewhat observed. I glanced around at the pupils.
D1	Deaggregation	I placed the bowl on the deck because Benjamin wanted to drink the bowl's water.
D2	Deaggregation	I placed the bowl on the deck. Benjamin wanted to drink the bowl's water.

Table 4.1: Narratological Sentence Planning Variations in Blogs

pragmatic marker insertion and *lexical choice* (Section 4.1.3 and 4.1.4). Finally, Fabula Tales utilizes the text plans derived in the EST to phrase the same utterance in different ways through *deaggregation* (D1 and D2 in Table 4.1, discussed in Section 4.1.5).

Fabula Tales parameters can be set by hand, or models with predefined parameters designed to communicate a certain narration style. For example, we might have a story told in the first person, without direct speech, instead communicating the narrator's voice throughout the entire narrative. Instead, another telling may be told in the third person from a neutral narrator perspective but contain direct speech acts of different characters in the story. Both tellings can employ different sentence planning aggregations.

We require a pool of content of different types of stories to test the storytelling and retelling framework. Personal narratives are of particular interest because they illustrate many different styles and varieties of topics familiar in our daily lives. Section 4.2 describes how new content is created as input to our framework through the Scheherazade annotation tool, and how

ID	Story	Example
A2	Startled Squirrel	I approached the bowl. I was startled because I saw my reflection. I leaped because I was startled. I fell over the deck's railing because I leaped because I was startled. I held the deck's railing with my paw. My paw slipped off the deck's railing. I fell.
A3	Botches Training	I stood at the classroom's front. I noticed my ankle to be somewhat observed. I looked nervously toward my ankle. I glanced around the students.

Table 4.2: Point of View Variations in Blogs

annotators were trained to adapt the story annotation process to create the large corpus of SIG personal narratives, PersonaBank.

4.1 Narratological Sentence Planning

4.1.1 Point of View

Biber [1991] claims that first person pronouns are markers of ego-involvement with a text. The subjects of cognitive verbs are usually first person pronouns, indicating that discussion of mental processes is a personal matter often associated with high ego-involvement" compared to the third person pronouns, which are outside of the immediate interaction [Biber, 1991]. Second person pronouns require a specific addressee and indicate a high degree of involvement with that addressee. They have been used as a marker of register differences. Third person personal pronouns mark relatively inexact reference to persons outside of the immediate interaction. Biber finds that third person pronouns co-occur frequently with past tense and perfect aspect forms, as a marker of narrative reported (versus immediate) styles [Biber, 1991].

According to Pizarro et al. [2003], different perspectives restrict narrative informa-

```
<dsyntnode class="verb" lexeme="leap" rel="II" tense="past">
 <dsyntnode article="def" class="common_noun" gender="neut" lexeme="squirrel"
      number="sg" person="" property_0="crazy" rel="I">
  <dsyntnode class="adjective" lexeme="crazy" rel="ATTR"/>
 </dsyntnode>
 <dsyntnode class="preposition" lexeme="because" rel="ATTR">
  <dsyntnode class="verb" lexeme="be" rel="II" tense="past">
  <dsyntnode class="adjective" lexeme="startled" rel="II"/>
  <dsyntnode article="def" class="common_noun" gender="neut" lexeme="squirrel"
       number="sg" person="" rel="I"/>
  </dsyntnode>
 </dsyntnode>
</dsyntnode>
```

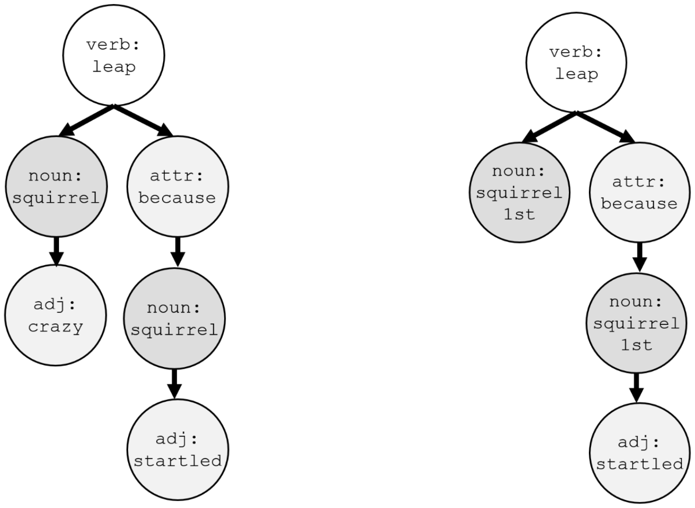
Table 4.3: DSYNTS for *The crazy squirrel leapt because it was startled*

```
<dsyntnode class="verb" lexeme="leap" rel="II" tense="past">
 <dsyntnode article="def" class="common_noun" gender="neut" lexeme="squirrel"
      number="sg" person="1st" property_0="crazy" rel="I"/>
 <dsyntnode class="preposition" lexeme="because" rel="ATTR">
  <dsyntnode class="verb" lexeme="be" rel="II" tense="past">
   <dsyntnode class="adjective" lexeme="startled" rel="II"/>
  <dsyntnode article="def" class="common_noun" gender="neut" lexeme="squirrel"
       number="sg" person="1st" rel="I"/>
  </dsyntnode>
 </dsyntnode>
</dsyntnode>
```

Table 4.4: DSYNTS for *I leapt because it was startled*

tion to the eye of a character, allowing the audience limited perception for a particular effect. We hypothesize H_1: there is a significant correlation between first person point of view and engagement because it brings the reader into the immediacy of a story.

Fabula Tales can change the point of view of any character in the story, including non-narrating characters such as the squirrel in *Startled Squirrel* (Table 4.2). Point of view is altered after performing the EST translation methodology by iterating through each DSYNTS and



(a) DSYNTS tree for "The crazy squirrel leapt because it was startled"

(b) DSYNTS tree for "I leapt because I was startled"

Figure 4.1: DSYNTS trees for *The squirrel leapt because it was startled*

making a change to its attributes, rather than performing the change on the realized surface form. Table 4.3 shows the DSYNTS for the sentence *The crazy squirrel leapt because it was startled* and the graphical tree representation is in Figure 4.1a with key attributes displayed. In order to transform a sentence into the first person, minor changes to the deep structure are necessary. At lines 2 and 8 in Table 4.3, we assign the `person` attribute to `1st` to specify a change of point of view to first person, reflected at lines 2 and 6 in Table 4.4 and the squirrel nodes in Figure 4.1b. The surface realizer interprets the `person` attribute and automatically changes the lexeme present at line 2 to *I*, and handles the coreference resolution at line 8 to *myself* [Lukin et al., 2015]. This is a major advantage of our computational framework: the deep linguistic representation allows us to specify changes we want without manipulating strings, and allows general rules for the point of view parameter.

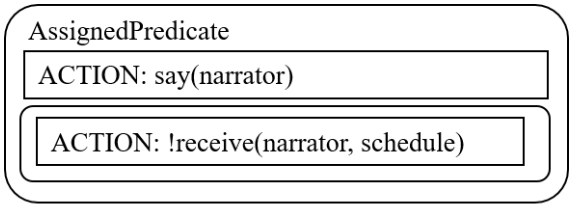


Figure 4.2: SIG representation for *The narrator said she didn't receive the schedule*

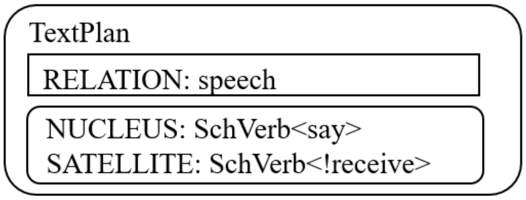


Figure 4.3: Text plan for speech relation

Despite this advantage, the surface realizer does not automatically handle the descriptive properties; changing the third person perspective of the sentence, "The crazy squirrel leapt" to first person without any additional modifications to the DSYNTS structure, it produces "Crazy I leapt". The property must be removed in the DSYNTS structure and is thus realized as "I leapt". Table 4.4 and Figure 4.1b reflect the removal of this adjective.

4.1.2 Direct Speech

When people tell stories, they know what their characters are feeling, and can express it in their telling. In order to reveal the depth of the characters and relate to them, we need insight into their personality and emotions via direct speech acts. Bal claims that "dialogue is a form in which the actors themselves, and not the primary narrator, utter language" [Bal, 1997] and that "the more dialogue a narrative text contains, the more dramatic it is". Thus, we hypothesize H_2 which claims there is a significant correlation between direct speech and

Original Content	Anne said she didn't receive the new schedule.
Relations	SPEECH (nuc:1, sat:2)
Content	1: say(Anne) 2: !receive(Anne, new schedule)
Indirect	Anne said she didn't receive the new schedule.
Direct	"I didn't receive the new schedule!" Anne said.

Table 4.5: A content plan for contingency and speech

engagement because direct speech allows characters to express themselves.

Speech acts in the SIG are encoded as indirect speech; it is not possible to encode direct speech. We use the WordNet index provided from the SIG annotation to identify if the main verb is a verb of communication. If so, we break apart the DSYNTS into the utterance to be uttered, and the explanatory phrase, or nucleus and satellite respectively, based on Rhetorical Structure Theory (RST) [Mann and Thompson, 1988] and create a text plan between propositions (Figure 4.3). In the sentence *The narrator said she didn't receive the new schedule* (Figure 4.2), Fabula Tales iterates through the SchLexSynt objects representing this sentence, looking for SchVerb<say>. Once found, Fabula Tales next identifies SchNoun<Anne> as its subject, and finds no object. The remainder of the tree starting from SchVerb<receive> as the root verb, which is what is to be uttered, is split it off from its parent verb of communication (SchVerb<say>), resulting in two distinct trees (1 and 2 in Table 4.5) [Lukin et al., 2015].

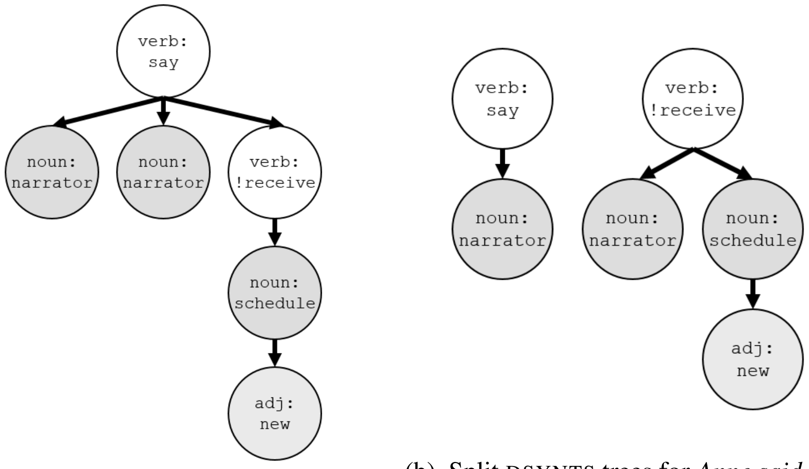
Table 4.6 and Figure 4.4a show the original, unsplit DSYNTS for the example in Table 4.5. After splitting, each tree is treated as a unique DSYNT, as seen in Table 4.7 and Figure 4.4b, which show two smaller DSYNTS. A speech text plan is constructed consisting of the two DSYNTS (Table 4.7 at line 12), which are combined by the surface realizer and realized in direct speech as *"I didn't receive the new schedule" Anne said.*

```
<dsyntnode class="verb" lexeme="say" rel="II" tense="past">
 <dsyntnode article="def" class="common_noun" gender="fem" lexeme="Anne"
     number="sg" person="" rel="I"/>
 <dsyntnode class="verb" lexeme="receive" polarity="neg" rel="III" tense="past">
  <dsyntnode article="def" class="common_noun" gender="fem" lexeme="Anne"
      number="sg" person="" rel="I"/>
  <dsyntnode article="no-art" class="proper_noun" gender="neut" lexeme="New
      schedule" number="sg" person="" rel="II"/>
 </dsyntnode>
</dsyntnode>
```

Table 4.6: DSYNTS for *Anne said she didn't receive the new schedule*

```
<dsynts id=1>
<dsyntnode class="verb" lexeme="say" rel="II" tense="past">
 <dsyntnode article="def" class="common_noun" gender="fem" lexeme="Anne"
     number="sg" person="1st" rel="I"/>
 </dsyntnode>
</dsynts>
<dsynts id=2>
 <dsyntnode class="verb" lexeme="receive" polarity="neg" rel="III" tense="past">
  <dsyntnode article="def" class="common_noun" gender="fem" lexeme="Anne"
      number="sg" person="" rel="I"/>
  <dsyntnode article="no-art" class="proper_noun" gender="neut" lexeme="New
      schedule" number="sg" person="" rel="II"/>
 </dsyntnode>
</dsynts>
<speechplan voice="narrator">
 <rstplan>
  <relation name="speech">
   <proposition id="1" ns="nucleus"/>
   <proposition id="2" ns="satellite"/>
  </relation>
 </rstplan>
 <proposition dialogue_act="1" id="1"/>
 <proposition dialogue_act="2" id="2"/>
</speechplan>
```

Table 4.7: Split DSYNTS and text plan for *Anne said "I didn't receive the new schedule"*



(a) DSYNTS tree for *Anne said she didn't receive the new schedule*

(b) Split DSYNTS trees for *Anne said* and *Anne didn't receive the new schedule*.

Figure 4.4: DSYNTS trees for direct and indirect speech

4.1.3 Pragmatic Markers

Direct speech allows the characters to speak in their own tone or style. Biber gives support for emotive words (*Oh!*), conative (*please*), modal and uncertainty (*perhaps*) as being indicators of personal stories, whereas these items are lacking in impersonal stories [Biber, 1991]. Many of these emotive, modal, and uncertainty words fall under the category of pragmatic markers. With direct speech able to separate the explanatory from what is uttered, Fabula Tales can generate a broader range of character styles supported by PERSONAGE parameters. Pragmatic marker insertion is utilized in Fabula Tales by implementing PyPersonage [Bowden et al., 2016].

This parameterizable voice model makes it easy to replace one voice model for another. In combination with changes to the point of view, direct speech, and voice, Fabula Tales

ID	Example
C2	The fox looked toward the crow. The fox said "Your beauty is quite incomparable, okay?" The crow thought "The fox was so-somewhat flattering."
C2	The fox said "Your feather's chromaticity is damn exquisite."
C3	The fox said "Your feather's chromaticity is so-somewhat exquisite."
C4	The fox said "Your feather's hue is exquisite."
C5	Yeah, I stood at the classroom's front. I noticed my ankle was damn observed! Oh God I glanced around the students.
C6	I stood at the classroom's front. I no-noticed my ankle was somewhat observed. I looked nervously toward my ankle. I glanced around the students.
C7	I stood at the classroom's front. I noticed my ankle was observed. I nervously looked toward my ankle. I glanced around the students.

Table 4.8: Voice variations in blogs and fables

has the capacity to express the narrators as storytellers and the characters in varied styles. We hypothesize that the implementation of character voices will yield a significant correlation between direct speech and certain styles and increased reader engagement because style features can be used in direct speech to act as character voice (H_3).

"The Fox and the Crow" fable has two characters and the narrator. PERSONAGE's three personality models in combination with Fabula Tales can assign different voices to the characters. For example, C2 in Table 4.8 uses the *laid-back* model for the fox's direct speech, the *shy* model for the crow's direct speech, and the *neutral* model for the narrator voice. The *laid-back* model uses emphasizers, hedges, exclamations, and expletives, whereas the *shy* model uses softener hedges, stuttering, and filled pauses. The *neutral* model is the simplest model that does not utilize any of the extremes of the PERSONAGE parameters. The models can be interchanged within Fabula Tales, to substitute each model for the fox's voice (C3 as outgoing, C4 as shy, and C5 as neutral in Table 4.8). Direct speech is not a prerequisite for adding voices.

```
1  <dsyntnode class="verb" lexeme="smear" rel="II" tense="past" wn_offset="846509">
2   <dsyntnode article="def" class="common_noun" gender="fem" lexeme="narrator"
        number="sg" person=""rel="I" wn_offset="10345804"/>
3   <dsyntnode article="def" class="common_noun" gender="neut" lexeme="viscera"
        number="sg" person="" rel="II" wn_offset="5298988">
4    <dsyntnode article="no-art" class="common_noun" gender="neut" lexeme="bug"
         number="pl" person="" rel="I" wn_offset="2236355"/>
5   </dsyntnode>
6   <dsyntnode class="preposition" lexeme="with" rel="ATTR">
7    <dsyntnode article="def" class="common_noun" gender="neut" lexeme="comic book"
         number="sg" person="" rel="II" wn_offset="6596364">
8     <dsyntnode class="adjective" lexeme="rolled" rel="ATTR" wn_offset="3469"/>
9    </dsyntnode>
10  </dsyntnode>
11  </dsyntnode>
```

Table 4.9: DSYNTS for *I smeared the bug's viscera with the rolled comicbook*

We can change the narrator to the first person and have the narrator's voice change as in C6, C7, and C8 in Table 4.8 as laid-back, shy, and neutral respectively.

4.1.4 Lexical Choice

Voice characteristics can additionally be altered by manipulating lexical choice. Access to WordNet and VerbNet senses allow for DSYNTS to be procedurally manipulated in synonym substitutions. Word senses are preserved from the SIG annotation by the EST, and PyPersonage facilitates the implementation of lexical choice in Fabula Tales. A simple lexical substitution in the sentence *I smeared the bug's innards with the rolled comicbook* swaps "innards" with "viscera". Table 4.9 shows the original DSYNTS with WordNet sense lookup information for each verb, noun, and adjective, enabling Fabula Tales to find synonyms of "innards" (see WN_OFFSET="5298988" in line 3) to find the "viscera".

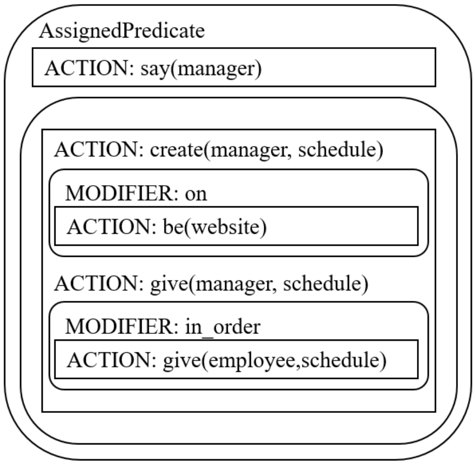

AssignedPredicate
ACTION: say(manager)
ACTION: create(manager, schedule)
MODIFIER: on
ACTION: be(website)
ACTION: give(manager, schedule)
MODIFIER: in_order
ACTION: give(employee,schedule)

Figure 4.5: SIG semantics for a nested AssignedPredicate

Verb substitutions can be performed as well, but we must ensure the synonyms and their arguments are interchangeable. For example, *I managed to squash the bug* is transformed to its argument equivalent *I managed to crush the bug*.

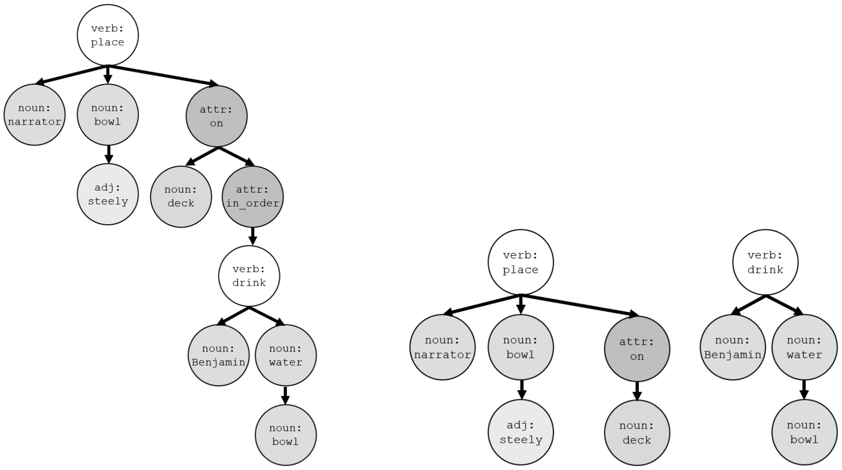
4.1.5 Deaggregation

We observe that some SIG AssignedPredicates contain long sentences not handled well as natural text because the WYSIWYM realizer's primary concern is preserving the meaning of the annotation. The SIG semantics in the nested AssignedPredicate in Figure 4.5 are realized by WYSIWYM as:

> The manager said that she had created a second document that wasn't on a website, and had given the second document to an employee in order for an employee to give the second document to the narrator.

The EST translates this AssignedPredicate as following:

> The manager said she created the new schedule and the manager gave the new schedule to the employee in order for the employee to give the new schedule to Anne.



(a) DSYNTS tree for *The narrator placed the steely bowl on the deck in order for Benjamin to drink the bowl's water.*

(b) Split DSYNTS trees for *The narrator placed the steely bowl on the deck* and *Benjamin drinks the bowl's water*

Figure 4.6: DSYNTS trees for deaggregation

In order to support more fluid retellings, Fabula Tales introduce support for new discourse relations by "de-aggregating" these aggregating clauses in order to have the flexibility to rephrase, restructure, or ignore clauses. Specifically, we focus on aggregating clauses related by the contingency discourse relation (one of many listed in the Penn Discourse Tree Bank (PDTB) [Miltsakaki et al., 2004]). We hypothesize that there is a significant correlation between deaggregation and engagement because aggregation creates more natural sounding and concise stories (H_4).

Similar to the creation of direct speech, Fabula Tales performs deaggregation and aggregation after SchLexSynts are created by the EST. In SIG encoding, contingency clauses are always expressed with the "in order to" relation. Candidate story points that contain a contingency relation are identified and the relationship is deliberately broken apart to create

Original Content	A narrator placed a steely and large bowl of water outside on the back deck **in order for** a dog to drink the water of the bowl.
Relations Content	CONTINGENCY (nuc:1, sat:2) 1: put(narrator, bowl, deck) 2: dog(drink, bowl)
Aggregation	
EST	I placed the bowl on the deck in order for Benjamin to drink the bowl's water.
becauseNS	I placed the bowl on the deck because Benjamin wanted to drink the bowl's water.
becauseSN	Because Benjamin wanted to drink the bowl's water, I placed the bowl on the deck.
NS	I placed the bowl on the deck. Benjamin wanted to drink the bowl's water.
N	I placed the bowl on the deck.
soSN	Benjamin wanted to drink the bowl's water, so I placed the bowl on the deck.

Table 4.10: A content plan and realizations for contingency

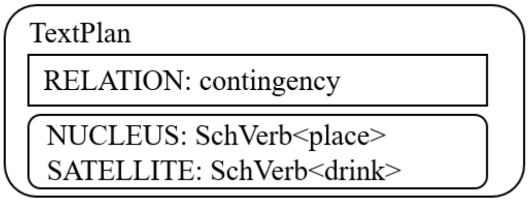


Figure 4.7: Text plan for contingency

nucleus and satellite DSYNTS that represents the entire sentence.

The unsplit DSYNTS for the sentence *The narrator put the bowl on the deck in order for Benjamin to drink the bowl's water* are shown in Table 4.11 and Figure 4.6a. By identifying the "in order to" clause in the SchLexSynt, we break the tree into two distinct trees in Table 4.12 and Figure 4.6b and create a text plan, line 25 in Table 4.12 to identify the nucleus and satellite.

Table 4.10 shows new sentence planning variations for the contingency relation. The *becauseNS* operation presents the nucleus first, followed by a *because*, and then the satellite, and similarly *becauseSN* swaps the order of the phrases. We can also treat the nucleus and satellite as two different sentences (*NS*) or completely leave off the satellite (*N*) if the satellite

```
<dsyntnode class="verb" lexeme="place" rel="II" tense="past">
 <dsyntnode article="def" class="common_noun" lexeme="narrator" rel="I"/>
 <dsyntnode article="def" class="common_noun" lexeme="bowl" rel="II">
  <dsyntnode class="adjective" lexeme="steely" rel="ATTR"/></dsyntnode>
 <dsyntnode class="preposition" lexeme="on" rel="ATTR">
  <dsyntnode article="def" class="common_noun" lexeme="deck" rel="II"/></dsyntnode>
 <dsyntnode class="preposition" lexeme="in_order" rel="ATTR">
  <dsyntnode class="verb" extrapo="+" lexeme="drink" mode="inf-to" mood="inf-to"
       rel="II" tense="inf-to">
   <dsyntnode article="no-art" class="proper_noun" lexeme="Benjamin" rel="I"/>
   <dsyntnode article="def" class="common_noun" lexeme="water" rel="II">
    <dsyntnode article="no-art" class="common_noun" lexeme="bowl" rel="I"/>
   </dsyntnode></dsyntnode></dsyntnode></dsyntnode>
```

Table 4.11: DSYNTS realized as *The narrator placed the steely bowl on the deck in order for Benjamin to drink the bowl's water.*

can be easily inferred from the prior context. The richness of the discourse information present in EST's preservation, and Fabula Tales' ability to deaggregate and then aggregate enables our framework to implement other discourse relations in future work.

4.2 A New Corpus to Explore Variations

The Scheherazade annotation tool allows humans to derive a *fabula* from a *sujet* [Elson and McKeown, 2009]. The annotation process involves sequentially labeling the original story sentences according to the SIG formalism using hooks to WordNet and VerbNet. Human annotators perform this process because there are no tools to automatically label a story to the degree of depth as we described in Section 3.1.1 when we first introduced the SIG.

Works like Callaway that carefully curate a single story can be explored in a variety of different ways [Callaway and Lester, 2002], whereas Scheherazade annotators and the EST-

```
<dsynts id="1">
<dsyntnode class="verb" lexeme="place" rel="II" tense="past">
 <dsyntnode article="def" class="common_noun" lexeme="narrator" rel="I"/>
 <dsyntnode article="def" class="common_noun" lexeme="bowl" rel="II">
  <dsyntnode class="adjective" lexeme="steely" rel="ATTR"/></dsyntnode>
 <dsyntnode class="preposition" lexeme="on" rel="ATTR">
  <dsyntnode article="def" class="common_noun" lexeme="deck" rel="II"/>
  </dsyntnode></dsyntnode></dsyntnode></dsynts>
<dsynts id="2">
<dsyntnode class="verb" lexeme="want" rel="II" tense="past">
 <dsyntnode class="verb" extrapo="+" lexeme="drink" rel="II" tense="inf-to">
  <dsyntnode article="no-art" class="proper_noun" lexeme="Benjamin" rel="I"/>
  <dsyntnode article="def" class="common_noun" lexeme="water" rel="II">
   <dsyntnode article="no-art" class="common_noun" lexeme="bowl" rel="I"/>
  </dsyntnode></dsyntnode></dsyntnode></dsyntnode></dsynts>
<speechplan><rstplan><relation name="contingency_cause">
   <proposition id="1" ns="nucleus"/>
   <proposition id="2" ns="satellite"/>
 </relation></rstplan>
 <proposition dialogue_act="1" id="1"/>
 <proposition dialogue_act="2" id="2"/>
</speechplan>
```

Table 4.12: DSYNTS realized as *The narrator placed the steely bowl on the deck* (1), *Benjamin wanted to drink the bowl's water* (2), and the text plan

Fabula Tales pipeline allows us to apply variations to any story we can annotate. Elson [2012b] collected a corpus of 110 stories annotated as SIGs called the DramaBank. The texts range from Aesop's fables, such as "The Fox and the Crow" and "The Wily Lion", to contemporary fiction, including "Beowulf" and "Gift of the Magi". In our initial development of the EST, we used these fables. However, we found that these stories are not persistent in our everyday lives. This motivated us to collect and create a diverse corpus that showcases different types and styles. We use personal narratives because they are abundant; millions are created daily about countless different topics. Most have a simple timeline and are told in chronological order. Using these criteria, we created the PersonaBank[1] collection of SIG annotated stories with several purposes in mind that will make the corpus useful to other researchers interested in everyday storytelling, narrative modeling, language generation, and language processing [Lukin et al., 2016].

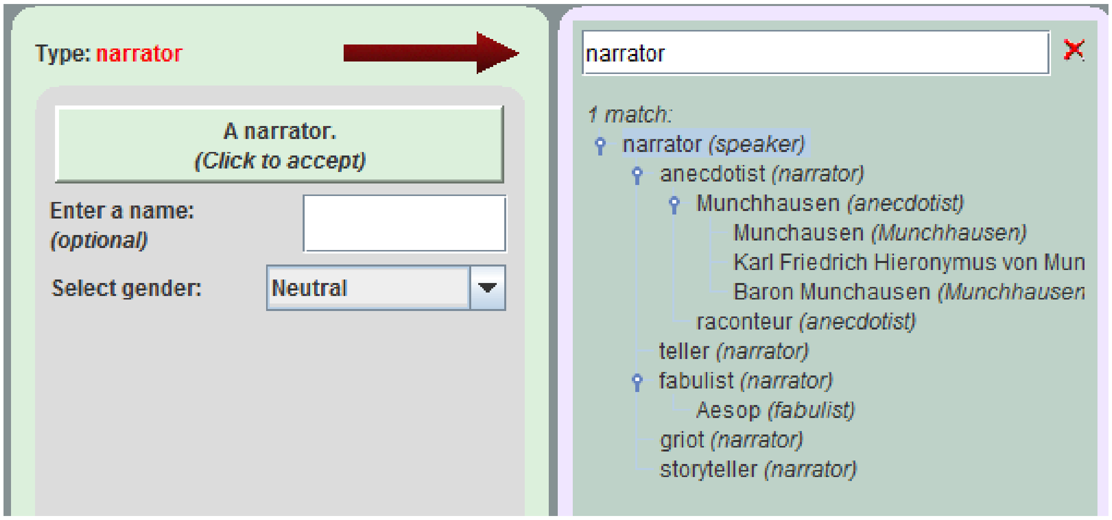
4.2.1 Story Annotation with Scheherazade

The annotation process starts by first defining characters and objects as props for the story, and then assigning actions and properties to them. Scheherazade uses the predicate-argument structures from the VerbNet lexical database and uses WordNet as its noun and adjectives taxonomy where the narrator is a known lexical type that the newly created character "a narrator" is an instance of. All instances of defining characters, objects, traits, or actions in Scheherazade are linked to WordNet and VerbNet.

Let us use for an example the *Protest Story* in Table 3.2. Figure 4.8 illustrates the start of the annotation process where the character "a narrator" is first created. The annotator types "narrator" and is returned a hierarchical list from WordNet of related words. For each choice, the "genus" (category) is shown in italics. We select the closest one: "narrator (speaker)" After selecting a type, the sub-panel on the left will show the slots that need filling for that type, as well as an accept button on top that allows us to complete the definition. Annotators define all characters and props in a similar manner. Figure 4.9a and 4.9b shows the characters and props that annotator selected in the Protest Story, *leaders*, *protestors*, and *police* are characters, and (*tear gas* and *police cars*) are props.

Next, the GUI displays the story that is to be annotated in the Scheherazade GUI

[1] http://nlds.soe.ucsc.edu/personabank



Type: narrator

A narrator.
(Click to accept)

Enter a name: (optional)

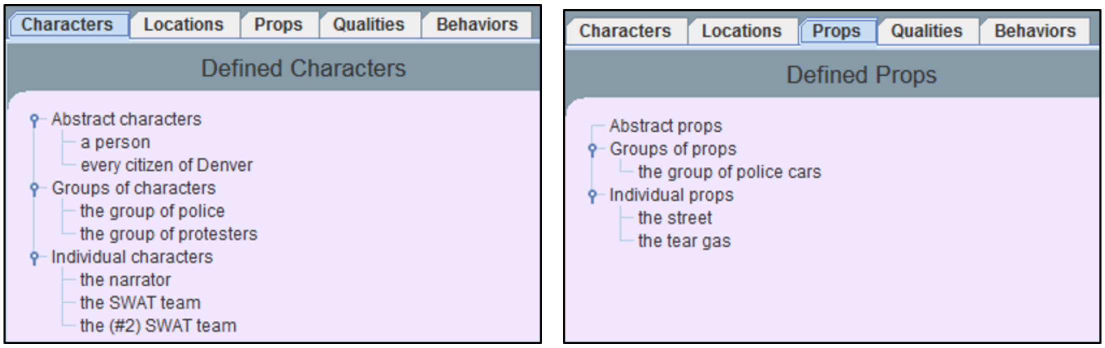
Select gender: Neutral

narrator

1 match:

- narrator (speaker)
  - anecdotist (narrator)
    - Munchhausen (anecdotist)
      - Munchausen (Munchhausen)
      - Karl Friedrich Hieronymus von Mun
      - Baron Munchausen (Munchhausen
    - raconteur (anecdotist)
  - teller (narrator)
  - fabulist (narrator)
    - Aesop (fabulist)
  - griot (narrator)
  - storyteller (narrator)

Figure 4.8: Character creation using Scheherazade in the *Protest Story*



Defined Characters

- Abstract characters
  - a person
  - every citizen of Denver
- Groups of characters
  - the group of police
  - the group of protesters
- Individual characters
  - the narrator
  - the SWAT team
  - the (#2) SWAT team

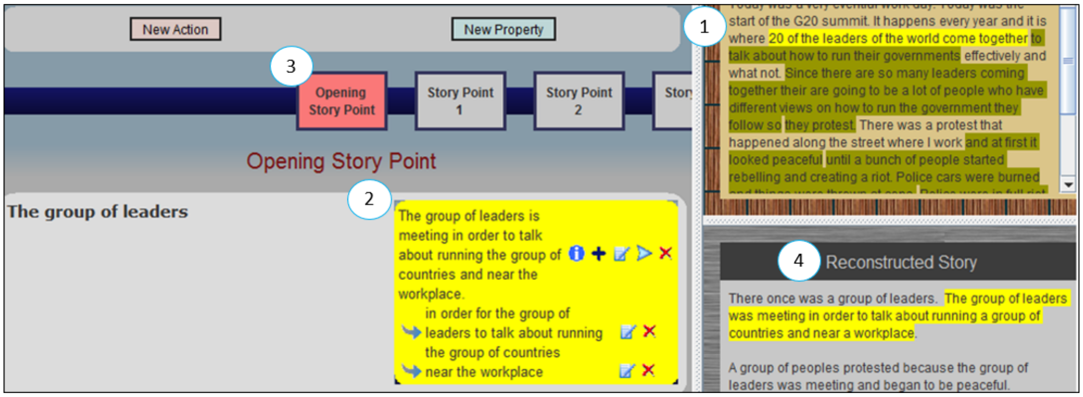
Defined Props

- Abstract props
- Groups of props
  - the group of police cars
- Individual props
  - the street
  - the tear gas

(a) Characters defined in the *Protest Story*

(b) Props defined in the *Protest Story*

(Figure 4.10 #1). The annotator highlights segments of text and creates story points for the selected segment. Figure 4.10 #2 shows the current story point; the user highlights a segment in the original story and encodes it in propositional structures where nodes correspond to lexical items that are linked by thematic relations, in a similar manner to that required for defining characters. Figure 4.11 shows a similar interface for defining actions. VerbNet returns frames related to the search word typed. For example, “something meets” has a “agent verb” frame, while “something meets something” has an “agent verb patient” frame. After selecting the



start of the G20 summit. It happens every year and it is where 20 of the leaders of the world come together to talk about how to run their governments effectively and what not. Since there are so many leaders coming together their are going to be a lot of people who have different views on how to run the government they follow so they protest. There was a protest that happened along the street where I work and at first it looked peaceful until a bunch of people started rebelling and creating a riot. Police cars were burned

The group of leaders is meeting in order to talk about running the group of countries and near the workplace.

in order for the group of leaders to talk about running the group of countries

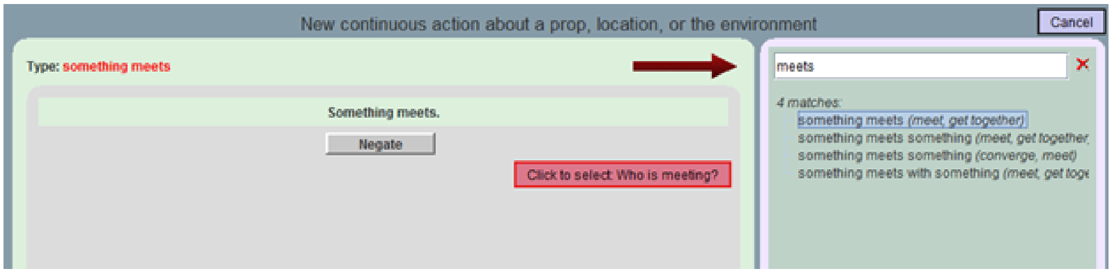
near the workplace

There once was a group of leaders. The group of leaders was meeting in order to talk about running a group of countries and near a workplace.

A group of peoples protested because the group of leaders was meeting and began to be peaceful.

Figure 4.10: Scheherazade screenshot of timeline layer for *Protest Story*

desired frame, annotators fill in the slots with the nouns and props previously defined.


New continuous action about a prop, location, or the environment
Cancel
Negate
Click to select: Who is meeting?

Type: something meets

Something meets.

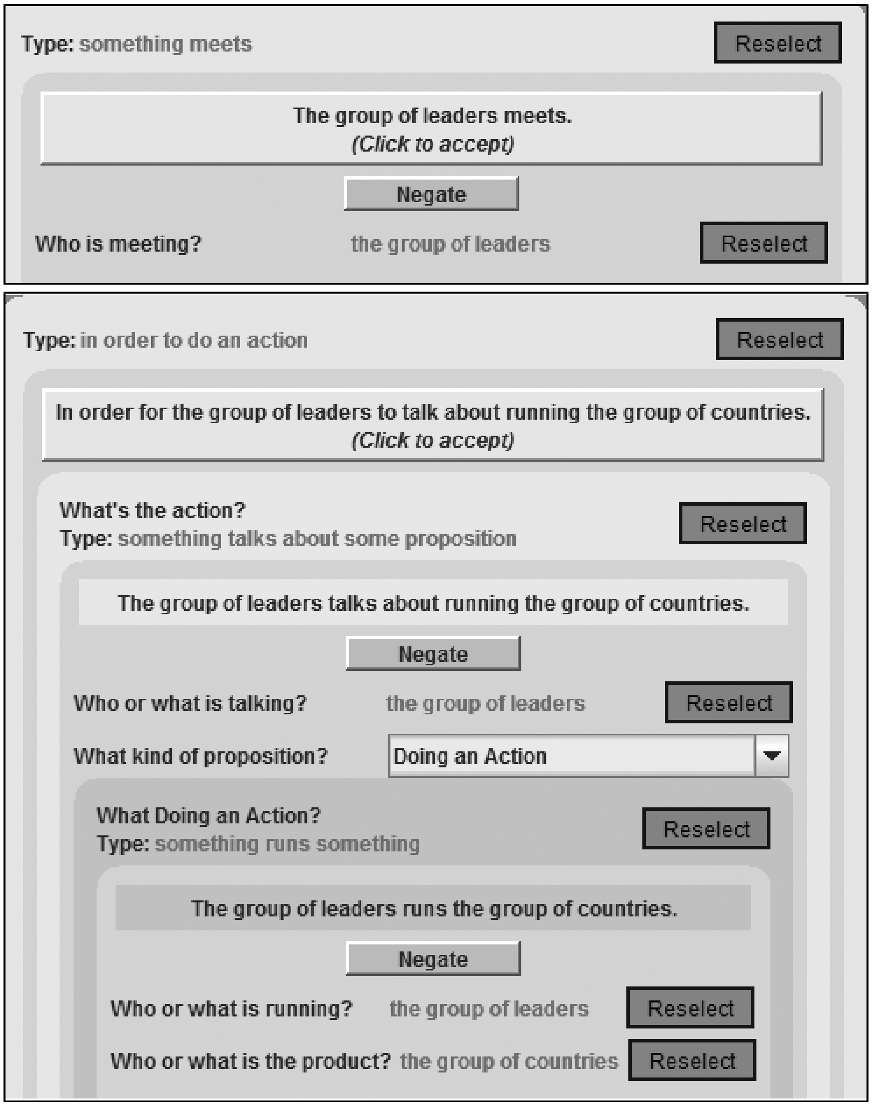
meets

4 matches:
something meets (meet, get together)
something meets something (meet, get together,
something meets something (converge, meet)
something meets with something (meet, get toge

Figure 4.11: Defining "meet" action in *Protest Story*

These story points create the timeline layer and make up a network of propositional structures (Figure 4.10 #3). Figure 4.10 #4 shows the WYSIWYM generation of the highlighted story point. The final encoding reflects one annotators' interpretation of the story, there are multiple possible encodings for each story, and the original DramaBank includes multiple encodings for some of the Aesop's Fables.

Figure 4.12 shows a screenshot of the Scheherazade annotation tool, illustrating the process of assigning propositional structure to the slots *The group of leaders meet in order to talk about running the group of countries*. This sentence is encoded as two nested propo-


Type: something meets

The group of leaders meets.
(Click to accept)

Who is meeting? the group of leaders

Type: in order to do an action

In order for the group of leaders to talk about running the group of countries.
(Click to accept)

What's the action?
Type: something talks about some proposition

The group of leaders talks about running the group of countries.

Who or what is talking? the group of leaders

What kind of proposition? Doing an Action

What Doing an Action?
Type: something runs something

The group of leaders runs the group of countries.

Who or what is running? the group of leaders

Who or what is the product? the group of countries

Figure 4.12: Scheherazade screenshot of propositional modeling

sitions `meet(group of leaders)` and the prepositional phrase `in order to(talk about running(the group of countries))`. Both actions (`meet` and `in order to talk`) contain references to the story characters and objects (`group of leaders` and `group of countries`) that fill in slots corresponding to semantic roles. Annotators continue to select text spans and annotate the story until they believe they have achieved good coverage of the story.

4.2.2 PersonaBank SIG Blog Corpus

We use Scheherazade to annotate a new corpus, PersonaBank, of 108 personal stories encoded as SIGs. These stories were selected from the Spinn3r corpus and annotated for story topic [Burton et al., 2009]. Swanson and Gordon [2012] use a lucene index in order to seed topics and rank stories from the 1.5 million stories. Starting with a list of seeds, for example, for a gardening topic we use [tree, trees, farm, garden, yarn, grass, plant, ...], we decide if the retrieved list of stories are relevant to the specified topic. Characteristics can be applied to the story to filter stories that are interesting, coherent, overall positive or negative. Story selection criteria for PersonaBank considers the following: the narrator is the storyteller, there a clear temporal sequence of events, and the story is not offensive for any reason.

We select 55 stories from this collection that are overall positive and 53 that are negative. *Bug Out For Blood*, *Startled Squirrel*, and *Protest Story* are all in PersonaBank. The stories are on average short: on average, each story has 269 words. We have a variety of topics and subtopics we assign based on observations in the data. The topics are not according to any known ontology, but our observed categories. Table 4.14 describes this distribution of the topics and subtopics of our stories. This table also shows how many of each topic are positive or negative stories. A few stories cover more than one topic. All names have been anonymized with “Anne”, “Jane”, “Jack” and “John”.

One motivation behind collecting and annotating stories for PersonaBank is that more than one domain shows generalizability. In selecting stories, our goal is to have at least one good representation of a story that we can use to test our story retelling apparatus. Therefore, all our

Statistics	Stories
Total stories	108
Positive stories	55
Negative stories	53
Average story length in words	269
Minimum	104
Maximum	959

Table 4.13: Overview Statistics of the PersonaBank Corpus

Topic (#pos, #neg)	Subtopics (#pos, #neg)
Health (1,15)	Life (1,1), Death (0,3), Sickness (0,4), Stress (0,2), Accident (0,3), Embarrassment (0,2)
Weather (8,1)	Snow (7,0), Storm (1,1)
Wildlife (10,3)	Squirrels (1,0), Bugs (1,1), Frogs (4,0), Fish (2,0), Birds (0,1), Sharks (1,1), Clams (1,0)
Activities (5,6)	Photography (1,0), Haircuts (0,4), Workouts (1,0), Gardening (2,0), Travel (1,2)
Sports (14,2)	Swimming (0,1), Scuba (4,0), Fishing (1,0), Running (1,0), Olympics (1,0), Camping (3,1), Sledding (4,0)
Holidays and Family (19,4)	Christmas (7,1), Easter (3,0), Family (9,3)
Romance (2,22)	New Romance (2,0), Breakups (0,22)
Everyday Events (10,8)	Dream (1,0), Arrest (0,1) Technology (3,0), Pets (3,1), Work (3,6)

Table 4.14: Topics and Subtopics of Annotated Stories

stories were annotated once with an expert annotator. While the DramaBank contains multiple encodings for some of the Aesop's Fables, we imagine it would be even more difficult with the complexity of personal narratives. Instead, we ensure that each annotation is very rich and complete in itself. The final realization reflects one annotator's interpretation of the story.

Trained undergraduate annotators associated with the Natural Language and Dialogue Systems Lab at the University of California, Santa Cruz, annotate the timeline layer of a story in about one hour. Annotating the interpretive and affectual layers requires more subjective judg-

Story id	Topic	Polarity	Excerpt
17	Health, Accident,	NEG	So my most recent strangely happy moment. I was in a car accident the other day. Sort of. Heheh. I was heading to Anne's B-day party (mixed with her brothers, they're all born in the same month) and this girl Jane hits me when I'm about to hit the stoplight coming off an exit. I get out of my car, she's spazzing.
57	Work, Everyday Events	NEG	Pf changs really messed up my training. It was one person really. If you wouldve seen this schedule i got you would understand. some of you did see it so you know what i mean. I went in last wednesday to take what i thought was my final training class. I was told i missed it and it was the day before. I was really confused becasue my schedule said i was off that day so i pulled it out and showed them. Then she says thats not your schedule. I was like its not what do you mean?
59	Weather, Snow	POS	The first day of winter is a huge event, especially for those who are in South Dakota. It is impossible to escape the snow if you're living in SD. For me, this year I was working when it was snowing. I was so sad because I was unable to go out and play in it like I have done ever since I was a child. Fortunately work ended shortly after that and I was able to call up my friends and head to their place.

Table 4.15: Excerpts from PersonaBank

ment and takes an additional hour for each story. We review the annotations with our annotators until they feel comfortable with the annotation process. We find that annotation is facilitated by using annotators that have a background in linguistics and most of our annotations have been produced by linguistics undergraduates working as research assistants. We are continually adding new stories to the corpus.

We create a new annotation tutorial specifically for personal narratives released with our language resource. We advise annotators that these blog stories are rich in language and description, and have many things that are not relevant to the annotation of the story events.

Before they begin annotating with the tool, they should read the entire story and determine the characters and events that are crucial to the plot and *fabula*.

When annotating personal narratives, there are some challenges in adapting the annotation process to the new domain. Personal narratives often contain much more description of the setting than what is typical for Aesop's Fables. Some words or concepts that are missing from WordNet or VerbNet are not easily represented using the SIG representation. In many cases, these descriptions do not pertain to the key aspects of the story. For example, the story 57 in Table 4.15 begins with the following sentences: *Pf changs really messed up my training. It was one person really. If you wouldve seen this schedule i got you would understand. some of you did see it so you know what i mean.* The action starts at the fifth sentence: *I went in last wednesday to take what i thought was my final training class.* We encourage annotators to ignore descriptive observations in texts such as these if they are not central to the action of the narrative, or to encode the observations appropriately as actions.

There are some situations where annotators cannot find the exact words or expressions from the original story in the WordNet or VerbNet dictionaries. We encourage them to choose an appropriate paraphrase that conveys the same concept. For example, the expletive *it* as in *It was hard to...* can be represented instead as *The situation was hard to...*. Our blogs often discuss events the narrator was involved with, and use the pronoun *we*. This cannot be annotated in WordNet or VerbNet, so *we decide to...* can be annotated as *A group of friends decided to...* where the annotator chooses an appropriate group of characters based on the context of the story. Similarly, annotators can use groups for plurals. Instead of *five trees*, annotators represent the group as *a group of trees*.

There are many possible interpretations of these stories, and thus many possible annotations. For the phrase *There was a protest that happened* an annotator may decide to annotate it as *the people protested against the group of leaders because the group of leaders was meeting* or *the people protested against the group of leaders because the people disagree about an ideality*. Annotators may instead choose a different verb for the deep representation such as *the people disagree about an ideality*, *the people dislike the government*, or *the people distrust the government because the people disagree about an ideality*.

There may be a question of consensus when subjects annotate SIGs. Elson [2012b] shows that there is high variance due to both how different people interpret stories, and how they select their annotations. In our work, we assume that our annotations are high quality and offer one interpretation of a *fabula*.

4.3 Summary

The EST translation introduced in Chapter 3 gives us access to the rich content from SIGs from which we induce a syntactic structure. This Chapter explores the variations that can be created from this generated syntactic structure. We build the Fabula Tales sentence planner, introduce narrative variations and hypothesize a significant correlation between each and reader engagement. First person point of view will bring the reader into the immediacy of a story. Direct speech allows characters to express themselves. Style features, including lexical choice and pragmatic markers can be used in direct speech to act as character voice. Deaggregation and aggregation creates more natural sounding and concise stories. We test these hypotheses in Chapter 5. Previous work in NLG narrative systems either are not able to represent any type of story domain, or they do not have a flexible sentence planner built for storytelling. The Fabula Tales architecture utilizes the EST and the SIG to provide a variety of different narrative retellings.

We introduce a new corpus that allows us to take advantage of the domain independent modeling affordances of the SIG. However, the SIG modeling is not yet automated, so we require human annotators. In order to explore and test our storytelling framework, we created and annotated a new corpus of personal narratives in the SIG formalism and created new guidelines for this annotation. These diverse stories also allow us to examine the domain independence of the EST framework.

Chapter 5 describes the evaluation and experimentation the EST as a translator, Fabula Tales as sentence planner, and reader perceptions and interpretations of the generated tellings.

Chapter 5

System Evaluation

It is difficult to evaluate a narrative NLG system effectively to judge the quality of its generated stories due to the subjective nature inherent in the task of creating different framings and measuring audience preference. First, we take a quantitative approach to develop and evaluate that our semantic-syntactic translation is comparable to a baseline generation. Section 5.1 describes quantitative evaluations that aim to ensure the generated content from the EST translation, without Fabula Tales' sentence planning, is accurate with respect to the WYSIWYM Scheherazade generation in order to measure the correctness of the EST semantic to syntactic translator. However, these quantitative evaluations can only take us so far; because our long term goal is to generate different retellings, we eventually want to measure the quality of different styles of retelling.

By priming stories differently, there are a number of aspects we believe will be affected by the different types of variation. We propose the following narrative metrics will holistically test the effectiveness of each retelling:

- Narrative immediacy: To what degree is the reader engaged with the story and characters
- Interest: To what degree would the reader desire to read the rest of the story
- Correctness: To what degree is the narrative well-formed
- Preference: Which framings do readers generally prefer to read
- Natural: To what degree is the language of the story natural

We propose a series of human evaluation tasks informed by previous research in this area [Callaway and Lester, 2002, Cheong and Young, 2008] to examine the quality and impact of these narrative metrics. These tasks (in Section 5.2) are posed as subjective survey questions that ask human subjects to rate the quality of a narrative on one of the desired aspects. We ask these questions both as a subjective valuation task using a rating scale, and as a ranking task between two or more alternative variations.

To evaluate the generated stories from the entire EST and Fabula Tales pipeline as well as the narrative parameters, we design a new evaluation paradigm based on overgenerate and rank methodology (Section 5.3). Fabula Tales generates many possible sentence variations from which readers select the sentences they like best to construct their own stories. We conclude with ablation and pairwise testing to prove the following domain independent hypotheses about general narrative parameters:

H_1: There is a significant correlation between first person point of view and engagement because it brings the reader into the immediacy of a story.

H_2: There is a significant correlation between direct speech and engagement because direct speech allows characters to express themselves.

H_3: There is a significant correlation between direct speech and certain styles and increased reader engagement because style features can be used in direct speech to act as character voice.

H_4: There is a significant correlation between domain independent deaggregation and engagement because aggregation creates more natural sounding and concise stories.

5.1 Development and Automatic Evaluation

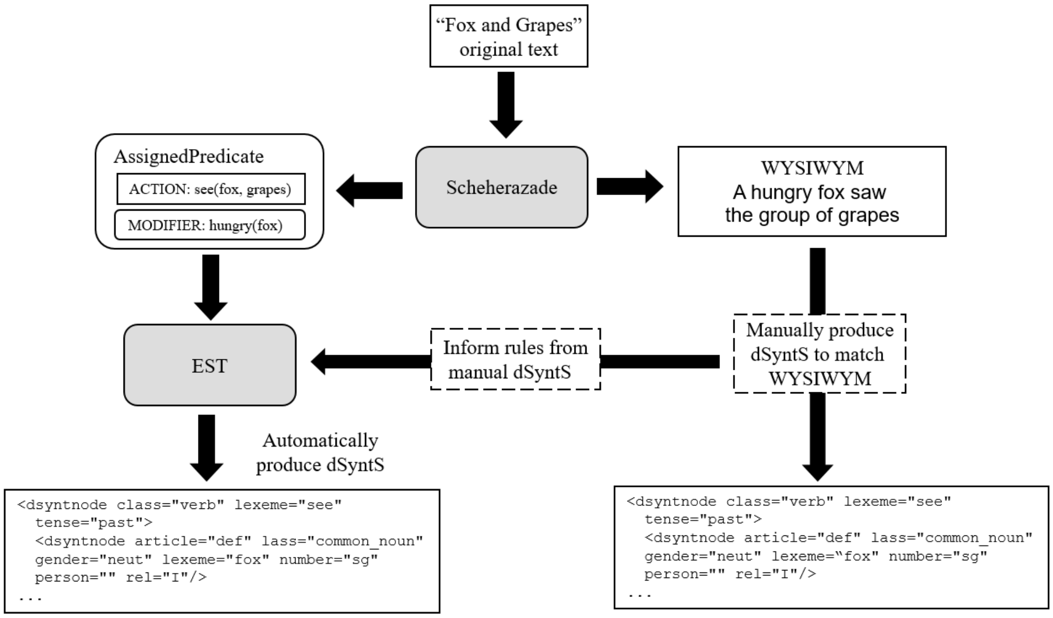
We develop and evaluate the EST rules using a neutral voice model and default narrative parameters, i.e. linear content planning, third person, no direct speech, no voice, no deaggregation, to realize a baseline referred to, again, as *Fabula Tales baseline*. We compare this to the Scheherazade WYSIWYM realization, which is realized directly from the SIG representation with its own rules, and treat this an objective function to ensure the essence of the story is preserved. Recall, that the WYSIWYM does not produce narrative variation, but only realizes the semantics of the SIG.

The WYSIWYM realization of "The Fox and The Grapes" from the DramaBank acts as a baseline gold standard for informing the Fabula Tales baseline reproduction (Table 5.1). We manually construct gold standard DSYNTS to faithfully produce this realization (right branch in Figure 5.1). Independently, in parallel, the EST was created with translation rules that derive the closest approximation of the gold standard DSYNTS (left branch of Figure 5.1).

Original Fable
A hungry Fox saw some fine bunches of Grapes hanging from a vine that was trained along a high trellis, and did his best to reach them by jumping as high as he could into the air. But it was all in vain, for they were just out of reach: so he gave up trying, and walked away with an air of dignity and unconcern, remarking, “I thought those Grapes were ripe, but I see now they are quite sour.”
WYSIWYM
Once, a group of grapes was hanging on a vine and the vine was hanging on a high trellis. A hungry fox saw the group of grapes. The fox jumped in order to obtain the group of grapes. The fox didn’t obtain the group of grapes because he wasn’t able to reach the group of grapes. The fox walked away from the group of grapes with dignity and with unconcern. The fox said that he had earlier thought that the group of grapes was ripe and said that he now saw it as being sour.
Fabula Tales baseline
The group of grapes hung on the vine. The vine hung on the trellis. The fox saw the group of grapes. The fox jumped in order for the fox to obtain the group of grapes. The fox did not obtain the group of grapes because the fox was not able to reach the group of grapes. The fox walked away from the group of grapes with dignity and unconcern. The fox said the fox earlier thought the group of grapes was ripe. The fox said the fox now saw the group of grapes was sour.

Table 5.1: “The Fox and the Grapes” for original story and baselines

Our primary goal of using “The Fox and the Grapes” is to generate text as close as possible from the DSYNTS the EST produces to the WYSIWYM output via the gold standard DSYNTS. A prerequisite for producing stylistic variations of a story is an ability to generate a “correct” retelling of the story. We first verify that the essential content of story is faithfully preserved when we translate from the SIG representation to DSYNTS using two metrics: BLEU score and Levenshtein distance. BLEU is an established standard for evaluating the quality of machine translation and summarization systems [Papineni et al., 2002]. The score ranges between 0 and 1, and measures the closeness of two documents, or stories, by comparing n-gram overlap, taking word order into account. BLEU score computes the overlap between two stories taking word order into consideration: a higher BLEU score indicates a closer match between candidate stories.


"Fox and Grapes" original text
Scheherazade
AssignedPredicate
ACTION: see(fox, grapes)
MODIFIER: hungry(fox)
WYSIWYM
A hungry fox saw the group of grapes
EST
Inform rules from manual dSyntS
Manually produce dSyntS to match WYSIWYM
Automatically produce dSyntS

```
<dsyntnode class="verb" lexeme="see"
  tense="past">
  <dsyntnode article="def" lass="common_noun"
  gender="neut" lexeme="fox" number="sg"
  person="" rel="I"/>
...
```

```
<dsyntnode class="verb" lexeme="see"
  tense="past">
  <dsyntnode article="def" lass="common_noun"
  gender="neut" lexeme="fox" number="sg"
  person="" rel="I"/>
...
```

Figure 5.1: Development of the EST (author actions in dashed boxes)

Levenshtein distance is the minimum edit distance between two strings. The objective is to minimize the total cost of character deletion, insertion, and replacement it takes to transform one string into another. In our case, we treat each word as a unit and measure word deletion, insertion, and replacement. We use word stemming as a preprocessing step to reduce the effect of individual word form variations. A lower score indicates a closer comparison.

These established metrics provide evidence that the generated words from EST match with the WYSIWYM realization. A criticism of this evaluation could question that this is an unfair comparison of matching syntactics against semantics. However, the DSYNTS do not have the semantic information of the SIG, so the comparison we make is against the syntactics of the semantics, in other words, the WYSIWYM which is a direct derivation of the semantics.

	WYSIWYM-FTBase	Orig-WYSIWYM	Orig-FTBase
Fables Dev	31	80	84
Fables Train	72 (40)	116 (41)	108 (31)
Blogs Test	110 (57)	736 (466)	733 (466)

Table 5.2: Levenshtein distance (lower is better)

	WYSIWYM-FTBase	Orig-WYSIWYM	Orig-FTBase
Fables Dev	.59	.04	.03
Fables Train	.32 (.11)	.06 (.02)	.03 (.02)
Blogs Test	.66 (.08)	.21 (.24)	.21 (.24)

Table 5.3: BLEU score (higher is better)

5.1.1 Developing EST on Fables

In Tables 5.2 and 5.3, **WYSIWYM-FTBase** compares the output of WYSIWYM with the Fabula Tales baseline. Because the rules were created on the development set to match the WYSIWYM realization, we use the results of **WYSIWYM-FTBase** to provide a topline for comparison to the test sets. On our development story "The Fox and The Grapes", we achieve a Levenshtein score of 31 and a BLEU score of 0.59 for **Fables Dev WYSIWYM-FTBase**. In machine translation applications, these scores indicate a mild amount of variance between generations, however we treat this as our topline in our translation since these metrics have not previously been applied to this type of problem.

We develop our rules to minimize Levenshtein distance or maximize BLEU score as much as possible for the development set. While overall we do very well in recreating the WYSIWYM realization for "The Fox and The Grapes", there are still issues with the Fabula Tales baseline realization. The EST did not support tense distinctions at the time of this first evaluation, defaulting everything to the past tense. This becomes particularly problematic in comparing the WYSIWYM realization for *The fox said that he had earlier thought* and *he now* *saw it as being sour* which in the Fabula Tales baseline is realized as *The fox earlier thought* and *The fox said the fox now saw the group of grapes was sour*. In addition, we note problems with the generation of distinct articles such as ‘a’ vs. ‘the’. There are a special set of surface realization rules in WYSIWYM that are currently missing from the EST that adds cue phrases such as ‘that’ and ‘once’. The EST also did not support coreference resolution at this time, resulting in many redundancies.

Next we compare the original story to the WYSIWYM and Fabula Tales baseline generation to see how different the language is from the original telling. This, in part, reflects the capacity of the Scheherazade annotation tool, which has a direct effect on the WYSIWYM realization and the EST derivation. Thus, these statistical results should not be compared to **WYSIWYM-FTBase** since both of these are outputs of an NLG engine without variations to induce sentence planning variation. The natural statement *A hungry Fox saw some fine bunches of Grapes hanging from a vine that was trained along a high trellis* is broken up into three separate sentences in Scheherazade and EST: *The group of grapes hung on the vine. The vine hung on the trellis. The fox saw the group of grapes.* The language of the original story is more natural (compare *and did his best to reach them by jumping as high as he could into the air* with *The fox jumped in order for the fox to obtain the group of grapes*). These obvious differences between the original fables and the generated versions which cause Fables Dev **Orig-WYSIWYM** to have a high Levenshtein distance of 80 and low BLEU of 0.04, and Fables Dev **Orig-FTBase** to have a Levenshtein of 84 and BLEU of 0.03.

We use a two-tailed Student’s t-test to compare the two realizers’ mean distance values to the original fables and determine statistical significance. The difference in Levenshtein distance between **Orig-WYSIWYM** and **Orig-FTBase** is not statistically significant (p = 0.08) on both development and test sets. This indicates that our rules generate a story which is similar to what WYSIWYM generates in terms of closeness to the original. However, Scheherazade shows a higher BLEU score with the original fables ($p < 0.001$). This is due to the fact that our translation rules assume a simple generation, not attempt to express tense, aspect or stative/non-stative complications, all of which contribute to a lower overlap of n-grams. The assignment of tense and aspect was an area of particular focus for Scheherazade's built-in realizer [Elson and McKeown, 2010].

5.1.2 Training EST on Fables

We then evaluate our transformation method on a training set of 35 other fables from DramaBank without adding any addition rules to the EST. We find an increase of Levenshtein distance to 71 and decrease of BLEU to 0.32 when we compare Fables Train **WYSIWYM-FTBase** on the 35 fables. The high variance in distances across the test dataset indicates a wide range of story complexity in DramaBank. An examination of the variance demonstrates that both WYSIWYM and FTBase have difficulty with the complexity of some stories. For example, "The Lion and The Boar" from the fables training set is shown in Table 5.4. Part of the WYSIWYM realization is *The group of vultures began to plan - if the boar were to later die, and the lion were to later die - for the group of vultures to eat.* This involves a hypothetical timeline and an if-then statement. Because this construction did not appear in "The Fox and the Grapes", the EST did not originally offer support for this statement, resulting in an ill-formed Fabula Tales baseline realization from the EST translation as *The group of vultures planned the*

Original Fable
On a summer day, when the great heat induced a general thirst, a Lion and a Boar came at the same moment to a small well to drink. They fiercely disputed which of them should drink first, and were soon engaged in the agonies of a mortal combat. On their stopping on a sudden to take breath for the fiercer renewal of the strife, they saw some Vultures waiting in the distance to feast on the one which should fall first. They at once made up their quarrel, saying: “It is better for us to make friends, than to become the food of Crows or Vultures, as will certainly happen if we are disabled.”
WYSIWYM
Once, the air was hot. A boar decided to drink from a spring, and a lion decided to drink from the spring. The boar quarrelled about whether later first drank from the spring, and the lion quarrelled about whether later first drank from the spring. The boar began to attack the lion, and the lion began to attack the boar. The boar stopped attacking the lion, and the lion stopped attacking the boar. The boar above saw a group of vultures being seated on some rock, and the lion above saw the group of vultures being seated on the rock. The group of vultures began to plan - if the boar were to later die, and the lion were to later die - for the group of vultures to eat. The boar sobered, and the lion sobered. The boar said to the lion that - if the lion were to not kill the boar - the group of vultures would not eat the boar, and the lion said to the boar that - if the boar were to not kill the lion - the group of vultures would not eat the lion. The boar didn’t kill the lion, and the lion didn’t kill the boar.
Fabula Tales baseline
The air was hot. The lion decided the lion drank from the spring. The boar decided the boar drank from the spring. The boar quarreled. The lion quarreled. The lion attacked the boar. The boar attacked the lion. The boar above saw the group of vultures was seated on the rock. The lion above saw the group of vultures was seated on the rock. The group of vultures planned the group of vultures ate. The boar sobered. The lion sobered. The boar said the group of vultures did not eat the boar to the lion. The lion said the group of vultures did not eat the lion to the boar. The boar did not kill the lion. The lion did not kill the boar.

Table 5.4: “The Lion and The Boar” for original story and baselines

group of vultures ate which not only excludes the proposition “if the boar were to later die, and the lion were to later die”, but also has an incorrect syntax; the realization should be “The group of vultures planned to eat”.

5.1.3 Testing EST on Blogs

After developing rules to further improve performance to accommodate the fable training set, we run the evaluation on a test set of 108 blogs from PersonaBank. To extend to the blog domain, more rules were added to the EST, including support for the limitations of multiple hypothetical timelines and if-then statements from *The Lion and The Boar* discussed above. Table 5.5 shows more examples including support for key words like "every" in *The narrator managed to kill every bug* and inclusive of prepositions *The big bug's fatty undoubtedly wanted to retaliate against her* which previously would have excluded the "against her".

Table 5.2 and Table 5.3 provide quantitative evidence that the style of the original blogs is very different from Aesop's Fables. The differences in Fables Train **WYSIWYM-FTBase** and Blogs Test **WYSIWYM-FTBase** is 72 and 110 respectively for Levenshtein and 0.32 and 0.66 for BLEU respectively. This indicates that it is difficult to even reproduce the WYSIWYM realization of the blog stories. *Bug Out For Blood* in Blogs Test generates the following for WYSIWYM from one AssignedPredicate:

> The narrator began to look around every corner of the apartment of the narrator, began to check the toilet seat of the narrator for the fatty of the group of bugs in order for she to sit down on the toilet seat of the narrator and she in due course expected for the fatty of the group of bugs to jump toward her

This sentence combines many propositions into one long sentence. The EST breaks this down into multiple sentences, but the repetitiveness of "the narrator" and the conjunctions in WYSIWYM create a difference in computing metrics:

> The narrator looked around every corner of my apartment. The narrator checked her toilet seat for the big bug's fatty in order for her to sit down on her toilet seat. The narrator in due course expected for the big bug's fatty to jump toward her.

Original Blog
Bug out for blood the other night, I left the patio door open just long enough to let in a dozen bugs of various size. I didn't notice them until the middle of the night, when I saw them clinging to the ceiling. Since I'm such a bugaphobe, I grabbed the closest object within reach, and with a rolled-up comic book I smote mine enemies and smeared their greasy bug guts. All except for the biggest one. I only clipped that one, taking off one of its limbs.
WYSIWYM
A narrator recently and momentarily opened the door of the patio of the narrator. A group of bugs entered the apartment of the narrator. The narrator didn't initially notice that the group of bugs had entered the apartment of the narrator. The narrator slept. The narrator awaked overnight and saw the group of bugs being on the ceiling of the apartment of the narrator. The narrator grabbed a reachable and close thing that was a rolled comic book because the group of bugs scared her, and it began to be the enemy of the narrator. The narrator began to hit the group of bugs with the comic book. The narrator smeared the greasy innards of the group of bugs and managed to kill every bug who wasn't the big fatty of the group of bugs who was more large than every bug.
Fabula Tales baseline
The narrator recently momentarily opened her patio's door. The bugs entered her apartment. The narrator did not initially notice that the bugs entered her apartment. The narrator slept. The narrator overnight awoke. The narrator saw the bugs was on her apartment's ceiling. The narrator grabbed the comicbook because the bugs scared her. The bugs were her enemy. The narrator smeared the greasy bug' innards. The narrator managed to kill every bug.

Table 5.5: *Bug Out For Blood* for original story and baselines

However we find that Fabula Tales baseline compares favorably to WYSIWYM on the blogs with a relatively low Levenshtein score, and higher BLEU score (Blogs Test **WYSIWYM-FTBase**) than the original Fables training evaluation (Fables Train **WYSIWYM-FTBase**). This indicates that even though the blogs have a diversity of language and style, our EST translation comes close to the WYSIWYM baseline. We get broad coverage at every stage, even using only the original rules developed on the "Fox and Grapes". Even though we developed the EST methodology from a single Fable story, it applies that the translation rules are sufficiently general. In our successive development of test set and blogs data set, we added corner cases that we had not seen (i.e. if-then statements).

5.2 Reader Perceptions of Parameters

5.2.1 Character Attitudes

We now shift our evaluations to collect reader perceptions of the "The Fox and the Crow" generated with direct speech and with different personality models (character voices) for each speech act, hypothesizing that different voice styles will effect readers' perceptions of the characters. We produce three versions of the story: the first entirely in the neutral voice, the second with the crow using a shy voice and the fox using a laid-back voice, and the third using a shy voice for the fox and a laid-back voice for the crow (Table 5.6).

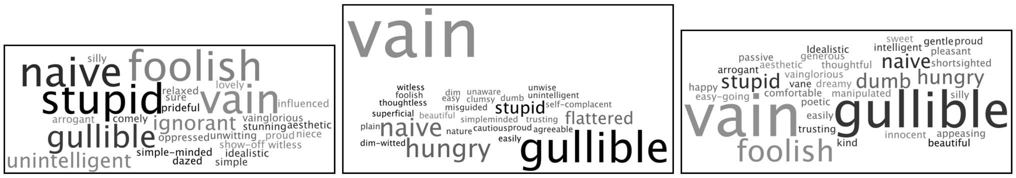
Voice	Generated Sentences
Laid-back	The fox said "Your feather's chromaticity is damn exquisite."
Shy	The fox said "Your feather's chromaticity is so-somewhat exquisite."
Neutral	The fox said "Your feather's hue is exquisite."

Table 5.6: Voice variations in blogs and fables

Crow	% Pos	% Neg	Fox	% Pos	% Neg
Neutral	13	29	Neutral	38	4
Shy	28	24	Shy	39	8
Laid-back	10	22	Laid-back	34	8

Table 5.7: Polarity of adjectives describing the Crow and Fox (% of total words)

Each version is shown to native English speakers on Mechanical Turk. They are given a free text box and asked to enter as many adjectives as they wish to describe the characters in the story. Table 5.7 shows the percentage of positive and negative descriptive words when categorized by the LIWC [Pennebaker et al., 2001]. Figures 5.2 and 5.3 show word clouds for the descriptions of the fox and crow. Some attributes include "clever" and "sneaky" for the

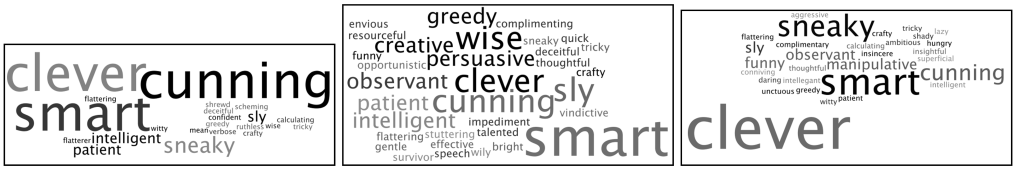


(a) Crow Neutral

(b) Crow Shy

(c) Crow Friendly

Figure 5.2: Word cloud adjectives for the Crow



(a) Fox Neutral

(b) Fox Shy

(c) Fox Friendly

Figure 5.3: Word cloud adjectives for the Fox

laid-back and neutral fox, and “shy” and “wise” for the shy fox. The laid-back and neutral crow was perceived as “naïve” and “gullible” whereas the shy crow is “stupid” and “foolish” [Lukin et al., 2014].

Overall, the crow’s shy voice is perceived as more positive than the crow’s neutral voice, (ttest(12) = -4.38, $p < 0.0001$), and the crow’s laid-back voice (ttest(12) = -6.32, $p < 0.0001$). We hypothesize that the stuttering and hesitations in the shy voice make the character seem gullible and less at fault in this context compared to the laid-back voice which is more boisterous. However, both the stuttering shy fox and the boisterous laid-back fox were seen equally as “cunning” and “smart”. Although we observe statistical difference in perception in only the crow’s voices, this is enough evidence to warrant further investigation of how reader perceptions change when the same content is realized in difference voices.

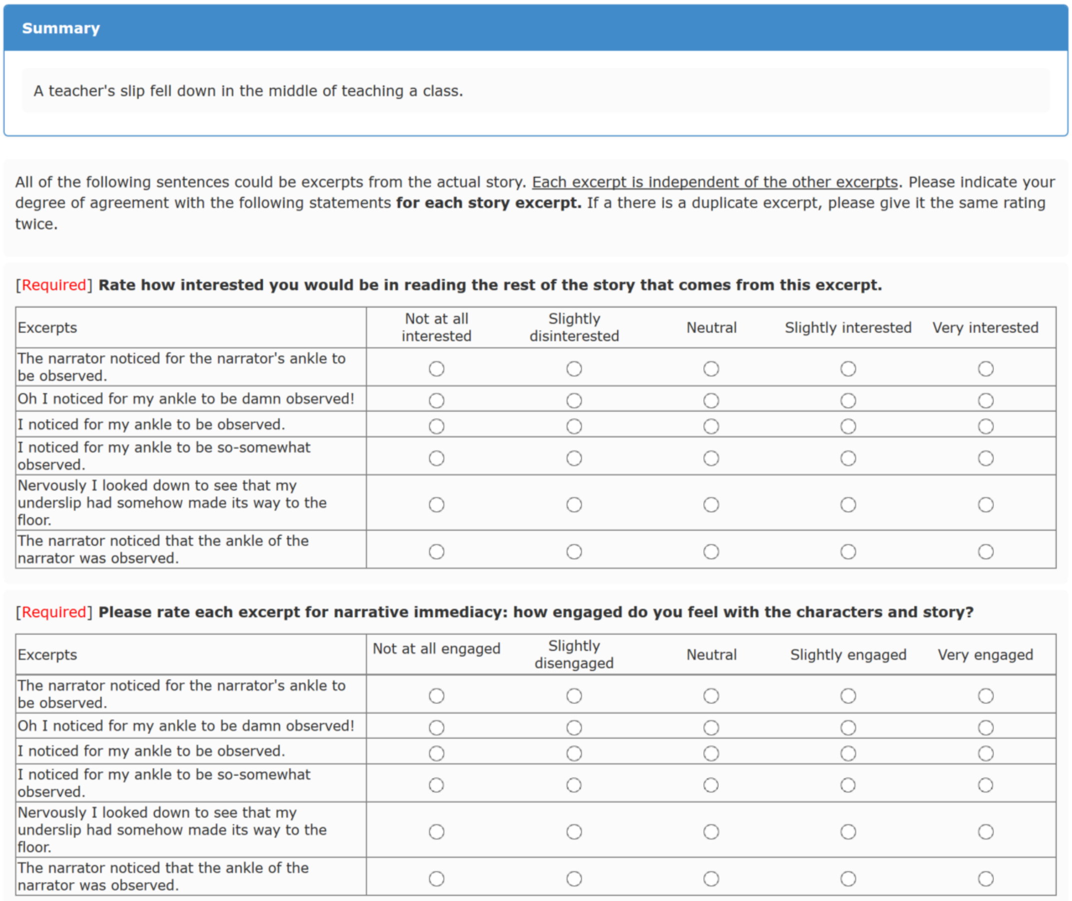
Note that these traits cannot be directly inferred through the SIG annotation unless the trait was explicitly annotated. But we can influence perceptions via models, such as PERSONAGE's introvert and extrovert models, and extend this to emotional models to further the effect of engagement (Section 6.3.1).

5.2.2 Engagement and Interest

Having found that character voice affects reader interpretation of characters, we examine how much different points of view combined with different voices interact with engagement and interest with a story, again, hypothesizing that excerpts in the first person will always be preferred over third person realization (H_1). We also compare an excerpt from the original blog story to test how close our best Fabula Tales realizations come to the fluid language of the blog, finally starting to tackle going beyond syntactic matching evaluated in Section 5.1.

We present native English speakers on Mechanical Turk with a one sentence summary of one of seven stories from PersonaBank and six generated variations of one sentence from that story, including Fabula Tales generations using the first person with a neutral, shy, and laid-back voice, a third person with a neutral voice, a sentence from the original blog story, and a sentence from WYSIWYM. These sentences are framed as "possible excerpts that could come from this summary" and subjects are asked to rate each excerpt on a 1-5 point scale for their interest in wanting to read more of the story based on the style and information given in the excerpt, and to indicate their engagement with the story given the excerpt. Figure 5.4 shows an example of the *Embarrassed Teacher* and its six retellings [Lukin and Walker, 2015].

Table 5.8 shows the means and standard deviation for engagement and interest ratings. We find an ordered ranking for engagement: the original sentence from the blog is scored



Summary

A teacher's slip fell down in the middle of teaching a class.

All of the following sentences could be excerpts from the actual story. Each excerpt is independent of the other excerpts. Please indicate your degree of agreement with the following statements **for each story excerpt.** If a there is a duplicate excerpt, please give it the same rating twice.

[Required] **Rate how interested you would be in reading the rest of the story that comes from this excerpt.**

Excerpts	Not at all interested	Slightly disinterested	Neutral	Slightly interested	Very interested
The narrator noticed for the narrator's ankle to be observed.	○	○	○	○	○
Oh I noticed for my ankle to be damn observed!	○	○	○	○	○
I noticed for my ankle to be observed.	○	○	○	○	○
I noticed for my ankle to be so-somewhat observed.	○	○	○	○	○
Nervously I looked down to see that my underslip had somehow made its way to the floor.	○	○	○	○	○
The narrator noticed that the ankle of the narrator was observed.	○	○	○	○	○

[Required] **Please rate each excerpt for narrative immediacy: how engaged do you feel with the characters and story?**

Excerpts	Not at all engaged	Slightly disengaged	Neutral	Slightly engaged	Very engaged
The narrator noticed for the narrator's ankle to be observed.	○	○	○	○	○
Oh I noticed for my ankle to be damn observed!	○	○	○	○	○
I noticed for my ankle to be observed.	○	○	○	○	○
I noticed for my ankle to be so-somewhat observed.	○	○	○	○	○
Nervously I looked down to see that my underslip had somehow made its way to the floor.	○	○	○	○	○
The narrator noticed that the ankle of the narrator was observed.	○	○	○	○	○

Figure 5.4: Screenshot of point of view and voice experiment for interest and narrative immediacy

highest, followed by first outgoing, first neutral, first shy, WYSIWYM, and third neutral. Figure 5.5 shows the average engagement and interest for all the sentences. We apply a Bonferroni correction to paired t-tests of engagement show that a significant difference between original and first outgoing (ttest(94) = -3.99, $p < 0.0001$), first outgoing and first shy (ttest(94) = 3.71, $p < 0.0001$), and first shy and WYSIWYM (ttest(94) = 5.60, $p < 0.0001$). However, there are no differences between first neutral and first outgoing (ttest(95) = -1.63, $p = 0.05$), and WYSIWYM and third neutral (ttest(94) = -0.31, $p = 0.38$). We expected that the original sentence would

Engagement	Orig	1st-out	1st-neutr	1st-shy	WYSIWYM	3rd-neutr
Mean	3.98	3.27	3.00	2.73	1.95	1.93
SD	1.07	1.39	1.19	1.25	1.07	1.06

Interest	Orig	1st-out	1st-neutr	1st-shy	WYSIWYM	3rd-neutr
Mean	3.91	3.02	3.02	2.81	1.90	1.87
SD	0.99	1.21	1.37	1.27	1.05	1.01

Table 5.8: Means and standard deviation for engagement and interest in perceptions experiment

score the highest because it is human authored text. Subjects next preferred all first person realizations claiming that the immediacy of the first person made them feel they were there in the story (H_1). However, as we discuss in Section 5.3, H_1 is not true at the story level.

Paired t-tests, with Bonferroni correction, for interest follow a similar trend and show a significant difference between original and first outgoing (ttest(93) = 5.59, $p < 0.0001$), and first shy and WYSIWYM (ttest(93) = 6.16, $p < 0.0001$). There is no difference between first outgoing and first neutral (ttest(93) = 0, $p < 0.5$), first neutral and first shy (ttest(93) = 2.20, $p = 0.01$), and WYSIWYM and third neutral (ttest(93) = 0.54, $p = 0.29$). An ANOVA found a significant effect on style (F(1) = 204.08, $p < 0.0001$), sentence (F(9) = 7.32, $p < 0.0001$), but no interaction between style and sentence (F(9) =0.64, $p = 1$) which we interpret as the combination of story content and style having no effect on subject's interest, as opposed to engagement.

We also performed an ANOVA and found there is a significant effect on style (F(1) = 224.24, $p < 0.0001$), sentence (F(9) = 5.49, $p < 0.0001$), and an interaction between style and sentence (F(9) =1.65, $p < 0.1$), which implies that for engagement, story content and style

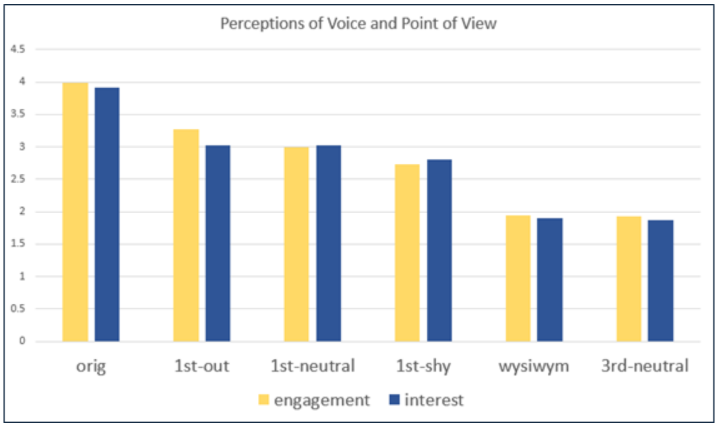
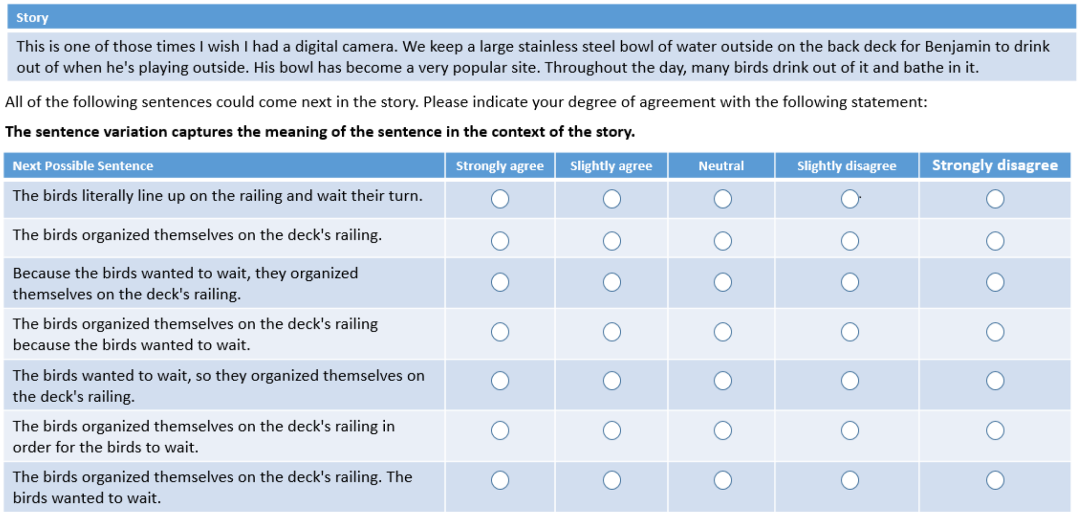


Figure 5.5: Engagement and interest for perceptions averaged across stories (higher is better)

together have an effect on subjects, suggesting that certain styles of narration are more appropriate or preferred than others given the context of the story. For example, subjects comment that the “curse words are used to express the severity of the situation wisely” and “adding the feeling of nervousness and where she looked made sense”, acknowledging the style fitting the situation. Information from the story may be used to influence and produce a more engaging realization, suggesting validation of our future work of applying emotion models based on character appraisal to suit the emotional intensity of the situation.

5.2.3 Correctness and Preference

We hypothesize that the deaggregation variations introduce more natural realizations from Fabula Tales (H_4), and explore 1) how the variations compare to each other; 2) if they come close to the natural language of the original blog story; and 3) if the Fabula Tales realization surpass the WYSIWYM realization. We create a Mechanical Turk experiment showing an excerpt from the original story and indicate to native English speakers that “any of the following



Story

This is one of those times I wish I had a digital camera. We keep a large stainless steel bowl of water outside on the back deck for Benjamin to drink out of when he's playing outside. His bowl has become a very popular site. Throughout the day, many birds drink out of it and bathe in it.

All of the following sentences could come next in the story. Please indicate your degree of agreement with the following statement:

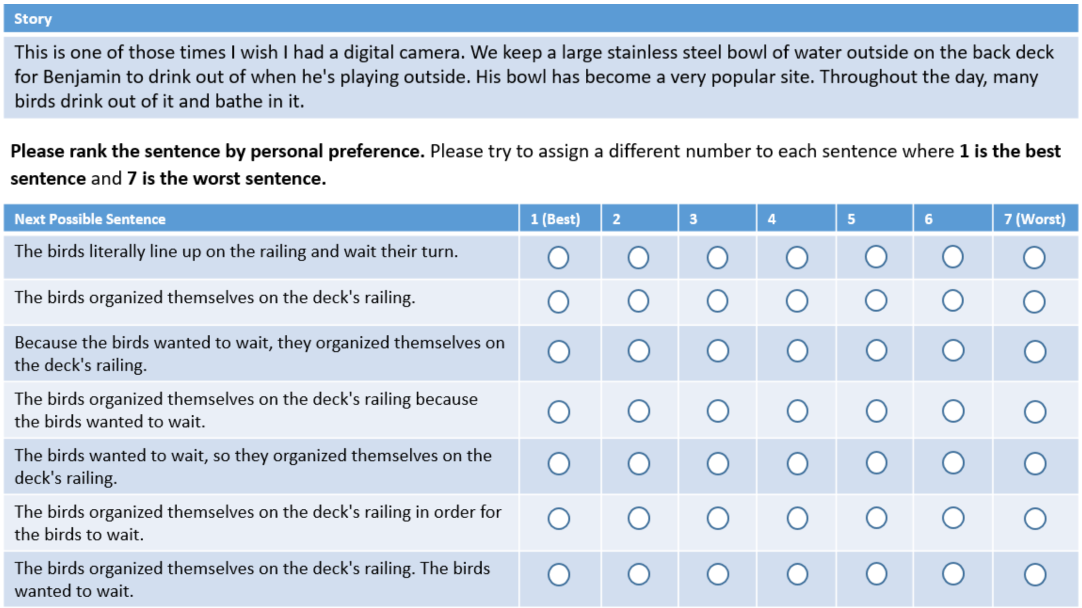
The sentence variation captures the meaning of the sentence in the context of the story.

Next Possible Sentence	Strongly agree	Slightly agree	Neutral	Slightly disagree	Strongly disagree
The birds literally line up on the railing and wait their turn.	○	○	○	○	○
The birds organized themselves on the deck's railing.	○	○	○	○	○
Because the birds wanted to wait, they organized themselves on the deck's railing.	○	○	○	○	○
The birds organized themselves on the deck's railing because the birds wanted to wait.	○	○	○	○	○
The birds wanted to wait, so they organized themselves on the deck's railing.	○	○	○	○	○
The birds organized themselves on the deck's railing in order for the birds to wait.	○	○	○	○	○
The birds organized themselves on the deck's railing. The birds wanted to wait.	○	○	○	○	○

Figure 5.6: Screenshot of deaggregation experiment 1 for correctness

sentences could come next in the story" (Figures 5.6 and 5.7) [Lukin et al., 2015]. Subjects are queried about the variations in terms of correctness and goodness of fit within the story context. They are then asked to rank the sentences by personal preference. We emphasize that subjects should read each variation in the context of the entire story, and encourage them to reread the story with each new sentence to understand this context.

In the first deaggregation experiment, seven participants on Mechanical Turk analyzed 16 story segments with the following variations: the original story, soSN, becauseNS, becauseSN, NS, N, and the non-deaggregated realization. All participants were native English speakers. Table 5.9 shows the means and standard deviations for correctness and preference rankings in the first experiment. We performed an ANOVA on preference and found that story had no significant effect on the results (F(1, 15) = 0.18, p = 1.00), indicating that all stories were well-formed and there were no outliers in the story selection. On the other hand, realization did have a significant effect on preference (F(1, 6) = 33.74, p < 0.0001). This supports our



Story

This is one of those times I wish I had a digital camera. We keep a large stainless steel bowl of water outside on the back deck for Benjamin to drink out of when he's playing outside. His bowl has become a very popular site. Throughout the day, many birds drink out of it and bathe in it.

Please rank the sentence by personal preference. Please try to assign a different number to each sentence where **1 is the best sentence** and **7 is the worst sentence.**

Next Possible Sentence	1 (Best)	2	3	4	5	6	7 (Worst)
The birds literally line up on the railing and wait their turn.	○	○	○	○	○	○	○
The birds organized themselves on the deck's railing.	○	○	○	○	○	○	○
Because the birds wanted to wait, they organized themselves on the deck's railing.	○	○	○	○	○	○	○
The birds organized themselves on the deck's railing because the birds wanted to wait.	○	○	○	○	○	○	○
The birds wanted to wait, so they organized themselves on the deck's railing.	○	○	○	○	○	○	○
The birds organized themselves on the deck's railing in order for the birds to wait.	○	○	○	○	○	○	○
The birds organized themselves on the deck's railing. The birds wanted to wait.	○	○	○	○	○	○	○

Figure 5.7: Screenshot of deaggregation experiment 1 for preference

hypothesis that the realizations are distinct from each other and there are preferences amongst them.

Figure 5.8 shows the average correctness and preference for all stories. Paired t-tests with Bonferroni correction showed a significant difference in reported correctness between **orig** and **soSN** ($p < 0.05$), but no difference between **soSN** and **becauseNS** ($p = 0.133$), or **becauseSN** ($p = 0.08$). There was a difference between **soSN** and **NS** ($p < 0.005$), as well as between the two different **because** operations and **NS** ($p < 0.05$). There were no other significant differences. We find that averaged across all stories, there is a clear order for correctness and preference: original, soSN, becauseNS, becauseSN, NS, non-deaggregated (indicated as None), N; suggesting that one variations are better than others, regardless of story context.

There are larger differences on the preference metric. Paired t-tests with Bonferroni correction show that there is a significant difference between **orig** and **soSN** ($p < 0.0001$) and

	Orig	soSN	becauseNS	becauseSN	NS	EST	N
Correctness	1.8	2.3	2.4	2.5	2.7	2.7	3.0
Preference	2.4	3.1	3.7	3.7	4.3	4.9	4.9

Table 5.9: Means for correctness and preference for deaggregation experiment 1 (lower is better)

soSN and **becauseNS** ($p < 0.05$). There is no difference in preference between **becauseNS** and **becauseSN** (p = 0.31). However there is a significant difference between **soSN** and **becauseSN** ($p < 0.005$) and **becauseNS** and **NS** ($p < 0.0001$). Finally, there is significant difference between **becauseSN** and **NS** ($p < 0.005$) and **NS** and **None** ($p < 0.005$). There is no difference between **None** and **N** (p = 0.375), but there is a difference between **NS** and **N** ($p < 0.05$). Overall, there are similar significant differences as in the correctness experiment. These results indicate that the original sentence, as expected, is the most correct and preferred. Subjects commented that while all variations were sufficient, most were “boring”, except for the original blog story excerpt.

Furthermore, we find a significant interaction between story and realization (F(2, 89) $= 1.70$, $p < 0.0001$), thus subjects’ preference of the realization are based on the context of the story. The N and NS variations are overall ranked at the lowest because they sometimes produce stilted language and remove pieces of content. However, in a few instances, these variations are ranked highly because the information they remove was deemed to be redundant in text realization or repeated content.

In the second deaggregation experiment, we compare the original blog sentence with the highest scoring deaggregation variation with a change of point of view, and the realization

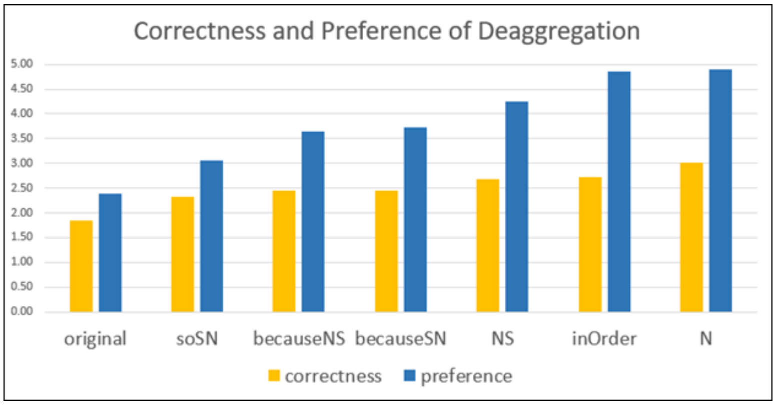


Figure 5.8: Correctness and preference for deaggregation experiment 1 (lower is better)

produced by WYSIWYM. We expect that WYSIWYM will score poorly in this instance because it cannot change point of view from third person to first person, even though its output is more fluent than Fabula Tales output in many cases. Seven native English speaking subjects analyzed each of the 19 story segments in a similar experimental setup as deaggregation experiment 1.

Table 5.10 shows the means and standard deviations for correctness and preference rankings for the second experiment. There is a clear order for correctness and preference: original, **soSN**, WYSIWYM, with significant differences between all pairs of realizations with Bonferroni correction ($p < 0.0001$). For the majority of the stories, subjects do not select WYSIWYM because of “the narrator” realization, commenting “forget the narrator sentence. From here on out it’s always the worst!”.

There are three story segments where WYSIWYM is rated on average higher than soSN. Upon closer examination, these story segments do not contain “I” or “the narrator” in the realization sentence, so the story segment is evaluated without the “narrator” bias. Compare the **soSN** realization in the *Protest Story*: *The leaders wanted to talk, so they met near the work-*

	Original	soSN	WYSIWYM
Correctness	1.6	2.5	3.5
Preference	1.4	1.9	2.7

Table 5.10: Means for correctness and preference for deaggregation experiment 2 (lower is better).

place with the **WYSIWYM**: *The group of leaders was meeting in order to talk about running a group of countries and near a workplace*. While Fabula Tales sentence planning have massively improved and overall is preferred to **WYSIWYM**, some semantic components are lost in the translation process from the SIG. Overall, the natural language of original blog is constantly preferred, yet the **soSN** realization comes close in ranking to matching it, though not statistically significant. We observe a distinct ordering in the deaggregation realizations, suggesting an overall general preference trend.

We are no longer interested in matching one realization to another, as was the goal with the BLEU and Levenshtein computation. As opposed to the evaluations in Section 5.1, here, these evaluations show the realizations using the Fabula Tales parameters are more interesting, correct, and preferred than the WYSIWYM and the Fabula Tales baseline. In the next section, we offer further support for our hypotheses.

5.3 Overgenerate and Rank Evaluation

We use the overgenerate and rank approach, first used in Langkilde and Knight [1998], to test the Fabula Tales sentence planning framework. Section 5.1 first validated that the baseline story is reproducible by the EST. We shifted our focus in Section 5.2 to then apply selected narrative variations of Fabula Tales at the sentence level, showing our narratological parameters are effective and preferred across a variety of blogs. We pull together the entire pipeline to generate and evaluate our parameters at the entire story level to prove our four hypotheses lead to engaging and preferred stories.

Recall our hypotheses: there is a significant correlation between first person point of view and engagement because first person brings the reader into the immediacy of a story (H_1); there is a significant correlation between direct speech and engagement because direct speech allows for characters to express themselves (H_2); there is a significant correlation between direct speech and certain styles leading to increased engagement (H_3); and finally there is a significant correlation between deaggregation and engagement to make more natural sounding and concise stories (H_4).

In the overgeneration phase, we showcase the flexibility of the system by generating hundreds of sentence variations using different combinations of narrative parameters. We create a novel evaluation paradigm for overgenerate and rank where annotators on Mechanical Turk construct their own story sentence-by-sentence by selecting from a subset of the generated sentences. This allows subjects to study the differences between the individual sentences and how they flow together in the context of an entire story. In the rank phase, we show the effectiveness of the parameters by performing statistical analysis to rank features in the selected sentences. We use these results to generate complete stories on which we conduct ablation tests by comparing stories without any variation against stories with high ranking features to test our hypotheses.

The EST and Fabula Tales can generate hundreds of different sentences that can be combined to create millions of complete stories. Figure 5.9 shows the process: we conduct an

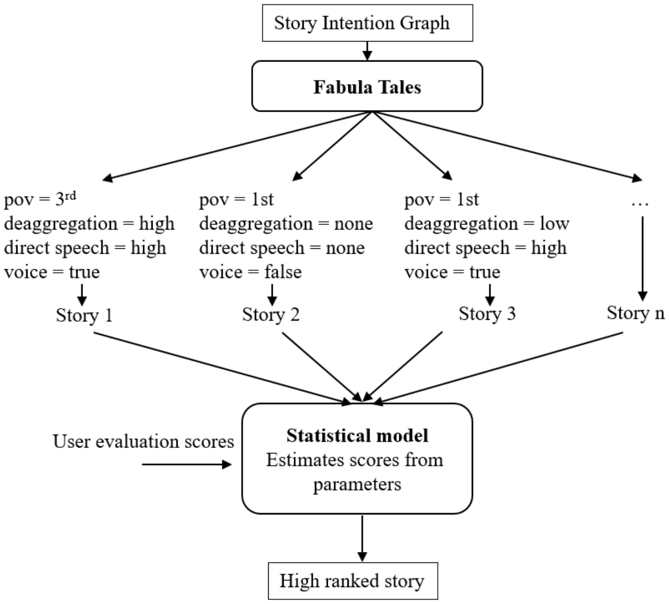


Figure 5.9: Overgenerate and rank process

experiment where subjects "Construct Your Own Story" sentence-by-sentence selecting from a subset of these generated variations. Subjects, native English speakers in our experiment, study the differences between the individual sentences and how the sentences fit together in the context of an entire story. Based on the sentences selected and the qualitative feedback of the subjects, we use the insights from the statistics and again, overgenerate creating entire stories without any parameters and compare against stories with different degrees of parameters.

5.3.1 Evaluation Paradigm

The overgenerate approach allows us to showcase the power of our parameterizable NLG engine and the generalizability of our narratological parameters. We make use of the narrative parameters point of view, quoted speech, deaggregation, and the following voice categories using PyPersonage: acknowledgments (e.g. "oh"), emphasizers ("actually", "rather"), competence mitigations ("come on"), down tones ("I mean"), and tag questions ("no?"). Many of these overlap with Biber's emotive, conative, modal and uncertainty words indicative of personal texts [Biber, 1991]. *"Oh, I didn't receive the new schedule!" said Anne* is defined as Character Voice. The acknowledgment "oh" and exclamation marks inside the quoted speech reflect the mental and emotional state of the character. Narrator Voice can use similar parameters but excludes the quoted speech, e.g. *Oh Anne exclaimed that she did not receive the new schedule.*

We select three stories from PersonaBank to generate different versions of each story using Fabula Tales' narrative parameters: *Bug Out For Blood*, *Startled Squirrel*, and *Employer Botches Training* for which we generate both first person and third person point of view stories treated independently. Depending on the compatible narratological, structural, or lexical features present in the encoding, Fabula Tales can produce up to 40 unique variations per sentence, yielding a huge potential combination of complete stories.

Tables 5.11 and 5.12 illustrate generated variations of the first and last sentence from the "Botched Training" Story from PersonaBank with abbreviated parameters used in the final column. The first sentence of the story, shown in Table 5.11, can be deaggregated into two clauses (*Anne excitedly entered PF Changs* and *The manager wanted to train Anne*). Id's 1, 3, 5, and 8 have different aggregation constructions (parameters denoted as *deagg:becauseNS, deagg:soSN, deagg:becauseSN, deagg:N*), while 2 and 4 don't use deaggregation (*deagg:none*). Id 1 has the emphasizer "rather" (*emph:rather*), 4 and 9, acknowledgments (*ack:right?*, *ack:ok*), and in 7 an exclamation mark (*exclaim:true*). There are both variations in the first and third point of view (*pov:1* and *pov:3*). Each sentence has its own style and way of starting the story.

id	Variation	Parameters Used
1	I rather excitedly entered PF Changs because the manager wanted to train me	deagg:becauseNS, pov:1, down:rather
2	I excitedly entered PF Changs in order for the manager to train me	deagg:none, pov:1
3	The manager wanted to train me, so I excitedly entered PF Changs	deagg:soSN, pov:1
4	Ok, I excitedly entered PF Changs in order for the manager to train me, right?	deagg:none, pov:1, ack:ok, ack:right?
5	Because the manager wanted to train me, I excitedly entered PF Changs	deagg:becauseSN, pov:1
6	The manager wanted to train Anne, so she excitedly entered PF Changs, as it were	deagg:soSN, pov:3, emph:as_it_were
7	Because the manager wanted to train Anne, she excitedly entered PF Changs!!	deagg:becauseSN, pov:3, exclaim:true
8	Anne excitedly entered PF Changs	deagg:N, pov:3
9	Essentially, ok, the manager wanted to train Anne, so she excitedly entered PF Changs	deagg:soSN, pov:3 emph:essentially, ack:ok
10	Actually, Anne excitedly entered PF Changs in order for the manager to train her	deagg:none, pov:3, emph:actually
11	The director wanted to train Anne, so she excitedly entered PF Changs	deagg:soSN, pov:3, synonym:true
12	The manager wanted to train me, so I excitedly entered PF Changs, okay?	deagg:soSN, pov:1, tagQ:okay?

Table 5.11: Variations of first sentence of *Botched Training* Story

Because there is no quoted speech, these utterances are defined as evoking Narrator Voice.

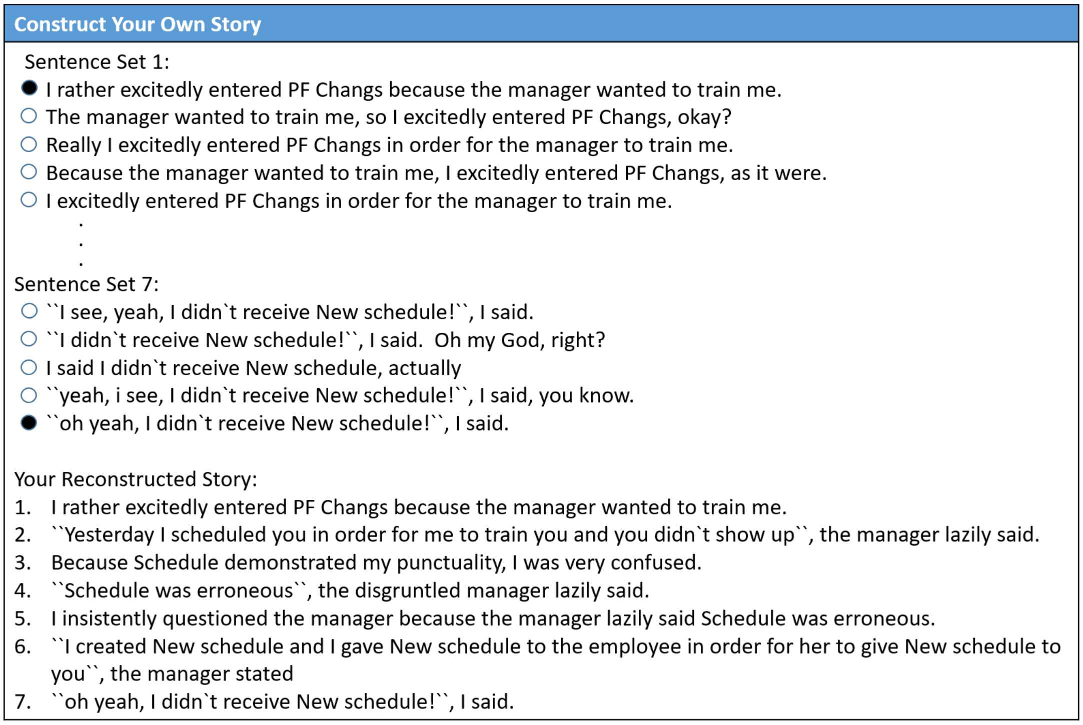
The sentence in Table 5.12 supports the quoted speech parameter (*qs:true*): see id's 13, 15, 16. Id's 13 and 15 include pragmatic features for Character Voice in quoted speech, compared to the Narrator Voice without the quotation marks (14, 21, with *qs:false*). This sentence is not possible to deaggregate because it contains one clause.

id	Variation	Parameters Used
13	“I see, yeah, I didn’t receive New schedule!”, I said	pov:1, qs:true, ack:i_see, ack:yeah, contraction:true
14	I said I didn’t receive New schedule, actually! !	pov:1, qs:false, exclaim:true, contraction:true
15	“Oh yeah, I didn’t receive New schedule!”, I said	pov:1, qs:true, ack:oh_yeah, contraction:true
16	“Well, obviously, I didn’t receive New schedule!”, I said	pov:1, qs:true, ack:well, comp_mit:obviously, contraction:true
17	“Oh God oh I didn’t receive well, New schedule!”, I said	pov:1, qs:true, ack:oh_god, ack:oh, contraction:true
18	I said I did not receive New schedule	pov:1, qs:false, contraction:false
19	“I did not receive New schedule!”, Anne said	pov:3, qs:true, contraction:false
20	“Oh I see, I didn’t receive New schedule!”, Anne said	pov:3, qs:true, contraction:true
21	“Oh yeah, I didn’t receive New schedule!”, Anne said	pov:3, qs:true, ack:oh_yeah, contraction:true
21	“Yeah, well, I didn’t receive New schedule!”, Anne said	pov:3, qs:true, contraction:true, ack:well, ack:yeah
21	Anne said she didn’t receive New schedule	pov:3, qs:false, down:rather, contraction:true
21	Anne said did not receive New schedule, actually	pov:3, qs:false, contraction:false, ack:actually
22	“yeah, I see, I didn’t receive New schedule!”, I said, you know	pov:1, qs:true, ack:yeah, ack:i_see, emph:you_know, contraction:true

Table 5.12: Variations of last sentence of “Botched Training” Story .

5.3.2 Overgenerate: Construct Your Own Story

Despite the hundreds of sentences and combinations that Fabula Tales’ narrative parameters can generate for any story, there is no guarantee that a naïve combination will produce appropriate narrative flow. To discover combinations of variation parameters that create engaging narrative flow, we designed an overgenerate and rank evaluation paradigm on Mechanical Turk that allows subjects to select from a subset of these generated variations to construct their



Construct Your Own Story

Sentence Set 1:

● I rather excitedly entered PF Changs because the manager wanted to train me.
○ The manager wanted to train me, so I excitedly entered PF Changs, okay?
○ Really I excitedly entered PF Changs in order for the manager to train me.
○ Because the manager wanted to train me, I excitedly entered PF Changs, as it were.
○ I excitedly entered PF Changs in order for the manager to train me.

.
.
.

Sentence Set 7:

○ ``I see, yeah, I didn`t receive New schedule!``, I said.
○ ``I didn`t receive New schedule!``, I said. Oh my God, right?
○ I said I didn`t receive New schedule, actually
○ ``yeah, i see, I didn`t receive New schedule!``, I said, you know.
● ``oh yeah, I didn`t receive New schedule!``, I said.

Your Reconstructed Story:

1. I rather excitedly entered PF Changs because the manager wanted to train me.
2. ``Yesterday I scheduled you in order for me to train you and you didn`t show up``, the manager lazily said.
3. Because Schedule demonstrated my punctuality, I was very confused.
4. ``Schedule was erroneous``, the disgruntled manager lazily said.
5. I insistently questioned the manager because the manager lazily said Schedule was erroneous.
6. ``I created New schedule and I gave New schedule to the employee in order for her to give New schedule to you``, the manager stated
7. ``oh yeah, I didn`t receive New schedule!``, I said.

Figure 5.10: Construct Your Own Story experimental design (sets 2-6 omitted for space)

own story. This subset is selected to showcase each feature in at least one sentence in a variety of combinations with other features.

Figure 5.10 shows a screenshot of the experimental design. For each sentence in the story, we present five different versions from which subjects select the sentence they like best. At the bottom of the experiment, the evolution of the reconstructed story is shown dynamically so subjects can read the full story to see how it flows. Subjects are encouraged to select and read each sentence within the context of the entire reconstructed story. At any time, subjects may select a different sentence, yielding a dynamic update to the experiment webpage. When finished, the subjects provide detailed feedback about why they selected the sentences they did.

Participants were familiar with reading and comprehension tasks, such as story sum-

Variant 1	Variant 2	Variant 3
Anne excitedly entered PF Changs in order for the manager to train her.	Because the manager wanted to train Anne, she excitedly entered PF Changs!!	I excitedly entered PF Changs in order for the manager to train me.
“Yesterday I scheduled you in order for me to train you and you didn’t show up”, the manager lazily stated.	“Yesterday I scheduled you in order for me to train you and you didn’t show up”, the manager lazily stated.	“Yesterday I scheduled you in order for me to train you and you didn’t show up”, the manager lazily said.
Anne was really confused because the schedule showed her punctuality.	Anne was confused.	I mean, I was confused because the schedule demonstrated my punctuality.
“The schedule was obviously, erroneous”, the disgruntled manager lazily said.	The rather disgruntled manager lazily said the schedule was erroneous.	“The schedule was erroneous”, the disgruntled manager lazily said.
Anne insistently questioned the manager because the manager lazily said the schedule was erroneous.	Right, the manager lazily said the schedule was erroneous, so Anne insistently questioned the manager.	I insistently questioned the manager because the manager lazily said the schedule was erroneous.
“I created the new schedule and I gave the new schedule to the employee in order for her to give document to you”, the manager melodramatically said.	“I created the new schedule and I gave the new schedule to the employee in order for her to give the new schedule to you”, the manager melodramatically said.	“I created the new schedule and I gave the new schedule to the employee in order for her to give the new schedule to you”, the manager stated.”
“I did not receive the new schedule!”, Anne said.	“Oh I see, I didn’t receive the new schedule!”, Anne said.	“I didn’t receive the new schedule!”, I said. Oh my God, right?

Table 5.13: Three stories constructed by annotators for “Botched Training” Story

marization, sarcasm detection, and sentiment detection. Nine subjects created thirty reconstructed stories on Mechanical Turk. Three reconstructed variants of the *Botched Training* story are in Table 5.13. Showing the reconstructed story at the bottom of the experiment allowed subjects to engage with their own perceptions of the flow of the narrative. Many subjects kept consistency, for example, using quoted speech consistently throughout the story.

5.3.2.1 Statistical Analysis

We rank the features across stories in the selected sentences to observe general trends. We define a **Category-Ratio** metric, the percentage of the time this particular feature is contained in a sentence selected by annotators with respect to the other features in each category:

$$catRatio_i = \frac{sel_i}{sel_{i.Total}} \tag{5.1}$$

where i is an item in a feature category and $i.Total$ is the entire feature category. For example, $i = Ack.True$ indicates the times the acknowledgment feature is used. From Table 5.14, $sel_{Ack.True} = 29$ indicates that an acknowledgment was in the selected sentences 29 times. $sel_{Ack.Total}$ is 217 which means that acknowledgments could have possibly been in 217 sentences. Thus, $catRatio_{Ack.True} = \frac{29}{217} = 0.13$, indicating 13% of the sentences selected had an acknowledgment, whereas, $catRatio_{Ack.False} = 0.87$ means that the other 87% of the selected sentences do not have an acknowledgment.

In the *Ack* category in Table 5.14, 29 are selected, calculated as a catRatio$_{ack}$ of 13%, i.e. 87% of the selected sentences did not contain an *Ack*. Of those that were selected, the highest scoring Acknowledgment is “yeah” selected 14 times, contributing to 6% of the total usage of Acknowledgment. In the *Competence mitigation* category, the catRatio$_{cm}$ is 1%, meaning that 99% of selected sentences did not have a competence mitigation. The only times it was selected was with the “obviously” hedge, selected 3 times. The only *Down* selected was “rather”, and appeared 17 times out of 217 possible times, contributing to a catRatio$_{down}$ of 8%. A variety of features in the Emphasizer (*Emph*) category were used, contributing to a catRatio$_{emph}$

Acknowledgments	*sel*	*catRatio*
{great, okay?}	0	0
{oh my god, ok, well, oh yeah}	1	< .01
{I see, oh, right}	3-4	.01-.02
yeah	14	.06
Acks True	29	.13
Acks False	188	.87
Acks Total	217	-
Competence mitigations	*sel*	*catRatio*
come on	0	0
obviously	3	01
Mitigations True	3	.01
Mitigations False	214	.99
Mitigations Total	217	-
Down	*sel*	*catRatio*
{I mean, like, somewhat}	0	0
rather	17	.08
Down True	17	.08
Down False	200	.92
Down Total	217	-
Emphasizers	*sel*	*catRatio*
{as it were, basically, essentially, obviously}	0	0
{great, very, you know, especially}	1	< .01
{actually, really}	9-11	.04-.05
Emph True	26	.12
Emph False	191	.88
Emph Total	217	-
Tag Questions	*sel*	*catRatio*
For real?, You're kidding, No?	0	0
Tag False	217	1
Exclamation True	50	.40
Exclamation False	74	.60
Contraction True	29	.85
Contraction False	5	.15
Deaggregation True	67	.74
Deaggregation False	23	.26
Quoted speech True	40	.69
Quoted speech False	18	.31

Table 5.14: Overgenerate Feature Categories from Fabula Tales

of 12%. “Really” was the highest selected emphasizer, selected a total of 11 times, and other high scoring words include “actually”, “especially”, and “very”. Emphasizers are not unique to direct speech, so it seems appropriate that it is used a lot. The final pragmatic insertion feature is the *Tag Question* category. However, no tag questions were in the selected sentences.

It is not surprising that subjects thought the dialogue parameters, such as tag questions, were not as appropriate in the monologic sense. Many of the pragmatic markers rarely or never appear in the selected sentences, including “I mean”, “come on”, “like”, and “no?”. Placement is an issue because these stories were not generated with any story-wide constraints. Popular features were the acknowledgment “yeah” and the emphasizers “really”, “very”, and “actually”. The competence mitigation toner is rarely used, and the tag question category is never selected. Fabula Tales also supports contractions with a $catRatio_{Contr.True}$ of an overwhelming 85%, which contributes to the flow and naturalness of everyday speech, regardless of narration or quoted speech. For sentences with exclamations, $catRatio_{Exclam.True}$ is 40%. Fabula Tales supports synonym replacement, however we see a $catRatio_{syn} = 5\%$. Upon closer inspection of which words are being replaced, we find that some words are being replaced with the incorrect sense. Fortunately, our subjects can recognize that, for example “defame” is not in the synonym set for “smear” in the sense of “smear the bug”. For the remainder of this experiment, we exclude variations that have an incorrect sense. Table 5.14 also shows that $catRatio_{Deagg.True} = 0.74$, that 74% of selected sentences have a deaggregation option. There is also an overall preference for quoted speech, with a $catRatio_{QS.True}$ of 69%.

Pragmatic marker usage in each constructed story, indicated as *perStoryRatio*, are shown in Table 5.15. A $perStoryRatio_{ack_0}$ of 0.5 indicates that 50% of the constructed sto-

perStoryRatio	0	1	2	3	4	5
Ack	.50	.33	.17	0	0	0
Comp	.96	.03	0	0	0	0
Down	.77	.23	0	0	0	0
Emph	.73	.27	0	0	0	0
Exclaim	.03	.47	.20	.06	.17	.06
Total	.60	.27	.07	. 01	.03	.01

Table 5.15: Ratio of Voice Features Per Story

ries do not use any acknowledgments. 33% of the constructed stories contain exactly one acknowledgment, 17% have exactly two, and none of the constructed stories use three or more acknowledgments.

While a high percentage of stories constructed did not contain any pragmatic features, about 30% had at least one pragmatic marker but only 7% of the stories had two pragmatic insertions per category per story. The same insertion feature is never used twice in one a story. We interpret this as there being a limited number of pragmatic markers being associated with voice, and that too many of these markers overwhelm the story. Exclamations marks have a different trend: half of the stories have at least one exclamation, and there are even several stories with 5 exclamations.

We make two general observations from the qualitative feedback:

- Annotators tried to create stories with a good flow and consistency;
- Features as character voice are pragmatically odd in many cases

This statistical analysis gives us insight into which features are selected, how often, and how we can use the high ranked features as information to Fabula Tales in generating future stories. Deaggregation, quoted speech, and contractions are popular and used consistently throughout a story. However, pragmatic features for character voice in quoted speech or narrator voice are not used as consistently because some are pragmatically odd, not taking context into account at the individual sentence level or across the entire story. For example, the production of the phrase *"I didn't receive the new schedule!", I said. Oh my God, right?* does not place the "Oh my God, right?" into the quoted speech, where it is more likely to fit. Furthermore, the combination of "Oh my God" and "right?" are a bit excessive for a single utterance, and could benefit from a better selection algorithm, as we discuss in the next section.

5.3.3 Rank: Evaluation of Complete Stories

We next refine the generated sentences as informed by the ranking statistics to test whether general narratological parameters lead to more engaging and preferred stories, achieving a balance between the preferred features and how often they appear in a story. Recall our hypotheses: first person is more engaging than third (H_1), quoted speech is preferred (H_2), pragmatic and lexical choice features may be used in quoted speech to support character voice (H_3), and deaggregation is preferred for producing natural sounding stories (H_4).

We first show support for our four hypotheses through ablation tests. Next we show that the statistics from subject ranking feedback make a significant impact on story preference. We then compare the stories generated according to our hypotheses against stories generated with random sets of parameters, to achieve a balance between randomness and ranking. Finally, we test the random stories against a baseline, to see if some random variation is better or worse than none. We create four sets of stories all informed by the statistics in the previous section:

- **Hypothesis**: Stories use sentence planning features from Fabula Tales: first person point of view, direct speech; a variety of deaggregation and pragmatic markers and lexical choice; and a *perStoryRatio* between 1-2 per story.

- **Random**: Stories have a mix of Fabula Tales features but are not balanced or consistent as in Hypothesis stories.

- **Baseline**: Stories have the most simple Fabula Tales baseline realization.

We use features where $catRatio_i > 0.01$, corresponding to a feature that was selected more than 1 time. Using these full stories, we create a new Mechanical Turk experiment comparing pairs of these stories across and within each category. Each pair of stories is annotated by seven subjects who were pre-qualified by participating in the "Construct Your Own Story" experiment.

5.3.3.1 Hypotheses vs. Baseline

We generate a set of eleven Hypothesis stories, turning on and off each parameter or testing combinations of parameters. We compare each story to a Baseline story generated without any parameters. The Hypotheses stories generated by the rank statistics are preferred with Cohen's $\kappa = 0.75$, a high measure of interannotator agreement.

$\mathbf{H}_1$ states that stories in first person point of view are preferred because they are more engaging and personal. We perform an ANOVA with a Bonferroni correction and find point of view is not significant ($F(1,200) = 3.62, p = 0.06$). One subject observes that first person can lead to more opportunities for the character to describe his feelings. By entering the mental state of characters, we create more engaging stories, and the first person point of view parameter allows us to do this. Another subject identified that the first person story added a "sense of immediacy" and "tension". This sense of "being there" mimics Biber's observation about personal pronouns.

However, we find that some subjects prefer the third person. This is in direct contrast to our previous observation about point of view from Section 5.2.2. We believe that this shift occurs because in this set of experimentation, we replace "the narrator" with the name of the character, such as "Anne". As we learned from our previous experiments, "the narrator" was seen as unfavorable, and because Fabula Tales can support naming characters, we added support for more natural style of narration, whereas, recall, the WYSIWYM cannot do this. We can use this information to offer different readers different perspectives of a story according to personal preference.

H$_2$ claims that quoted speech is preferred. We find that quoted speech is preferred ($F(1,200) = 22.38, p < 0.001$). Readers commented that the alternating narrative description and dialogue adds more personality to the story, and that it allows more "back and forth banter" between the characters in the story.

H$_3$ claims that pragmatic markers and lexical choice are effective as character voices when they are used within quoted speech. We find that these features are preferred ($F(1,200) = 13.2, p < 0.001$) and that there is an interaction between quoted speech and pragmatic and lexical features ($F(2,200) = 11.98, p < 0.001$). We examine in greater detail the interactions. First we compare *Quoted Speech Only* and *Narrator Voice* excerpts in Table 5.16. *Quoted Speech Only* is preferred with $\kappa = 1.0$. Subjects say *Quoted Speech Only* is easier to understand. Comparing stories with a *Narrator Voice* to stories with *Character Voice*, *Character Voice* stories

Quoted Speech Only	“The schedule was erroneous”, the manager said. Anne questioned the manager because the manager said the schedule was erroneous. ... “I didn’t receive the new schedule”, Anne said.
Narrator Voice	The manager said the schedule was obviously, erroneous. Anne questioned the manager because the manager said the schedule was erroneous. ... Yeah, Anne said she didn’t receive the new schedule.
Character Voice	“The schedule was obviously, erroneous”, the manager said. Anne questioned the manager because the manager said the schedule was erroneous. ... “Yeah, right, I didn’t receive the new schedule”, Anne said.

Table 5.16: Excerpts of different speech conditions in ablation test

are preferred (Cohen’s $\kappa = 0.95$). Subjects preferred *Character Voice* because of its use of conversation. The conversation breaks the story up into easily understandable parts, and makes the story easier to follow.

However, comparing *Character Voice* stories with *Quoted Speech Only* stories, we observe Cohen’s $\kappa = 0.71$ with preference for *Quoted Speech Only*. One subject comments that *Character Voice* with pragmatic and lexical choice features acting as voice yields “mixed results”. Annotators comment on what might be an issue of context insensitivity. For example, in *Character Voice* in Table 5.16, the generated text includes *Yeah, right* as part of what Anne, the narrator, says in the quoted speech. Subjects say this creates a tone not appropriate with respect to Anne’s tone in the rest of the story. This offers further evidence that characters’ voices have an effect on how readers perceive them. While these voice and style features are more acceptable as *Character Voice* than as *Narrator Voice*, the *Character Voices* must take context or personality of the character into account to be pragmatically cohesive.

$\mathbf{H}_4$ claims deaggregation is preferred and an ANOVA shows this is true ($F(1,200) =$

Pair	Hypothesis	Baseline
1	Because the bugs scared John, he grabbed the rolled comic book.	John grabbed the rolled comic book because the bugs scared him.
2	The squirrel fell over the deck's railing.	The squirrel fell over the deck's railing because the squirrel leaped because the squirrel was startled.
3	The squirrel was startled, so the squirrel leaped.	The squirrel leaped because the squirrel was startled.

Table 5.17: Excerpts of different deaggregation condition in ablation test

$10.68, p < 0.01$). Subjects prefer to hear the emotional response of the story characters before hearing the action they take. For example, stories with sentences of the following construction: *Because the bugs scared John, he grabbed the rolled comic book.* are preferred over the baseline, *John grabbed the rolled comic book because the bugs scared him.* Deaggregation also creates cleaner and crisper stories, preferred because they are shorter compared to the baseline stories, but still get the point across (Table 5.17).

5.3.3.2 Hypotheses vs. Random

We compare stories generated according to the rank statistics vs. stories created with random parameters. For example, in a rank controlled story, an acknowledgment may appear at most two times in Character Speech, whereas, in a random story, the same acknowledgment may appear four times in Character Speech. Ten story pairs in this condition are annotated by the same seven Mechanical Turkers. Hypotheses rank statistics stories were preferred with Cohen's $\kappa = 0.92$. The random condition does not yield consistent parameters used within the story, such as quoted speech, and uses more pragmatic and lexical features than preferred. Subjects find stories constructed using specific statistical models are easier to read and understand. We conclude that the statistical analysis of how often features are selected in the ranking experiment (*catRatio*) and the number of times a feature is used per story from (*perStoryRatio*) do have an effect on preference.

5.3.3.3 Random vs. Baseline

Finally, we compare the stories with random parameters against baseline stories with no variation. Six pairs of stories are in this condition, annotated by the same seven Mechanical Turkers. The baseline stories were preferred with Cohen's $\kappa = 0.96$. Annotators noted that no variation was better than poor variation, even if the baseline stories were a little dry. In one pair, the random story contained quoted speech, a high ranked feature, with pragmatic and lexical choice features in the quoted speech. However, annotators found the dialogue to be stilted and did not contribute to a good character voice, and so they therefore preferred no variation. In the cases where an annotator preferred the random story, it was because the baseline was too dry.

These experiments show that we need to achieve a balance between bland stories without any variation, and stories with too many pragmatically odd features due to placement ignoring the narrative context. From our findings, a combination of statistical analysis with our narratological hypotheses create well-balanced stories.

5.4 Summary

The EST and Fabula Tales creation began with an evaluation of the semantic-to-syntactic algorithm using a Fabula Tales baseline realization against the WYSIWYM in an attempt to match the realization without embellishments or narrative content or sentence planning parameters. Section 5.1 first validated that the baseline story is reproducible by the EST. The focus of evaluations shifted in Section 5.2 to apply selected narrative variations of Fabula Tales at the sentence level, showing our narratological parameters are effective and preferred across a variety of blogs using metrics of interest in the storytelling domain. Experiments show these narratological parameters lead to different perceptions of the story, and we find evidence that there are significant differences in reader's interest and engagement in a story dependent only upon the style.

We make use of the overgenerate and rank method to explode variations of generated stories, showcasing the flexibility of the Fabula Tales system, then have subjects sift through the generations to identify the best utterances and associated parameters in the context of the flow of an entire story. The evaluation paradigm learns statistics of features associated with sentences selected in a story reconstruction experiment. The story preferences from Mechanical Turkers tell us which narrative variations produced by the Fabula Tales system are effective. By using the Fabula Tales framework to combine different parameters within a single story, subjects match the ones they believe contribute to the best flow.

Generating new stories with the learned features and performing ablation tests show strong support for H_2 (a strong correlation between engagement and quoted speech) and H_4 (a strong correlation between engagement and deaggregation) across all stories without having to pay attention to the story context. Character Voice is better than Narrator Voice for pragmatic marker insertion and lexical choice as voice (H_3), but there are still problems with pragmatically odd variations, e.g. *Yeah, right* in Character Voice from Table 5.17. We find that point of view preference is not significantly correlated to engagement in complete stories (H_1), allowing us to create different profiles based on readers. Finally, we show that stories constructed from statistics from the annotator ranking evaluation are more preferable than randomly constructed stories, and that poor random variation is worse than none at all.

In future work, we propose new models be built for Fabula Tales incorporating context into parameter selection to avoid the pragmatically odd placements of pragmatic insertions in Character Voices. The WordNet and VerbNet ontologies integrated into the SIG representation can be used to learn a discourse model for domain specific information that could be used in retelling. Furthermore, insights into the emotions of characters could be useful for creating context appropriate emotional character voice reactions. We discuss preliminary work of a discourse planner in Section 6.2.1 and adapting emotional voices to the context in Section 6.3.1.

Previous work uses machine learning to learn features from the overgeneration phase associated with preferred utterances. We learn over the selected and not selected sentences from the "Construct Your Own Story" experiment. Our best model had an f-score of 0.83 on the not-selected class. Even though this result seems high, the most informative features overfit, suggesting the training data was too small for learning trends with machine learning methods. We offer a more detailed discussion and analysis in Section 6.2.2.

Chapter 6

Conclusion

This thesis presents a framework to bridge the NLG gap in a domain independent storyteller by modeling story semantics derived from the SIG representation as new semantic-syntactic structures compatible with syntactically flexible and semantically preserving DSYNTS and text plans. From a single *fabula*, the EST and Fabula Tales generates many different *sujet*, playing with narrative variations through content planning and sentence planning NLG manipulations that emulate stories told in the wild to achieve different effects on a particular audience.

Section 6.1 summarizes the contributions presented in this thesis. Section 6.2 discusses the limitations and proposed improvements to the EST and Fabula Tales framework. The future work we have alluded to throughout this thesis, including generating emotional text, temporal manipulation, and focalization, is elaborated upon in Section 6.3. We conclude with a detailed discussion how the EST and Fabula Tales can directly integrate with interactive narrative systems, virtual agents, and be used for dialogue authoring (Section 6.4).

6.1 Conclusions and Contributions

6.1.1 The Expressive-Story Translator and Content Planning

The first contribution of this thesis bridges the NLG gap by creating a semantic-syntactic mapping allowing the system to represent diverse story content and generate rich language. We design the Expressive Story Translator (EST) as the translation between the narrative representation of SIGs to the language representation of DSYNTS and text plans.

The SIG's robust story representation makes it an ideal candidate for a content pool. Its domain independent formalism and its vocabulary hooks into WordNet and VerbNet are beneficial to create our multi-domain NLG framework (Section 3.1.1). However, the WYSIWYM generation engine for annotator feedback does not allow for expressive language, but only the intended ground truth *fabula* of the annotation.

The parameterizable structure of DSYNTS can induce different styles through content planning and sentence planning (Section 3.1.2). However, alone, DSYNTS do not store contextual information about the entire narrative. This lack of a discourse planner cannot ensure a fluid story telling over multiple sentences. Furthermore, hand authoring new content is tedious and time consuming, and new methods for DSYNTS creation have not yet been explored.

The EST is the first end-to-end pipeline making use of DSYNTS with automatic tools for their creation. The EST translates between the SIG formalism and DSYNTS by applying a theory of syntax and semantics to the SIG (Section 3.2). By defining classes loosely associated with parts of speech, syntactic information is captured, and the underlying structure can be changed in a sentence planner. Semantic information from SIGs are encoded, including rhetorical relationships that describe narrative information.

The EST is more than a one-to-many syntactic mapping; it is a new model that maintains both semantic and syntactic story information providing for narrative content planning and manipulation of the *fabula* by mining the interpretation layer for patterns for appraisal modeling and temporal manipulation (Section 3.3). Non-narrative specific content planning operations can be derived from the semantics encoded during the EST transformation.

Other narrative systems that implement content planning, especially temporal manipulation or story generation, often represent data as text that can be mined (e.g. Say Anything [Swanson and Gordon, 2008]), or a set of tuples with story events, character beliefs, and goals connected by paths of their relationships (e.g. Scheherazade [Li, 2015]). We have not fully explored narrative content generation, the hallmark these systems and others such as Tale-Spin [Meehan, 1977], and include a discussion of how the EST can implement a more sophisticated temporal planner in Section 6.3.2.

6.1.2 Fabula Tales and Sentence Planning

The second contribution of this thesis is the Fabula Tales sentence planner. Building from the DSYNTS and text plans generated by the EST, Fabula Tales manipulates the syntactic structures to perform narrative sentence planning including changing point of view, inserting direct speech, and associate pragmatic markers and lexical choice with character voices. Fabula Tales further uses contextual information about the story semantics to perform aggregation and deaggregation (Section 4.1).

Fabula Tales implements similar narrative parameters as Callaway and Lester [2002]'s

Storybook narrative system. However, while other narrative systems are limited by their domain dependence, including Storybook and Montfort's Curveship [Montfort, 2007], Fabula Tales' narrative parameters can be applied to any syntactic representation of stories as DSYNTS.

Data-driven approaches for plot or prose generation require an enormous amount of training data [Li, 2015, Ritter et al., 2011, Sordoni et al., 2015, Swanson and Gordon, 2008, Vinyals and Le, 2015], and even so, sometimes have difficulty with personalization beyond generic responses. To overcome this, researchers often fine-tune their models and change their objective functions, requiring a deep understanding of the underlying model. On the other hand, other systems, such as Tale-Spin, do not automatically map the semantic representation to syntactic structures, but instead pull text from a series of hardcoded story points that have no underlying syntactics that can be modified once selected.

Parameterization of the syntactic DSYNTS produced automatically by the EST allows us to isolate certain stylistic aspects of language, such as freely interchanging character voice models inside direct speech. This affords more opportunities for personalization and customization. We further discuss how to build rich models for displaying emotion in generated text in Section 6.3.1 to be incorporated with the content planning appraisal we perform to make more engaging characters and increase tension.

6.1.3 PersonaBank and Domain Independence

The SIG's flexible content representation gives the EST and Fabula Tales the ability to model any story topic in this formalism. The annotation tool Scheherazade is available for annotators to create new content to our NLG framework, in any number of topics. To enhance the quality of the translator and to showcase the domain independence of the entire framework, we collect a corpus of over 100 personal narratives annotated with the SIG representation (Section 4.2). This corpus will be of general utility both for theoretical analyses of narrative structure and for applications related to storytelling and dialogue.

Previous works in narrative systems have often been restricted to a single domain, due to the complex input required to the system, such as the commonsense rules in Storybook's implementation of Little Red Riding Hood. Our approach allows for easier annotation of input stories using Scheherazade for creating SIGs, then the EST extracts semantic information from a new SIG and the rest of the NLG process is streamlined.

PERSONAGE and its descendants requires manual construction of content as DSYNTS, which is time consuming and does not easily yield content reusability for different domains. While authors have fine control over utterances, this technique for DSYNTS generation is time consuming and only skilled annotators and linguists are capable of such a feat in a short amount of time. If authors want a new domain or set of utterances, each one must be hand crafted.

Our EST translation methodology provides a trade-off in annotation time and difficulty: instead of requiring skilled experts to create DSYNTS, we require minimally trained annotators to create a SIG. This trade off is worthwhile and justifiable because SIGs can be annotated in under an hour (time trade-off), and it makes use of the existing lexical ontologies provided by WordNet and VerbNet, to allow for morphological diversity in retelling.

6.1.4 Building a Generation Dictionary

The ability for the EST and Fabula Tales to automatically map, plan, and generate story variations in a number of different domains leads to automatically creating or learning a generation dictionary. Rather than learn one dictionary for each story, as the EST does now, a compilation of translated stories and the semantics could be learned from the SIG, and associated translated semantics-syntactics could be learned from the EST. This would allow for reuse of previously seen story content and provide more syntactic variation with easy-to-author content and stylistic variation.

6.1.5 NLG System Evaluation

The third contribution of this thesis defines an evaluation paradigm for the entire NLG system by exploring interactions between the narratological sentence planning parameters and generated stories. But first, we ensure that the EST translation performance is close to the baseline WYSIWYM realization, which was derived directly from the semantics the EST models (Section 5.1).

Experimentation at the sentence level shows that the capability to apply narrative parameters is general and can be applied to all types of informal personal narratives (Section 5.2). Perceptual experiments show that these narratological parameters in Fabula Tales generations lead to different perceptions of the story, e.g. a retelling of “The Fox and the Crow” in conjunction with a first point of view and different voice styles and quoted speech leads to overall positive views of the Fox with mixed perceptions of the Crow. Further experimentation offers subjects different perspectives and points of view to test levels of interest, naturalness, and narrative immediacy. We discover that, at the sentence level, Fabula Tales generation using first person point of view, voice, and deaggregation are preferred over the baseline Fabula Tales and WYSIWYM.

The overgenerate and rank evaluation allows us to test and refine the generation and parameters by eliciting feedback from both humans and statistical analysis into the paradigm (Section 5.3). This gives way for additional narrative parameters compatible with the EST and Fabula Tales framework to be explored in order to increase story engagement (Section 6.2.1).

We hypothesize that the following general characteristics will always lead to increased engagement:

H_1: There is a significant correlation between first person point of view and engagement because it brings the reader into the immediacy of a story

H_2: There is a significant correlation between direct speech and engagement because direct speech allows characters to express themselves

H_3: There is a significant correlation between direct speech and certain styles and increased reader engagement because style features can be used in direct speech to act as character voice

H_4: There is a significant correlation between domain independent deaggregation and engagement because aggregation creates more natural sounding and concise stories

We use the overgenerate and rank approach to test our proposed hypotheses and evaluate the system. In the overgeneration phase, Fabula Tales' flexibility is shown by generating hundreds of sentence variations using different combinations of narrative sentence planning parameters. We propose and test a novel evaluation paradigm for overgenerate and rank where annotators "Construct Your Own Story" sentence-by-sentence by selecting from a subset of the generated sentences. Subjects study differences between the individual sentences and how they flow together in the context of an entire story. This new paradigm iteratively combines human and statistical feedback to build the most engaging and preferred models.

In the rank phase, statistical analysis ranks the features in the selected sentences. These results are used to generate complete stories on which we conduct ablation tests by comparing stories without any variation against stories with high ranking features to test the four hypotheses. There is strong support for quoted speech (H_2) and deaggregation (H_4) across all stories without having to pay attention to the story context. Character Voice is better than Narrator Voice for voice features (H_3), but there are still problems with pragmatically odd variations. Point of view preference is not significant (H_1), allowing us to create different profiles based on readers. Stories constructed from ranking statistics are more preferable than randomly constructed stories, and that poor random variation is worse than none at all.

6.2 Limitations of Framework

6.2.1 Discourse Planner

The EST and Fabula Tales boast a domain independent treatment of NLG, yet we find there are some instances where we must be cognizant of the content. We propose to incorporate context with a rule-based approach, building on the SIG's general SIG arcs or senses in WordNet or VerbNet, as well as by building on recent work on statistical models for expressive generation [Langkilde, 1998, Mairesse and Walker, 2011, Paiva and Evans, 2004, Rieser and Lemon, 2011, Rowe et al., 2008]. Some style features inserted in Character Voice utterances were not appropriate, e.g. *"Yeah, right, I didn't receive the new schedule", Anne said* from Table 5.16. The current model does not examine domain or story specific knowledge, but we hypothesize the SIG and appropriate WordNet and VerbNet ontologies and emotional valence and more semantic information can be used as a discourse model to learn domain specific information to influence the retelling. In some stories in the deaggregation experiment, we found two stories where preferences of subjects vary from the overall trend. These stories suggest future possible experiments where we might vary more aspects of the story context and audience [Petty et al., 1981].

Antoun et al. [2015] have already made use of SIGs as discourse planner. They use the SIG as an intermediate representation of meaning by transforming a play trace of the PromWeek game into a representation which can be used to generate natural language recaps of the game. Using the SIG as an intermediary allows them to keep the affordances of the SIG representation that can be used to make decisions about what language to generate in the play trace.

Our discussion of PERSONAGE in Section 3.1.2 mentions that its lack of a discourse planner does not support different voice and style parameters for multiple turns within a single story. An example full story generated from PERSONAGE is:

> I see, oh gosh I recently opened my patio's door! my patio's door! Oh well, the bugs entered my apartment! I see, yeah, I didn't initially notice for the bugs to enter my apartment! Oh gosh oh God I slept! Oh I overnight awoke!

This rendering does not keep track of what features were already used. To remove these redundancies in full stories, we use PyPersonage for our overgenerate and rank evaluations.

PyPersonage's discourse planner makes informed decisions about what and how many times to use these features across the entire story. Although PyPersonage is inspired by PERSONAGE, most of its parameters were optimized for casual dialogue. For example, the hedges trailing in the following sentences are much more expected to appear in a dialogue between two people co-telling a story, rather than in a single narrator: *The slimy bugs quietly entered my apartment ... are you sure?* and *The slimy bugs quietly entered my apartment, you're kidding, right?* While PyPersonage solves the contextual placement problem, i.e. one or two of the same style feature appeared in an entire story so we never get a telling like the PERSONAGE example above, we discovered that the new generator is not ideal for Fabula Tales' single narrator storytelling. To solve this, more features for individual style must be integrated into PyPersonage for it to be more effective, and parameterized models can be created distinctly for dialogic co-telling and for monologic storytelling.

6.2.2 Learning Features

Previous work uses machine learning to learn features associated with preferred utterances [Walker et al., 2007]. We treated the "Construct Your Own Story" as a classification problem at the sentence level on the sentences that were Selected and Not Selected by annotators in the experiment. We keep in mind that the selected labels might be a result of the least-worst sentences, but this would still yield a subset of better potential sentences. We use features inspired by Walker et al. [2007] with the following feature sets with a binary value:

- **N-gram features**: unigrams, bigrams, and trigrams
- **Punct features**: punctuation marks
- **Pragmatic features**: capture insertions of these parameters. Uses each sentence feature as a single feature in the feature vector. E.g. "ack:ok"
- **Pragmatic* features**: capture further semantic information about the context of these parameters. Also counts how many different categories of insertions are present.
- **Fabula Tales features** (FT): including deaggregation and narratological point of view and direct speech parameters

Our baseline for predicting True is 0.4 and False is 0.6. All models reveal they are good at predicting the False class, that is the sentences that will not be used. Our intuition is because certain key words are present, such as the pragmatic dialogic features inserted with PyPersonage, which as we previously discussed, do not make sense in the monologic storytelling setting (e.g. "are you sure?").

Table 6.1 shows our scores for the classification task using our best Weka model, SMO, and Table 6.2 shows a decision tree. We see that the **N-gram** feature set performs the best with a weighted average precision of 0.7, recall of 0.7, and f-measure of 0.7. On it's own, the **FT** features are not very informative, and in some cases, actually bring down the scores when combined with other feature sets. However, **FT** with **N-grams** only slightly hurts in performance. The **Pragmatic*** supplemented features do not perform any better than the original **Pragmatic** features.

	Precision			Recall			F-Measure		
Feature Set	True	False	Avg	True	False	Avg	True	False	Avg
ngram	0.52	0.79	0.70	0.55	0.77	0.7	0.53	0.78	0.70
punct	0.54	0.78	0.71	0.52	0.79	0.71	0.53	0.79	0.71
prag	0.25	0.68	0.55	0.07	0.91	0.64	0.11	0.78	0.57
ft	0	0.69	0.47	0	1	0.69	0	0.81	0.56
ngram-ft	0.49	0.78	0.69	0.55	0.74	0.68	0.52	0.76	0.68
ngram-punct	0.51	0.8	0.71	0.6	0.74	0.7	0.55	0.77	0.7
ngram-prag	0.52	0.8	0.71	0.57	0.76	0.7	0.54	0.78	0.7
punct-ft	0.54	0.8	0.72	0.57	0.78	0.71	0.56	0.79	0.71
prag-ft	0.3	0.69	0.56	0.07	0.93	0.66	0.11	0.79	0.58
prag-punct	0.54	0.77	0.7	0.48	0.81	0.71	0.5	0.8	0.7
ngram-prag-ft	0.49	0.78	0.69	0.54	0.74	0.68	0.51	0.76	0.68
ngram-prag-puct	0.51	0.8	0.71	0.59	0.74	0.7	0.55	0.77	0.7
prag-punct-ft	0.54	0.78	0.70	0.5	0.80	0.71	0.52	0.79	0.70
ngram-prag-punct-ft	0.52	0.8	0.71	0.59	0.75	0.7	0.55	0.77	0.71

Table 6.1: Classification with SMO in Weka

	Precision			Recall			F-Measure		
Feature Set	True	False	Avg	True	False	Avg	True	False	Avg
ngram	0.36	0.69	0.59	0.09	0.93	0.66	0.15	0.79	0.59
punct	0.54	0.78	0.71	0.52	0.79	0.71	0.53	0.79	0.71
prag	0	0.68	0.47	0	0.97	0.66	0	0.8	0.55
ft	0	0.69	0.47	0	1	0.69	0	0.81	0.57
ngram-ft	0.33	0.69	0.57	0.09	0.92	0.66	0.14	0.79	0.58
ngram-punct	0.67	0.81	0.76	0.55	0.88	0.77	0.6	0.84	0.76
ngram-prag	0.36	0.69	0.59	0.09	0.93	0.66	0.15	0.79	0.59
punct-ft	0.51	0.75	0.69	0.41	0.82	0.69	0.46	0.79	0.68
prag-ft	0	0.67	0.46	0	0.93	0.64	0	0.78	0.53
prag-punct	0.56	0.79	0.72	0.55	0.81	0.72	0.55	0.8	0.72
ngram-prag-ft	0.33	0.69	0.58	0.09	0.92	0.66	0.14	0.79	0.58
ngram-prag-puct	0.65	0.81	0.76	0.55	0.87	0.76	0.59	0.83	0.76
prag-punct-ft	0.6	0.81	0.74	0.57	0.82	0.74	0.58	0.81	0.74
ngram-prag-punct-ft	0.65	0.81	0.76	0.55	0.87	0.76	0.59	0.83	0.76

Table 6.2: Classification with J48 in Weka

Our best model had an f-score of 0.83 on the not-selected class, and 0.6 on the selected class. Even though these results seem high, the most informative features did not reveal general insights. For example, for one SMO classifier, the most informative positive features are (Right, 0.59), (door, 0.36), and (The, 0.29), (oh, 2.2). "door" and "The" are particular to individual stories, and indicate our model overfit to this data. For the decision tree, we get a very deep and narrow tree, again seemingly overfit to one particular story. We conclude that there is not enough diversity for applying a model to the test set for classification, and leave further machine learning on a larger collection of constructed stories to future work.

6.3 Future Work

6.3.1 Building Emotional Voice Models

Section 3.3.1 explores how we model character emotion in the EST framework using appraisal theories. We hypothesize that parameterizable voice models can be built from emotional studies, similar to and inspired by personality models in PERSONAGE and by character models in Lin [2016]. We predict these models will be effective based on own intuition and qualitative feedback from subjects in perceptual and overgenerate and rank experiments where subjects identified and preferred when the voice features matched the emotional value of the character. In these experiments, the overlap was by chance; now we aim to purposefully induce a matching tone.

We first posit that general, neurotic and emotionally stable models may be appropriate in the story context for expressing hopes and fears. Mairesse and Walker [2008] create

	Neurotic	Emotionally Stable
Traits	calm, even-tempered, reliable, peaceful, confident	neurotic, anxious, depressed, self-conscious, oversensitive, vulnerable
Generated Example	I am not really sure. Cent'anni is the only restaurant I would recommend. It's an italian place. It offers bad at-at-atmosphere, but it features like, nice waiters, though. It provides good food. I mean, it's bloody expensive. Err... its price is 45 dollars.	Let's see what we can find on Chimichurri Grill. Basically, it's the best.

Table 6.3: Neurotic and Emotionally Stable examples from Mairesse and Walker [2008]

Parameters	Neurotic	Emotionally Stable
Content polarity	low	high
Repetition	high	low
Verbosity	high	low
Softener hedges	low	high
Filled pauses	high	low
Emphasizer hedges really, actually	high	low
Emphasizer hedges basically, just	low	high
Acknowledgments	high	low
Tag questions	high	low
Expletives	high	low
Stuttering	high	low

Table 6.4: Neurotic and Emotionally Stable models from Mairesse and Walker [2008]

parameterizable models for neuroticism and emotional stability. Table 6.3 shows the Big Five traits and generated output from PERSONAGE and Table 6.4 shows the PERSONAGE parameter models. We envision fear paired with a neurotic voice, exhibiting high repetition, verbosity, and filled pauses.

Many psychology studies on speech further inspire emotional voice models. Eichler [1965] surveys Weintraub and Aronson [1963]'s general categories of speech that are broken down into more detailed speaker mental states. Speech characteristics exhibited in anxiety use a high degree of denial [Weintraub and Aronson, 1964], psychotic speech includes hostile, guilt evoking thoughts exhibited through high negators [Weintraub and Aronson, 1965], and speech associated with depression including intolerable sadness through nonpersonal reference and expression of feeling, and guilt through high evaluators [Weintraub and Aronson, 1967]. Furthermore, speech disruption cues displayed include breaks, corrections, and stuttering repetitions [Dibner, 1956, Krause and Pilisuk, 1961, Mahl, 1956].

From these surveys, we propose emotion models for anger, such as implementing anxiety features such as intrusion, a nonverbal sound that breaks speech, e.g. a cough, sigh, or laugh if the subject becomes threatened; a high verb/adj ratio, a particular speech habit that reveals a deep personality traits [Collier, 2014]; higher emotions and instability; and negation, indicative of retraction.

We propose a series of experiment to test these voices "in action" by generating appropriate emotionally charged text to accompany each appraisal story point. The first objective would query subjects on the Ten Item Personality Test [Gosling et al., 2003] to determine if they can identify the emotional language within a story context when reading an entire story either in third person with direct speech in a neurotic or stable voice, or in the first person in a neurotic or stable voice.

The second evaluation objective examines if subjects can select the most appropriate emotional language via style. Given a story excerpt with the neutral model, we would ask subjects "based on your understanding of the story so far, which next sentence is the best fit/aligned with what the character is feeling?" and present utterances in the character voice in a neurotic, stable, or neutral voice compatible with the emotional state of the story point.

A potential critique of this work comes from Scherer and Ceschi [2000]: "One can rarely generalize from a single case to the general process of emotion expression". However, Busemann [1925] provides a counter argument: "these style differences depend very little upon the subject matter dealt with" giving substantial support for our domain independent framework and that these voices may be able to generalize across domain and topic, and be used based on the emotion traces derived in the EST.

6.3.2 Focalization and Temporal Manipulation

We discuss in Section 3.3.2 that the EST models the temporal ordering of story points in a linear fashion, and that the goals and arcs from the interpretation layer are preserved in the translation. The work of Bae et al. [2011] explicitly models character focalization in a library of plans with character's use of operators and preconditions on their goals, essentially a knowledge base of goals and perceptions from each character. We posit that the EST does have these character goals and perceptions through the SIG interpretation layer.

However, as mentioned in Section 3.3.2, the EST does not currently have an additional focalization model to track the reader's attention for evoking narrative effects, such as suspense. Bae and Young [2009] creates stories with surprise endings according to Structural Affect Theory, which is used to rearrange the order of events to elicit specific emotions to the reader. They use a planner to model the attention of the reader's knowledge compared to the knowledge of the characters in the story.

We propose similar attention models could model the reader knowledge in the EST framework. At each story point in the temporally linear story, we would apply a model of focalization for each character based on VerbNet and WordNet agent-patient frames, giving the EST access to what information characters know. This could be used to generate different framing based on a focalizing character. Recall the Tale-Spin story about Wilma and George; the plot driving the story forward is Wilma's goal. Applying a model of focalization could remove George in Wilma's focalization. These focalization models could be extended to track the listener knowledge at each story point and used for suspense modeling and story generation.

6.4 Applications of Framework

The bridging of the semantic-syntactic gap in NLG systems allows for a variety of applications making use of the new EST domain independent semantic-syntactic modeling, and Fabula Tales' sentence planning narrative parameters. We explore several areas where our framework can be integrated out-of-the-box to enhance systems similar to those we discussed in Chapter 2, including virtual agents, interactive narrative systems, and dialogue authoring.

6.4.1 Virtual Agents

Personalization is a critical feature of natural interaction, and there is growing evidence that virtual agents that deliver personalized interaction are more effective. Virtual agents that match the user's personality have been shown to help the user spend more time doing their exercises [Tapus and Mataric, 2008], or are judged as more competent at the task [Cassell and Bickmore, 2003, Isbister and Nass, 2000].

Furthermore, embodied virtual agents lead to even more engagement [Broekens et al., 2012, Swartout et al., 2010], especially by coordinating gestures with speech to increase the naturalness of human like communication [Bergmann et al., 2013, Wang and Neff, 2013]. The Fabula Tales framework could easily be integrated into a virtual agent environment, as such, Hu et al. [2016] takes advantage of the domain independent affordances of Fabula Tales by rendering dialogue from the EST between two virtual agents complete with gestures and placement annotations.

To further integrate the telling with the story content, the virtual agent should take advantage of the emotional information the EST can extract at each story point to match their voice and style with the context and tone of the story. Acoustics can be used for exhibiting emotions in a context free manner. Cahn [1989] explores producing affect through synthetic speech by using lexically emotionally neutral statements and applying different treatments of acoustic cues. Scherer and Oshinsky [1977] and Cahn [1989] find cues associated with specific emotions. For example, happiness has a large pitch variation, down pitch contour, low pitch level, and fast tempo, while anger on the other hand, has small pitch variation, up pitch contour, high pitch level, and fast tempo. They find other general cues associated with fear and sadness. Despite this work on observing and isolating characteristic of voice, there are no publicly available systems that can use these cues to generate an angry voice model. We propose the EST and Fabula Tales framework could easily make use of such a voice system.

Lexical choice can of course have an impact on how responses are interpreted felt by the reader [Snow et al., 2008, Strapparava and Mihalcea, 2007]. Fleischman and Hovy [2002a] examine slants and agency selection, discussing verb, adverb, and adjective selection based on agent affect or whose side you're on. In the sentence *The driver collided with the mother*, the driver is the agent and the mother the patient, whereas in *The mother was hit* there is no agent. This work seeks to understand previous work in psychology literature on cues for emotions and anxiety in speech that are context independent. Our own experiments have confirmed that people do change their perceptions of characters based on how they speak, and hypothesize that our framework can be integrated with a multi-modal virtual agent.

6.4.2 Interactive Narrative Systems

The EST and Fabula Tales framework can enhance narrative systems as well as make them more customized to the user. Interactive narrative systems are effective for education, training and persuasive applications involving the user's motivation. Interactivity engages the user's sense of agency and investment in story outcomes [Kelso et al., 1993, Read et al., 2006]. Narrative structures are more persuasive than other formats of argumentation: telling the listener a personal story is a more effective means to change beliefs. Incorporating narrative into interactive learning environments elicits different student experiences than traditional education, and can affect intrinsic motivation, self-efficacy beliefs and positive affect [Malone and Lepper, 1987, Mcquiggan et al., 2008].

An existing narrative learning system is the Tactical Language Training System [Johnson et al., 2004] where listeners acquire communicative competence in spoken Arabic and other languages. By performing missions in an interactive story environment, it trains soldiers in not only the language of another country, but gives them an experience that engages them in the customs and culture of that country. This immersion is proven to be more effective than alternative methods for cultural acquisition. The FearNot system allows children to explore what happens in bullying in an nonthreatening environment in which they took responsibility for what happened to a victim, without themselves feeling victimized [Aylett et al., 2005].

The EST and Fabula Tales can be enhanced to support these types of interactive learning systems that are grounded in the narrative representation. Our framework affords interactivity and feedback from listeners in many available modes such as text interfacing with text selection, free typing, and feedback from the gestures and posture of the listener. With the ability for our system to expand to any domain, we can use stories written by real people from our blogs corpus to show people a story from different perspectives, making use of the point of view parameter and other narrative content and sentence planning manipulations.

6.4.3 Dialogue Authoring

Dialogue authoring in large games requires not only the creation of content, but the subtlety of its delivery which should vary from character to character. Manually authoring this dialogue can be tedious and time-consuming. The task becomes particularly difficult for games and stories with dynamic open worlds in which character parameters that should produce linguistic variation may change during gameplay or are decided procedurally at runtime. Short of writing all possible variants pertaining to all possible character parameters for all a game's dialogue segments, authors working with highly dynamic systems currently have no recourse for producing the extent of content that would be required to account for all linguistically meaningful character states. As such, we find open-world games today filled with stock dialogue segments that are used repetitively by many characters without any linguistic variation, even as these characters are modeled richly enough by their systems to give an actionable account of how their speech may vary [Klabunde, 2013]. We are building computational systems that are far more expressive than current authoring practice constricts them into being. These concerns are at play in linear games, too, in which the number of story paths may be limited to reduce authoring time or which require a large number of authors to create a variety of story paths.

As a potential future direction, we explore the potential of applying this approach to games with expansive open worlds with non-player characters (NPCs) who come from different parts of the world and have varied backgrounds, but currently all speak the same dialogue in the same way. While we have discussed how our method could be used to generate dialogue that varies according to character personality, the EST could also be used to produce dialogue variants corresponding to in-game regional dialects. PERSONAGE models are not restricted to the Big Five personality traits, but rather comprise values for the 67 parameters, from which models for unique regional dialects could easily be sculpted. Toward this, Walker et al. [2013] created a story world called *Heart of Shadows* and populated it with characters with unique character models. They began to create their own dialect for the realm with custom hedges, but to date the full flexibility of PERSONAGE and its parameters has not been fully exploited. Other recent work has made great strides toward richer modeling of social-group membership for virtual characters [Harrell et al., 2014]. Our approach to automatically producing linguistic variation according to such models would greatly enhance the impact of this type of modeling.

However, some open questions remain in dialogue modeling that we hinted at in our first discussion of dialogue in interactive narrative systems:

- (Q1) Can NPCs be given a personality fitted to the player?

- (Q2) Upon replay, do the NPCs utterances change under the assumption that the story is the same?
- (Q3) Are the characters impressionable? Can they form emotions or opinions about the each other or the player?
- (Q4) If Q4 is true, can these opinions alter the dialogue of the NPCs in conversation with each other or their responses to the player?

We posit that games and narrative systems with dialogue can take advantage of Fabula Tales' integration with voice and style parameters for Q1 and Q2. Q3 and Q4 could result in some integration of the emotional rule modeling, bolstering the EST and Fabula Tales as a discourse planner.

6.5 Summary of Contributions

This thesis expanded the NLG pipeline by proposing a solution to the NLG gap. The contributions of this thesis are the following:

1. The creation of the EST, a general translator to bridge the NLG gap by transforming the semantics of a story represented as a SIG to the syntactic DSYNTS formalism, maintaining semantic context, such as the rhetorical relations for temporal order, contrast, and causality;
2. The EST, not only a one-to-many syntactic translator, but a content planner for modeling character appraisal, temporal relationships, and non-narrative specific operations such as

verbosity and repetition;

3. The creation of the Fabula Tales sentence planner for altering point of view, inserting direct speech acts, and supplementing character voice using operations for lexical selection, aggregation, and pragmatic marker insertions;

4. The automatic creation of DSYNTS and a generation dictionary through the framework, alleviating authorial burdens;

5. The creation of the PersonaBank corpus of over 100 blogs encoded as SIGs to showcase the domain independence of the EST methodology and Fabula Tales narrative parameters;

6. Experimentation showing reader perceptions of characters and the overall correctness and engagement with a story changes as narratological parameters are varied;

7. A novel evaluation methodology of narrative retellings of an NLG system with an overgenerate and rank "Construct Your Own Story" paradigm with observations that there are general reader preference trends for narratological parameters (H_2, H_3, H_4 significant), but that individual reader preferences exist (H_1 not significant).

The EST and Fabula Tales NLG framework not only implements narrative variations, but demonstrates the variations can be applied to any story that can be encoded semantically as a SIG, and preserves semantic content in the *fabula*. Building on our methodology, we position future work to explore storytelling and natural human-agent interactions in different contextual environments using more diverse language and styles as we continue to develop more believable agents and dialogues by harnessing expressive NLG techniques.

Bibliography

Abbott, H. P. (2008). *The Cambridge Introduction to Narrative*. Cambridge University Press.

André, E., Rist, T., Van Mulken, S., Klesen, M., and Baldes, S. (2000). The Automated Design of Believable Dialogues for Animated Presentation Teams. *Embodied conversational agents*, pages 220–255.

Antoun, C., Antoun, M., Ryan, J. O., Samuel, B., Swanson, R., and Walker, M. A. (2015). Generating Natural Language Retellings from Prom Week Play Traces. *Proceedings of the PCG in Games*.

Aylett, R. S., Louchart, S., Dias, J., Paiva, A., and Vala, M. (2005). FearNot!–An Experiment in Emergent Narrative. In *Intelligent Virtual Agents*, pages 305–316. Springer.

Bae, B.-C., Cheong, Y.-G., and Young, R. M. (2011). Toward a Computational Model of Focalization in Narrative. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 313–315. ACM.

Bae, B.-C. and Young, R. M. (2009). Suspense? Surprise! or How to Generate Stories with

Surprise Endings by Exploiting the Disparity of Knowledge between a Story's Reader and its Characters. In *Interactive Storytelling*, pages 304–307. Springer Berlin Heidelberg.

Bal, M. (1997). Narratology. Introduction to the Theory of Narrative.

Bergmann, K., Kahl, S., and Kopp, S. (2013). Modeling the Semantic Coordination of Speech and Gesture under Cognitive and Linguistic Constraints. In *Intelligent Virtual Agents*, pages 203–216. Springer.

Biber, D. (1991). *Variation Across Speech and Writing*. Cambridge University Press.

Bohanek, J. G., Marin, K. A., Fivush, R., and Duke, M. P. (2006). Family Narrative Interaction and Children's Sense of Self. *Family process*, 45(1):39–54.

Bouayad-Agha, N., Scott, D. R., and Power, R. (1998). Integrating Content and Style in Documents: A Case Study of Patient Information Leaflets. *Information design journal*, 9(2-3):161–176.

Bowden, K., Lin, G., Reed, L., Fox Tree, J., and Walker, M. (2016). M2D: Monolog to Dialog Generation for Conversational Story Telling. In *Interactive Storytelling*. Springer International Publishing.

Brewer, W. F. and Lichtenstein, E. H. (1980). Event Schemas, Story Schemas, and Story Grammars.

Brewer, W. F. and Lichtenstein, E. H. (1982). Stories are to Entertain: A Structural-Affect Theory of Stories. *Journal of Pragmatics*, 6(5):473–486.

Broekens, J., Harbers, M., Brinkman, W.-P., Jonker, C. M., Van den Bosch, K., and Meyer, J.-J. (2012). Virtual Reality Negotiation Training Increases Negotiation Knowledge and Skill. In *Intelligent Virtual Agents*, pages 218–230. Springer.

Bruner, J. (1991). The Narrative Construction of Reality. *Critical Inquiry*, 18:1–21.

Bulitko, V., Solomon, S., Gratch, J., and Van Lent, M. (2008). Modeling Culturally and Emotionally Affected Behavior. In *AIIDE*.

Burton, K., Java, A., and Soboroff, I. (2009). The ICWSM 2009 Spinn3r Dataset. In *Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM 2009), San Jose, CA*.

Busemann, A. (1925). *Die Sprache der Jugend als Ausdruck der Entwicklungsrhythmik: sprachstatistische Untersuchungen*. Number 2. G. Fischer.

Cahill, L., Carroll, J., Evans, R., Paiva, D., Power, R., Scott, D., and van Deemter, K. (2001). From RAGS to RICHES: Exploiting the Potential of a Flexible Generation Architecture. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 106–113. Association for Computational Linguistics.

Cahn, J. E. (1989). Generating Expression in Synthesized Speech. Master's thesis, Massachusetts Institute of Technology, Dept. of Architecture.

Callaway, C. B. and Lester, J. C. (2002). Narrative Prose Generation. *Artificial Intelligence*, 139(2):213–252.

Carenini, G. and Moore, J. D. (2000). A Strategy for Generating Evaluative Arguments. In *Proceedings of the first international conference on Natural language generation-Volume 14*, pages 47–54. Association for Computational Linguistics.

Cassell, J. and Bickmore, T. (2003). Negotiated Collusion: Modeling Social Language and its Relationship Effects in Intelligent Agents. *User Modeling and User-Adapted Interaction*, 13(1-2):89–132.

Cavazza, M. and Charles, F. (2005). Dialogue Generation in Character-Based Interactive Storytelling. In *AIIDE*, pages 21–26.

Cheong, Y.-G. and Young, R. M. (2008). Narrative Generation for Suspense: Modeling and Evaluation. In *International Conference on Interactive Digital Storytelling*.

Collier, G. (2014). *Emotional Expression*. Psychology Press.

Conati, C. and Maclare, H. (2004). Evaluating a Probabilistic Model of Student Affect. In *Intelligent tutoring systems*, pages 55–66. Springer.

Dethlefs, N., Cuayáhuitl, H., Hastie, H., Rieser, V., and Lemon, O. (2014). Cluster-based Prediction of User Ratings for Stylistic Surface Realisation. *EACL 2014*, page 702.

Dias, J., Mascarenhas, S., and Paiva, A. (2011). FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework. In *Proceedings of the International Workshop on Standards for Emotion Modeling*.

Dibner, A. S. (1956). Cue-Counting: A Measure of Anxiety in Interviews. *Journal of Consulting Psychology*, 20(6):475.

DiMarco, C. and Hirst, G. (1993). A Computational Theory of Goal-Directed Style in Syntax. *Computational Linguistics*, 19(3):451–499.

Eichler, M. (1965). The Application of Verbal Behavior Analysis to the Study of Psychological Defense Mechanisms: Speech Patterns Associated with Sociopathic Behavior. *The Journal of Nervous and Mental Disease*, 141(6):658–663.

Elhadad, M. and Robin, J. (1996). An Overview of SURGE: A Reusable Comprehensive Syntactic Realization Component. Technical report, Technical Report 96-03, Ben Gurion University, Dept. of Computer Science, Beer Sheva, Israel.

Elson, D. (2012a). *Modeling Narrative Discourse*. PhD thesis, Columbia University, Dept. of Computer Science.

Elson, D. K. (2012b). Detecting Story Analogies from Annotations of Time, Action and Agency. In *Proceedings of the LREC 2012 Workshop on Computational Models of Narrative, Istanbul, Turkey*.

Elson, D. K. and McKeown, K. R. (2009). A Tool for Deep Semantic Encoding of Narrative Texts. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 9–12. Association for Computational Linguistics.

Elson, D. K. and McKeown, K. R. (2010). Tense and Aspect Assignment in Narrative Discourse. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 47–56. Association for Computational Linguistics.

Fellbaum, C. (2010). WordNet: An Electronic Lexical Database. *WordNet is available from http://www. cogsci. princeton. edu/wn*.

Fleischman, M. and Hovy, E. (2002a). Emotional Variation in Speech-Based Natural Language Generation. In *international natural language generation conference, Arden House, NY*, volume 2, page 4.

Fleischman, M. and Hovy, E. (2002b). Towards Emotional Variation in Speech-Based Natural Language Generation. In *Proceedings of the Second International Natural Language Generation Conference (INLG02)*, pages 57–64.

Forbes-Riley, K. and Litman, D. (2011). Designing and Evaluating a Wizarded Uncertainty-Adaptive Spoken Dialogue Tutoring System. *Computer Speech & Language*, 25(1):105–126.

Genette, G. and Lewin, J. E. (1983). *Narrative Discourse: An Essay in Method*. Cornell University Press.

Gerrig, R. (1993). *Experiencing narrative worlds: On the psychological activities of reading*. Yale University Press.

Gosling, S. D., Rentfrow, P. J., and Swann, W. B. (2003). A Very Brief Measure of the Big-Five Personality Domains. *Journal of Research in personality*, 37(6):504–528.

Gratch, J. and Marsella, S. (2001). Tears and Fears: Modeling Emotions and Emotional Behaviors in Synthetic Agents. In *Proceedings of the fifth international conference on Autonomous agents*, pages 278–285. ACM.

Gratch, J. and Marsella, S. (2003). Fight the Way you Train: The Role and Limits of Emotions in Training for Combat. *Brown J. World Aff.*, 10:63.

Gratch, J. and Marsella, S. (2004). Evaluating the Modeling and Use of Emotion in Virtual Humans. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 320–327. IEEE Computer Society.

Green, M. C. (2004). Transportation into Narrative Worlds: The Role of Prior Knowledge and Perceived Realism. *Discourse Processes*, 38(2):247–266.

Green, M. C. and Brock, T. C. (2000). The Role of Transportation in the Persuasiveness of Public Narratives. *Journal of personality and social psychology*, 79(5):701.

Green, S. J. and DiMarco, C. (1996). Stylistic Decision-Making in Natural Language Generation. In *Trends in Natural Language Generation An Artificial Intelligence Perspective*, pages 125–143. Springer.

Harrell, D., Kao, D., Lim, C., Lipshin, J., Sutherland, J., and Makivic, J. (2014). The Chimeria Platform: An Intelligent Narrative System for Modeling Social Identity-Related Experiences. Springer Berlin Heidelberg.

Hovy, E. (1987). Generating Natural Language Under Pragmatic Constraints. *Journal of Pragmatics*, 11(6):689–719.

Howcroft, D. M., Nakatsu, C., and White, M. (2013). Enhancing the Expression of Contrast in the SPaRKy Restaurant Corpus. *ENLG 2013*, page 30.

Hu, C., Walker, M. A., Neff, M., and Tree, J. E. F. (2015). Storytelling Agents with Personality and Adaptivity. In *Intelligent Virtual Agents*, pages 181–193. Springer.

Hu, Z., Dick, M., Chang, C.-N., Bowden, K., Neff, M., Fox Tree, J. E., and Walker, M. A. (2016). A Corpus of Gesture-Annotated Dialogs for Monologue-to-Dialogue Generation from Personal Narratives. Proc. of the 10th International Conference on Language Resources and Evaluation.

Hu, Z., Halberg, G., Jimenez, C., and Walker, M. (2014). Entrainment in Pedestrian Direction Giving: How Many Kinds of Entrainment. *Proc. IWSDS*, pages 90–101.

Inkpen, D. Z. and Hirst, G. (2004). Near-Synonym Choice in Natural Language Generation. In *Recent Advances in Natural Language Processing*, volume 3, pages 141–152.

Isard, A., Brockmann, C., and Oberlander, J. (2006). Individuality and Alignment in Generated Dialogues. *Proceedings of the 4th International Natural Language Generation Conference*.

Isbister, K. and Nass, C. (2000). Consistency of Personality in Interactive Characters: Verbal Cues, Non-Verbal Cues, and User Characteristics. *International journal of human-computer studies*, 53(2):251–267.

Johnson, W. L., Beal, C., Fowles-Winkler, A., Lauper, U., Marsella, S., Narayanan, S., Papachristou, D., and Vilhjálmsson, H. (2004). Tactical Language Training System: An Interim Report. In *Intelligent Tutoring Systems*, pages 336–345. Springer.

Joshi, A. K. and Schabes, Y. (1991). Tree-Adjoining Grammars and Lexicalized Grammars.

Kelso, M. T., Weyhrauch, P., and Bates, J. (1993). Dramatic Presence. *Presence*, 2(1):1–15.

Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2006). Extensive Classifications of English Verbs. In *Proceedings of the 12th EURALEX International Congress*, pages 1–15.

Klabunde, R. (2013). Greetings Generation in Video Role Playing Games. *ENLG 2013*, page 167.

Krause, M. S. and Pilisuk, M. (1961). Anxiety in Verbal Behavior: A Validation Study. *Journal of Consulting Psychology*, 25(5):414.

Kühne, V., Rosenthal-von der Pütten, A. M., and Krämer, N. C. (2013). Using Linguistic Alignment to Enhance Learning Experience with Pedagogical Agents: The Special Case of Dialect. In *Intelligent Virtual Agents*, pages 149–158. Springer.

Langkilde, I. (1998). Forest-based statistical sentence generation. In *In Proceedings of the 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, pages 170–177.

Langkilde, I. and Knight, K. (1998). Generation that Exploits Corpus-Based Statistical Knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.

Langkilde-Geary, I. (2002). An Empirical Verification of Coverage and Correctness for a General-Purpose Sentence Generator. In *Proceedings of the 12th International Natural Language Generation Workshop*, pages 17–24.

Lavoie, B. and Rambow, O. (1997). A Fast and Portable Realizer for Text Generation Systems. In *Proceedings of the fifth conference on Applied natural language processing*, pages 265–268. Association for Computational Linguistics.

Lebowitz, M. (1983). Creating a Story-Telling Universe. In *IJCAI*, pages 63–65. Citeseer.

Lebowitz, M. (1985). Story-Telling as Planning and Learning. *Poetics*, 14(6):483–502.

Li, B. (2015). *Learning Knowledge to Support Domain-Independent Narrative Intelligence*. PhD thesis, Georgia Institute of Technology.

Li, B., Lee-Urban, S., Johnston, G., and Riedl, M. O. (2013). Story Generation with Crowdsourced Plot Graphs. In *The 27th AAAI Conferece on Artificial Intelligence*.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2016). A Persona-Based Neural Conversation Model. *arXiv preprint arXiv:1603.06155*.

Lin, G. (2016). *Character Modeling through Dialogue for Expressive Natural Language Generation*. PhD thesis, University of California, Santa Cruz, Dept. of Computer Science.

Lin, G. I. and Walker, M. A. (2011). All the World's a Stage: Learning Character Models from Film. In *AIIDE*.

Lönneker, B. (2005). Narratological Knowledge for Natural Language Generation. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, pages 91–100. Citeseer.

Lukin, S. M., Bowden, K., Barackman, C., and Walker, M. A. (2016). PersonaBank: A Corpus of Personal Narratives and Their Story Intention Graphs. Proceedings of the 10th International Conference on Language Resources and Evaluation.

Lukin, S. M., Reed, L. I., and Walker, M. A. (2015). Generating Sentence Planning Variations for Story Telling. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 188.

Lukin, S. M., Ryan, J. O., and Walker, M. A. (2014). Automating Direct Speech Variations in Stories and Games. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Lukin, S. M. and Walker, M. A. (2015). Narrative Variations in a Virtual Storyteller. In *Intelligent Virtual Agents*, pages 320–331. Springer.

Madden, M. (2006). *99 Ways to Tell a Story*. Random House.

Mahl, G. F. (1956). Disturbances and Silences in the Patient's Speech in Psychotherapy. *The Journal of Abnormal and Social Psychology*, 53(1):1.

Mairesse, F. and Walker, M. A. (2008). A Personality-Based Framework for Utterance Generation in Dialogue Applications. In *AAAI Spring Symposium: Emotion, Personality, and Social Behavior*, pages 80–87.

Mairesse, F. and Walker, M. A. (2010). Towards Personality-Based User Adaptation: Psychologically Informed Stylistic Language Generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278.

Mairesse, F. and Walker, M. A. (2011). Controlling User Perceptions of Linguistic Style: Trainable Generation of Personality Traits. *Computational Linguistics*, 37(3):455–488.

Malone, T. W. and Lepper, M. R. (1987). Making Learning Fun: A Taxonomy of Intrinsic Motivations for Learning. *Aptitude, learning, and instruction*, 3(1987):223–253.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Maslow, A. H. (1943). A Theory of Human Motivation. *Psychological review*, 50(4):370.

Mateas, M. (2001). A Preliminary Poetics for Interactive Drama and Games. *Digital Creativity*, 12(3):140–152.

Mateas, M. and Stern, A. (2002). A Behavior Language for Story-Based Believable Agents. *IEEE Intelligent Systems*, 17(4):39–47.

Max-Neef, M., Elizalde, A., and Hopenhayn, M. (1992). Development and Human Needs. *Real-life economics: Understanding wealth creation*, pages 197–213.

McAdams, D. P., Josselson, R. E., and Lieblich, A. E. (2006). *Identity and Story: Creating Self in Narrative*. American Psychological Association.

McCoy, J., Treanor, M., Samuel, B., Mateas, M., and Wardrip-Fruin, N. (2011). Prom Week: Social Physics as Gameplay. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 319–321. ACM.

Mcquiggan, S. W., Mott, B. W., and Lester, J. C. (2008). Modeling Self-Efficacy in Intelligent Tutoring Systems: An Inductive Approach. *User modeling and user-adapted interaction*, 18(1-2):81–123.

Meehan, J. R. (1976). The Metanovel: Writing Stories by Computer. Technical report, Yale University, Dept. of Computer Science.

Meehan, J. R. (1977). TALE-SPIN, An Interactive Program that Writes Stories. In *IJCAI*, volume 77, pages 91–98. Citeseer.

Mel'čuk, I. A. (1988). *Dependency Syntax: Theory and Practice*. SUNY Press.

Miltsakaki, E., Prasad, R., Joshi, A. K., and Webber, B. L. (2004). The Penn Discourse Treebank. In *LREC*.

Montfort, N. (2007). *Generating Narrative Variation in Interactive Fiction*. PhD thesis, University of Pennsylvania, Dept. of Computer and Information Science.

Montfort, N., Stayton, E., and Campana, A. (2014). Expressing the Narrators Expectations. In *Seventh Intelligent Narrative Technologies Workshop*.

Morson, G. S. (1996). *Narrative and Freedom: The Shadows of Time*. Yale University Press.

Mott, B. and Lester, J. (2006). Narrative-Centered Tutorial Planning for Inquiry-Based Learning Environments. In *Intelligent Tutoring Systems*, pages 675–684. Springer.

Mott, B., McQuiggan, S., Lee, S., Lee, S. Y., and Lester, J. C. (2006). Narrative-Centered Environments for Guided Exploratory Learning. In *Proceedings of the AAMAS 2006 Workshop on Agent-Based Systems for Human Learning*, pages 22–28.

Munishkina, L., Parrish, J., and Walker, M. A. (2013). Fully-Automatic Interactive Story Design from Film Scripts. In *International Conference on Interactive Digital Storytelling*, pages 229–232. Springer.

Nakatsu, C. and White, M. (2010). Generating with Discourse Combinatory Categorial Grammar. *Linguistic Issues in Language Technology*, 4(1):1–62.

Ortony, A., Clore, G. L., and Collins, A. (1990). *The Cognitive Structure of Emotions*. Cambridge university press.

Paiva, D. S. and Evans, R. (2004). A Framework for stylistically controlled generation. In Belz, A., Evans, R., and Piwek, P., editors, *Natural Language Generation, Third Internatonal Conference, INLG 2004*, number 3123 in LNAI, pages 120–129. Springer.

Palmer, A. (2007). Universal Minds. *Semiotica*, 2007(165):205–225.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Paris, C. and Scott, D. (1994). Stylistic Variation in Multilingual Instructions. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 45–52. Association for Computational Linguistics.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic Inquiry and Word Count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001.

Pérez ý 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.

Petty, R. E., Cacioppo, J. T., and Goldman, R. (1981). Personal Involvement as a Determinant of Argument-Based Persuasion. *Journal of personality and social psychology*, 41(5):847.

Piwek, P. (2003). A Flexible Pragmatics-Driven Language Generator for Animated Agents. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 151–154. Association for Computational Linguistics.

Pizarro, D., Uhlmann, E., and Salovey, P. (2003). Asymmetry in Judgments of Moral Blame and Praise The Role of Perceived Metadesires. *Psychological Science*, 14(3):267–272.

Porayska-Pomsta, K. and Mellish, C. (2004). Modelling Politeness in Natural Language Generation. In *Natural Language Generation*, pages 141–150. Springer.

Power, R., Scott, D., and Bouayad-Agha, N. (2003). Generating Texts with Style. In *Computational Linguistics and Intelligent Text Processing*, pages 444–452. Springer.

Prince, G. (1974). *A Grammar of Stories: An Introduction*, volume 13. Walter de Gruyter.

Prince, G. (2003). *A Dictionary of Narratology*. University of Nebraska Press.

Propp, V. (1969). *Morphology of the Folktale*. University of Texas Press, second edition.

Queneau, R. and Wright, B. (1981). *Exercises in Style*, volume 513. New Directions Publishing.

Read, S. J., Miller, L. C., Appleby, P. R., Nwosu, M. E., Reynaldo, S., Lauren, A., and Putcha, A. (2006). Socially Optimized Learning in a Virtual Environment: Reducing Risky Sexual Behavior Among Men Who Have Sex with Men. *Human Communication Research*, 32(1):1–34.

Reed, A. A., Samuel, B., Sullivan, A., Grant, R., Grow, A., Lazaro, J., Mahal, J., Kurniawan, S., Walker, M. A., and Wardrip-Fruin, N. (2011). A Step Towards the Future of Role-Playing Games: The SpyFeet Mobile RPG Project. In *AIIDE*.

Reiter, E., Dale, R., and Feng, Z. (2000). *Building Natural Language Generation Systems*, volume 33. MIT Press.

Reiter, E., Sripada, S., Hunter, J., Yu, J., and Davy, I. (2005). Choosing Words in Computer-Generated Weather Forecasts. *Artificial Intelligence*, 167(1):137–169.

Riedl, M. O. and Young, R. M. (2006). Story Planning as Exploratory Creativity: Techniques for Expanding the Narrative Search Space. *New Generation Computing*, 24(3):303–323.

Rieser, V. and Lemon, O. (2011). *Reinforcement Learning for Adaptive Dialogue Systems: A Data-Driven Methodology for Dialogue Management and Natural Language Generation*. Springer Science & Business Media.

Rishes, E., Lukin, S. M., Elson, D. K., and Walker, M. A. (2013). Generating Different Story Tellings from Semantic Representations of Narrative. In *Interactive Storytelling*, pages 192–204. Springer International Publishing.

Ritter, A., Cherry, C., and Dolan, W. B. (2011). Data-Driven Response Generation in Social Media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.

Rowe, J., Mott, B., McQuiggan, S., Robison, J., Lee, S., and Lester, J. (2009). Crystal Island: A Narrative-Centered Learning Environment for Eighth Grade Microbiology. In *Workshop on Intelligent Educational Games at the 14th International Conference on Artificial Intelligence in Education, Brighton, UK*, pages 11–20.

Rowe, J. P., Ha, E. Y., and Lester, J. C. (2008). Archetype-Driven Character Dialogue Generation for Interactive Narrative. In *Intelligent Virtual Agents*, pages 45–58. Springer.

Rowe, J. P., Shores, L. R., Mott, B. W., and Lester, J. C. (2010). A Framework for Narrative Adaptation in Interactive Story-Based Learning Environments. In *Proceedings of the Intelligent Narrative Technologies III Workshop*, page 14. ACM.

Scherer, K. R. and Ceschi, G. (2000). Criteria for Emotion Recognition from Verbal and Nonverbal Expression: Studying Baggage Loss in the Airport. *Personality and social psychology bulletin*, 26(3):327–339.

Scherer, K. R. and Oshinsky, J. S. (1977). Cue Utilization in Emotion Attribution from Auditory Stimuli. *Motivation and emotion*, 1(4):331–346.

Scott, D. and de Souza, C. S. (1990). Getting the Message Across in RST-Based Text Generation. *Current research in natural language generation*, 4:47–73.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and Fast—But is it Good?:

Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015). A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. *arXiv preprint arXiv:1506.06714*.

Speer, N. K., Reynolds, J. R., Swallow, K. M., and Zacks, J. M. (2009). Reading Stories Activates Neural Representations of Visual and Motor Experiences. *Psychological Science*, 20(8):989–999.

Sripada, S. G., Reiter, E., Hunter, J., and Yu, J. (2003). Summarizing Neonatal Time Series Data. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 167–170. Association for Computational Linguistics.

Stanton, A. (2013). *What Makes A Good Story?* NPR/TED Staff.

Strapparava, C. and Mihalcea, R. (2007). Semeval-2007 Task 14: Affective Text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

Swanson, R. and Gordon, A. S. (2008). Say Anything: A Massively Collaborative Open Domain Story Writing Companion. In *Interactive Storytelling*, pages 32–40. Springer.

Swanson, R. and Gordon, A. S. (2012). Say Anything: Using Textual Case-Based Reasoning to

Enable Open-Domain Interactive Storytelling. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):16.

Swartout, W., Traum, D., Artstein, R., Noren, D., Debevec, P., Bronnenkant, K., Williams, J., Leuski, A., Narayanan, S., Piepol, D., et al. (2010). Ada and Grace: Toward Realistic and Engaging Virtual Museum Guides. In *Intelligent Virtual Agents*, pages 286–300. Springer.

Tapus, A. and Mataric, M. J. (2008). Socially Assistive Robots: The Link between Personality, Empathy, Physiological Signals, and Task Performance. In *AAAI Spring Symposium: Emotion, Personality, and Social Behavior*, pages 133–140.

Thorne, A. (1987). The Press of Personality: A Study of Conversations between Introverts and Extraverts. *Journal of Personality and Social Psychology*, 53(4):718.

Thorne, A. and McLean, K. C. (2003). Telling Traumatic Events in Adolescence: A Study of Master Narrative Positioning. *Connecting culture and memory: The development of an autobiographical self*, pages 169–185.

Turner, S. R. (1994). *The Creative Process: A Computer Model of Storytelling and Creativity*. Lawrence Erlbaum.

Vassos, S., Malliaraki, E., Dal Falco, F., Di Maggio, J., Massimetti, M., Giulia Nocentini, M., and Testa, A. (2016). Art-Bots: Toward Chat-based Conversational Experiences in Museums. In *International Conference on Interactive Digital Storytelling*. Springer.

Vinyals, O. and Le, Q. (2015). A Neural Conversational Model. *arXiv preprint arXiv:1506.05869*.

Walker, M., Rambow, O., and Rogati, M. (2002). Training a Sentence Planner for Spoken Dialogue Using Boosting. *Computer Speech and Language: Special Issue on Spoken Language Generation*, 16(3-4):409–433.

Walker, M., Stent, A., Mairesse, F., and Prasad, R. (2007). Individual and Domain Adaptation in Sentence Planning for Dialogue. *Journal of Artificial Intelligence Research*, 30:413–456.

Walker, M. A., Sawyer, J., Jimenez, C., Rishes, E., Lin, G. I., Hu, Z., Pinckard, J., and Wardrip-Fruin, N. (2013). Using Expressive Language Generation to Increase Authorial Leverage. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Wang, N., Johnson, W. L., Mayer, R. E., Rizzo, P., Shaw, E., and Collins, H. (2008). The Politeness Effect: Pedagogical Agents and Learning Outcomes. *International Journal of Human-Computer Studies*, 66(2):98–112.

Wang, Y. and Neff, M. (2013). The Influence of Prosody on the Requirements for Gesture-Text Alignment. In *Intelligent Virtual Agents*, pages 180–188. Springer.

Ward, A., McKeown, M., Utay, C., Medvedeva, O., and Crowley, R. (2012). Interactive Stories and Motivation to Read in the Raft Dyslexia Fluency Tutor. In *International Conference on Intelligent Virtual Agents*, pages 260–267. Springer.

Wardrip-Fruin, N. (2009). *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. MIT press.

Ware, S. G. and Young, R. M. (2011). CPOCL: A Narrative Planner Supporting Conflict. In *AIIDE*.

Ware, S. G. and Young, R. M. (2012). Validating a Plan-Based Model of Narrative Conflict. In *Proceedings of the International Conference on the Foundations of Digital Games*, pages 220–227. ACM.

Ware, S. G., Young, R. M., Harrison, B., and Roberts, D. L. (2014). A computational model of plan-based narrative conflict at the fabula level. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(3):271–288.

Weintraub, W. and Aronson, H. (1963). Clinical Judgment in Psychopharmacological Research. *Journal of neuropsychiatry*, 4:65.

Weintraub, W. and Aronson, H. (1964). The Application of Verbal Behavior Analysis to the Study of Psychological Defence Mechanisms II: Speech Pattern Associated with Impulsive Behavior. *The Journal of Nervous and Mental Disease*, 139(1):75–82.

Weintraub, W. and Aronson, H. (1965). The Application of Verbal Behavior Analysis to the Study of Psychological Defence Mechanisms III: Speech Pattern Associated with Delusional Behavior. *The Journal of nervous and mental disease*, 141(2):172–179.

Weintraub, W. and Aronson, H. (1967). The Application of Verbal Behavior Analysis to the Study of Psychological Defence Mechanisms IV: Speech Pattern Associated with Depressive Behavior. *The Journal of nervous and mental disease*, 144(1):22–28.

Yu, J., Reiter, E., Hunter, J., and Mellish, C. (2007). Choosing the Content of Textual Summaries of Large Time-Series Data Sets. *Natural Language Engineering*, 13(01):25–49.

Zukerman, I. and Litman, D. (2001). Natural Language Processing and User Modeling: Synergies and Limitations. *User Modeling and User-Adapted Interaction*, 11(1-2):129–158.

Appendix A

Translator Methodology

After defining our SchLexSynt model based on the Meaning-Text theory that DSYNTS employ, we build translation rules from the SIG to SchLexSynt. However, the SIG underlying data structures are complex, and we require a deep knowledge of the representation in order to build the EST transformations. Appendix A.1 discusses in detail the SIG data structures which allow us to perform the transformations detailed in Appendix A.2.

A.1 Story Intention Graph Data Structures

Annotator actions performed in the Scheherazade GUI are recorded and saved in a .VGL text file. It records when new characters are created, actions and properties are assigned, and items are deleted or modified. The sentence "The squirrel approaches the bowl" is represented as Table A.1 where *StateTimeType.Present;4.0==> #StateTimeType.Present;5.0* indicates the timeline this story point is in, the Present, and what number story point in the sequence this is. The root verb is *verb2053941_approach_something _approaches_something*. The first

```
StateTimeType.Present;4.0==>#StateTimeType.Present;5.0, **5:

actionAssert:verb 2053941_approach_something_approaches_something)

("anon_noun 2355227:squirrel(CharacterGender.Neutral)_1)",

  $anon_noun2881193:bowl()_1)$)
```

Table A.1: VGL file

approach is the lexeme of the verb. The *something approaches something* is the VerbNet frame, indicating a agent verb patient frame. The number *2053941* is the WordNet offset for the verb approach. *anon_noun2355227:squirrel (CharacterGender.Neutral)_1* following the root verb is the first argument of the verb with the lexeme is *squirrel* and the WordNet offset is *2355227*. The CharacterGender of the character has not been defined as male or female, so it is neutral by default. *anon_noun2881193:bowl()_1* is the patient of the root verb where the WordNet offset is *2881193* and the lexeme is *bowl*.

The Scheherazade API offers built in support to access much of the information we deem necessary in our semantic-to-syntactic transformation. The main hurdle is first understanding how to access the appropriate information. We add our own accessor methods in our EST framework for reusability that we discuss in Section 3.2. A rough sketch of the actual API data structures is shown in Figure A.1. For the purposes of our discussion and creation of the EST, the items discussed are the most relevant. There are many layers pertaining to the structure of the story points that we need to dig through in order to access the semantic information easily readable in the VGL format.

We provide a brief overview of important data structures below in order to better understand the EST structure later described; Scheherazade classes will be indicated in **bold**.

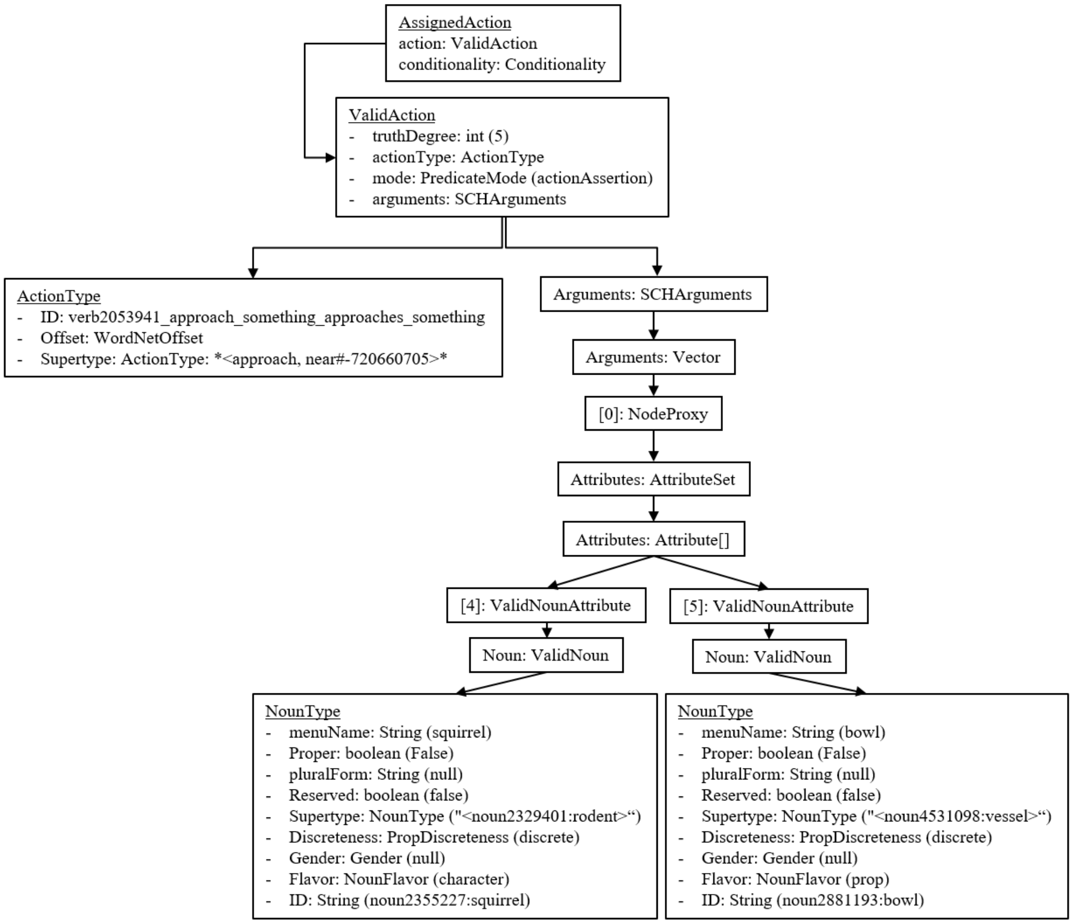

AssignedAction
action: ValidAction
conditionality: Conditionality
ValidAction
- truthDegree: int (5)
- actionType: ActionType
- mode: PredicateMode (actionAssertion)
- arguments: SCHArguments
ActionType
- ID: verb2053941_approach_something_approaches_something
- Offset: WordNetOffset
- Supertype: ActionType: *<approach, near#-720660705>*
Arguments: SCHArguments
Arguments: Vector
[0]: NodeProxy
Attributes: AttributeSet
Attributes: Attribute[]
[4]: ValidNounAttribute
[5]: ValidNounAttribute
Noun: ValidNoun
Noun: ValidNoun
NounType
- menuName: String (squirrel)
- Proper: boolean (False)
- pluralForm: String (null)
- Reserved: boolean (false)
- Supertype: NounType ("<noun2329401:rodent>")
- Discreteness: PropDiscreteness (discrete)
- Gender: Gender (null)
- Flavor: NounFlavor (character)
- ID: String (noun2355227:squirrel)
NounType
- menuName: String (bowl)
- Proper: boolean (False)
- pluralForm: String (null)
- Reserved: boolean (false)
- Supertype: NounType ("<noun4531098:vessel>")
- Discreteness: PropDiscreteness (discrete)
- Gender: Gender (null)
- Flavor: NounFlavor (prop)
- ID: String (noun2881193:bowl)

Figure A.1: Scheherazade data structure for verbs and nouns

ValidTimelines are created each time a new timeline is created in the GUI. By default, there is one for the present timeline. This corresponds to the *#StateTimeType.Present* in the VGL file. New timelines can be created in the GUI and are associated with new ValidTimeline objects.

Every span of text in the original story can be associated with an annotation. Each story point is associated with a ValidTimeline. This associated object is a story point is an **AssignedPredicate** and inherit an **AssignedAction** or **AssignedCondition**. These objects included information about the timeline in which the action or condition resides.

Nested inside the AssignedAction or AssignedConditon is a **ValidAction** or **ValidCondition** respectively. The ValidAction or Condition holds the actionType– information about the current verb including a *truthDegree* an INT that tells us whether the action happened or didn't happen and a **PredicateMode** which is an ENUM class indicating if the action or condition is an Assert, Gerund, Imperative, or Infinitive. It also has an **ActionType** or **ConditionType**. The Action or ConditionType has an *ID* in the form of "verb2053941_approach_something_ approaches_something" where the "2053491" is the WordNet offset, the first "approach" is the lexeme, and the "something_approaches_something" is the frame. It has a **WordNetOffset** and a *supertype* of an Action or ConditionType. For example <*approach, near#-720660705*> is the ID for the supertype ActionType of this ActionType with ID "approach".

The AssignedAction and AssignedCondition also holds the **SCHArguments**, which contains the subject and object. By iterating through the SCHArguments (Figure A.1 skips over some of the details of the nested objects), we eventually get to **ValidNoun** objects which represent the subject (if the ValidNoun is in index 4) and the object (if the ValidNoun is in index 5). ValidNoun has a ValidTimeline, linking this noun to the timeline it was created in, and a NounType. **NounType** holds all the information about the noun. In the example in Figure A.1, we see the *menuType* is the lexeme, e.g. "squirrel", *Proper* indicates if the noun is a common or proper noun, **Gender** which can be Female, Male, or Neutral (if not defined), and a **pluralForm** if the form cannot be made plural by regular rules and has a special lexeme. A *supertype* NounType shows the NounType above this one in the WordNet hierarchy, e.g. <noun2329401:rodent>. **Discreteness** indicates if this noun is continuous or not. The **NounFlavor** indicates if it is a Prop or Character, and finally, the *ID* e.g. noun2355227:squirrel.

From the AssignedAction, AssignedCondition, we can access all the verb and noun information. In order to get modifiers, we must reach into the ValidTimeline and find modifiers that match the Assigned Action or Condition. ValidActions and ValidConditions are not predicates and do not have links to the story interpreter for modifier information, thus modifiers cannot be directly derived from the ValidAction or ValidCondition. By iterating through each AssignedAction, we get an **AssignedModifier**, if one exists, for the action. From this, we get a **ModifierGrammaticalType** which can be *Adverbial* or not. To get the lexeme, we iterate to the **ModifierType** and get the raw string. The ModifierGrammaticalType could also be a **Preposition**, a set of ENUMs including Below, Between, From, In, Over, Through, To.

The SIG gives us the many affordances to the modeling of different levels of narratives. It provides a representation of the temporal sequence of events from the Timeline layer grounded in lexical resources. It also describes the underlying goals and plans affecting the needs and beliefs of the characters. We use the SIG and its affordances that are compatible with existing NLG technology. Although the SIG data structures are complex, the Scheherazade API allows us access and insight into how to access the information we need for our mapping.

A.2 Expressive Story Translator Rules

In most cases, the SchLexSynt attributes are filled by extracting information from Scheherazade data types. However, this does not help us with deciding the structure, or handling the more complicated logic of iterating through the Scheherazade data types. We begin by loading a SIG into the EST to access the API. We create a Story Manager class from the

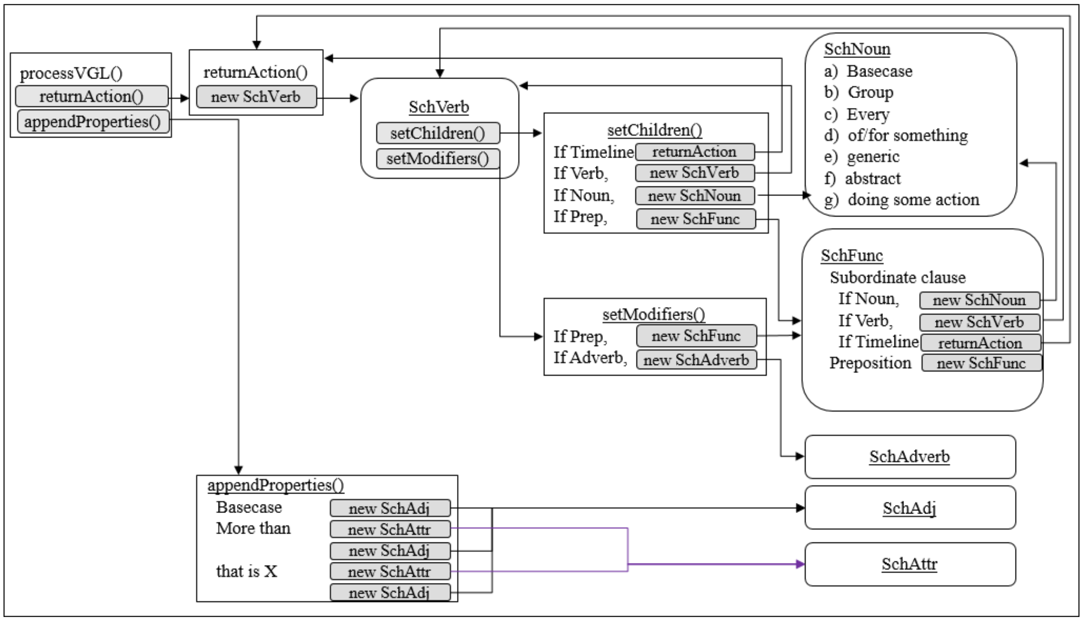


Figure A.2: SchLexSynt node creation

API which handles interfacing between the semantics of the SIG and the EST. We load all story points into the EST using the API (loadStoryPoints() in Figure 3.8) to give us a set of AssignedPredicates. We call processVGL() on the story points, which iterates through each AssignedPredicate to create a SchLexSynt tree. This method processes an AssignedPredicate story point and builds up the SchLexSynt tree. After processing each story point in processVGL(), we post-process each SchLexSynt tree in postProcessing() allowing for any major restructuring changes to correctly format the dsynts for writing. Here we apply narrative parameters directly onto the DSYNTS, using salient information from the SIG content. Finally, we write the properties to disk in writeDsynts().

In processVGL(), returnAction() is called on each AssignedPredicate which formulates the creation of SchLexSynt nodes, and appendProperties(), which connects adjectives in

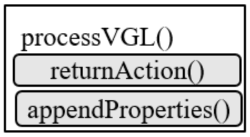


Figure A.3: Starting point for each AssignedPredicate story point retrieved from the API

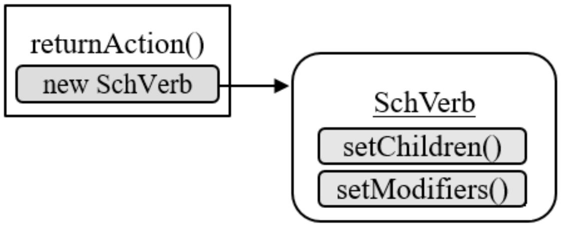


Figure A.4: SchLexSynt node creation

the SchLexSynt nodes (Figure A.3). For each retrieved AssignedPredicate, we create a new SchVerb. The initialization of SchVerbs involves first setting its children, then setting any adverbs (Figure A.4). In setChildren() (Figure A.5) we iterate through the SCHArguments of the AssignedPredicate, keeping track of the order they are found and assign them as the subject or object of the verb. A SCHArgument could be one of the four possible Scheherazade data types in Figure A.5. The first possible SCHArgument type is a ValidTimeline, indicating a reference to a hypothetical or alternative timeline in the story. The method begins again by calling returnAction() on the ValidTimeline argument (Figure A.4), recursively and eventually returning a SchVerb that can be appended to the root SchVerb.

The second SCHArgument type is a ValidAction, ValidCondition, AssignedAction, or AssignedCondition. The process is again recursive and we create a new SchVerb using the base case SchVerb class construction.

The third, and most complex child SCHArgument type is a NounName which has

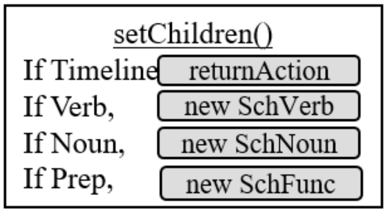


Figure A.5: Possible SCHArgument types in setChildren()

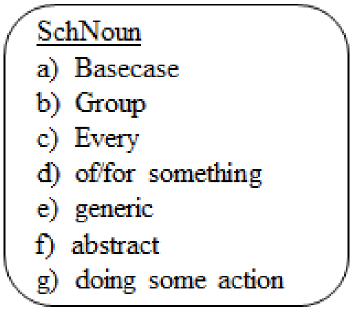


Figure A.6: Data types for new SchNoun

many types of input (Figure A.6). In the *base case* (a in Figure A.6), the lexeme can be simply assigned to the SchNoun. If the NounName is a *group* (b in Figure A.6), the lexeme of the NounName is set to the SchNoun and it is made plural. If the argument contains *every* (c in Figure A.6), (e.g. "every corner") a supplemental SchAttr object is created to support the lexeme "every" and appended as a parent of the SchNoun. The NounName may be a *type of something* or *role of something* (d in Figure A.6). For example, "the bird's feather" is a type of object unique to the bird, and in "the child's father" the father fills the role of parent. We get the arguments of the first noun and set all remaining children (NounNames) to be children of the SchNoun without an article and set the current SchNoun to be possessive. If the child contains *generic* (e in Figure A.6), we just set this SchNoun to the argument of the generic object. If the child contains *abstract* (f in Figure A.6) If the child contains *doing some action* (g in Figure A.6) this is a gerund.

The final SCHArgument for setChildren() is if the child is a Scheherazade Preposition (Figure A.7). If the Preposition is a subordinate clause (e.g. in order to, because, but, with something, of something), we create a SchFunc with the lexeme, then append the arguments which may be created as a new SchNoun, SchVerb, or ValidTimeline node which is appended to the new SchFunc.

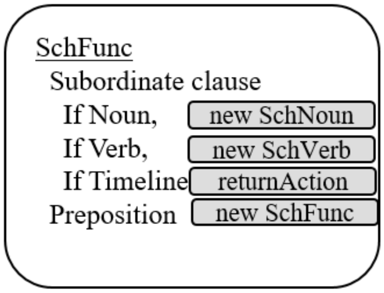


Figure A.7: Data types for new SchFunc()

After setting all the children in new SchVerb, we setModifiers() and map them to adverbs (Figure A.8). We iterate through the AssignedModifiers associated with the SchVerb. For each AssignedModifiers, we get the ModifierGrammaticalType. If it is an Adverbial type, we create a new SchAdverb from the lexeme and set it as a child of the SchVerb it modifies. If the AssignedModifier is not an adverbial, we repeat the SchFunc creation.

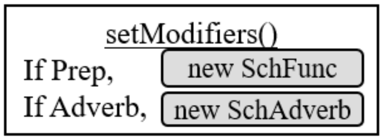


Figure A.8: Possible SCHArgument types in setModifiers()

After extracting the core arguments from the story points, we also have to extract adjectives and conditions in appendProperties() (Figure A.9). We follow a similar iteration as

```
<dsyntnode article="def" class="common_noun" gender="neut" lexeme="bug" number="pl" person="" property_0="slimy" rel="I" wn_offset="2236355">
  <dsyntnode class="adjective" lexeme="slimy" rel="ATTR" wn_offset="1133017"/>
</dsyntnode>
```

Table A.2: DSYNTS realized as The slimy bugs

setModifiers by iterating through the properties and the mapped story points to see if there is a match. New SchAdjs or SchAttr are created to support the properties. In addition, the property tag is put in the DSYNTS to be able to indicate to the discourse planner whether to include or exclude this property when realizing (Table A.2).

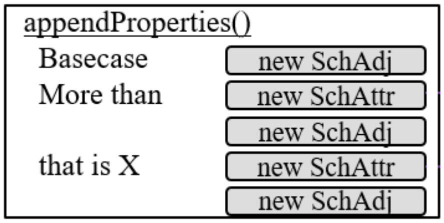


Figure A.9: appendProperties() conditions

returnAction() is repeated for every retrieved AssignedPredicate from loadStoryPoints() until we have created SchVerbs for every story point. The result is a list of SchLexSynts corresponding to each story point.

There are three main issues that can only be addressed after the SchLexSynt trees are constructed. To support infinitives, we must recognize there is a SchVerb child of a SchVerb. The final realization e.g. "The narrator planned to make mead". The naïve SchLexSynt construction would append the same subject to the second verb, resulting in a realization of "The narrator planned for her to make the mead". By removing the first subject of the second verb, we solve this problem. On the other hand, if the second subject is the same as the first and there is no object, we remove the second subject. For example “Lillian instructed the groups for the groups to free the wasp” becomes “Lillian instructed the groups to free the wasp”.

Another special case we have to handle is if we have an if-then clause. The tree structure in Scheherazade is different from the order that PERSONAGE needs the DSYNTS to be, so the clauses must be reordered. Similarly, to coordinate verbs and subjects, the conjunction node must be added and the tree restructured. Finally, Scheherazade encodes the structure of “to be able to” clauses differently from how DSYNTS requires. This case is also handled. Some of these post processing cases are covered in the walkthrough in Section 3.4.

After the post processing is complete, the SchLexSynt trees are structured properly and the SchLexSynt nodes have enough information to convert to DSYNTS (writeDsynts() in Figure 3.8). The algorithm traverses the SchLexSynt tree in-order and creates an XML node for each lexico-syntactic unit, outputting the required DSYNTS features.

Appendix B

Prolog Appraisal Rules

```
% Allows for attempting to cause a good thing or prevent a bad thing
hope(A,C) :-
    attemptToCause(A,B),
    providesFor(B,C).
hope(A,C) :-
    attemptToCause(A,B),
    wouldCause(B,C).
hope(A,C) :-
    attemptToPrevent(A,B),
    damages(B,C).
hope(A,C) :-
    attemptToPrevent(A,B),
    wouldPrevent(B,C).

% Allows for causing a bad thing or ceasing a good thing
despair(A,C) :-
   ceases(A,B),
   providesFor(B,C).
depair(A,C) :-
   ceases(A,B),
   wouldCause(B,C).
```

```
depair(A,C) :-
   actualizes(A,B),
   damages(B,C).
depair(A,C) :-
   actualizes(A,B),
   wouldPrevent(B,C).

% Allows for causing a good thing or ceasing a bad thing
joy(A,C) :-
   actualizes(A,B),
   providesFor(B,C).
joy(A,C) :-
   actualizes(A,B),
   wouldCause(B,C).
joy(A,C) :-
   ceases(A,B),
   damages(B,C).
joy(A,C) :-
   ceases(A,B),
   wouldPrevent(B,C).

% Allows for attempting to cause a bad thing or prevent a good thing
fear(A,C) :-
   attemptToPrevent(A,B),
   providesFor(B,C).
fear(A,C) :-
   attemptToPrevent(A,B),
   wouldCause(B,C).
fear(A,C) :-
   attemptToCause(A,B),
   damages(B,C).
fear(A,C) :-
   attemptToCause(A,B),
   wouldPrevent(B,C).
```