CS 224N Final Project Report

Ryan Thompson

"Mixing Deterministic and Probabilistic Models for Simple Story Generation"

The Problem

Creativity has never been a strong point for computers. Statistical Natural Language Generation is a difficult topic, since context and meaning are carried over for more than just 3 words ahead or 3 words behind. In highly deterministic settings where the structure of sentences can be crafted by humans ahead of time, such as in question answering, this is a simpler problem. But when discussing the problem of generating text in a freer setting, the old adage about monkeys and typewriters comes to mind.

In educational circles, with some slight crossover into natural language processing, story grammars have been debated and explored in the interest of best helping young children learn to read. Children recognize a certain grammar and set of story elements, which has led to discussion about how to write educational stories in a way that helps children learning to read, rather than hindering them. There are some commercial story generation tools that have been criticized in academic papers for not adequately providing this story grammar framework.

For my project, I set out to combine this story grammar and deterministic structure with the large statistical NLP base of knowledge in order to probabilistically generate sentences within the given framework. In order to achieve this, I used a corpus of 450+ fairy tales from the Andrew Lang fairy tales collection, which provided a hefty 1.45 million words with which to develop a language model, perform parts-of-speech tagging, and generate my own pseudo-fairytales.

The Model

In 1996, Anderson and Evans developed a canonical story grammar, synthesizing the work of several research groups from the earlier decades. In its simplest sense, a story was:

Setting -> Beginning Event -> Internal Reaction -> Attempt -> Ending

Element	**Definition**
Setting	Introduction of the main character and description of the time, location, and/or social context of the story.
Beginning Event	A cause which initiates a reaction or response of the main character.

Internal Reaction	An emotional response by the character which leads to the creation of a goal.
Attempt	An action of the character to achieve the goal.
Ending	Attainment or nonattainment of the goal by the character and/or the character's reaction to the outcome and/or a moral.

In this project, I broke this grammar sequence into several distinct parts: Characters, Places, Actions, Emotions, Goals, and Outcomes. They are pretty self-explanatory, although they interact in certain specific ways: an Action generally has a Character as the subject or object of its verb, and it may occur at a Place. Similarly, a Goal consists of a Character and an Action, Characters have Descriptions and Relationships, etc.

Grammar:

Story -> Setting BeginningEvent Reaction Attempt Ending

Setting -> Place Characters | Characters

Characters -> Character Characters | Character

Place -> Name Description

Description -> Adjective | Adjective Description

Character -> Name Description Interactions | Name Description Interactions Place

Interactions -> Interaction Interactions | e

Interaction -> Character Relationship

Beginning Event -> Action

Action -> Subject Verb Object Place | Subject Verb Object

Subject -> Character | noun | e

Object -> Character | noun | e

Reaction -> Emotion Goal

Emotion -> Character Feeling

Goal -> Character Verb Object

Attempt -> Action Attempt | Action

Ending -> Outcome Reaction

Outcome -> positive | negative

Implementation

When implementing this story generation project, I made extensive use of the RASP tool to

lexicalize, parse, and tag parts of the fairytale corpus. While I was unable to tag the entire corpora in CLAWS-7 POS tags due to the huge processing time and propensity to get booted from corn machines while running, I managed to get more than 12 MB of data from RASP. Using the CLAWS-7 tags and grammar relations on the data from RASP, I was able to generate Counters of words and their parts of speech, as well as CounterMaps of noun-adj pairs, verb-subject pairs, etc.

In addition, I trained a Trigram language model on the entire fairytale corpus – since we are only concerned with generation, instead of testing unseen data, no smoothing is needed, just measured counts. In addition to the traditional forward-trigram model, I also kept track of the word just before the current word, as well as the bigram before the current word, which is necessary for the most difficult part of generation – linking words in a way that results in coherent English phrases.

Sentences were generated basically according to the grammar above, with a rigid deterministic structure in place while constructing all sentences except for Actions and Goals. This mainly consisted of a limited set of words, phrases, transitions, etc chosen probabilistically and mixed in with traits like Character names and Place descriptions. For example, for an Emotion, the structure is: It made CHARACTER.NAME feel DESCRIPTOR FEELING. Descriptor is probabilistically selected from all words tagged as "RG" in the RASP output, FEELING is one of 3 positive or 3 negative emotions, and the Character's name is the name of the Character the emotion is being generator for.

Without any special work for Actions or Goals, a story at this point is not very probabilistic at all – we have a rigid structure, choose words from a limited vocabulary of, in most cases, preselected words, and everything is fairly repetitive. There is a lot of work that went into developing the framework, but it is not actually very interesting – the same goes for using RASP and developing the corpus. What really adds creativity to the story generator is the diversity in Actions that results from breaking out of the Subject Verb Object mold and linking up words in ways that still make sense. A Goal with verb=engage and object=affection can generate this: "`engage himself to the giant s affection`", which strikes the fancy and adds a more interesting element to the story.

I ended up using the same linkWords method for any set of two words, but almost exclusively for verbs. The general premise behind the method is that it takes two words and uses the trigram model to work inwards to find a match in a part of speech. However, instead of focusing on either the most likely part of speech or a random one, it creates a set and examines all parts of speech above a certain threshold. If word a and word b are being compared, we examine the most likely parts of speech to follow a and precede b. First we check if a's POS is in set B and if b's POS is in set A, then compare set A and set B for intersections. If there is a match, we find the most likely match via the language model counts, add it to the sentence, and return. If there is no match, we choose highly likely words (first the max from trigram, then backing off to a probabilistic bigram selection) and recursively find the link between those two new words. This allows us to keep the links fairly short and avoid the long rambling inanity that previously linked words.

Error Analysis

Verb choice: In order to make the story seem convincing, a lot of work was done to make sure that appropriate verbs were selected. Dialogue is difficult to randomly generate, at least much moreso that a series of plot points driven by action verbs. Early test stories were ruined by nonsensical verb choices that were either poorly represented in the language model (low counts), more often seen as nouns (ruining part-of-speech matching), or overwhelmingly seen in a single bigram or trigram.

From cvisual.com I downloaded a list of action verbs and matched them with the words already present in abundance in the corpus. For words that did not fit, or often led to poor constructions, I merely commented them out of my list of action verbs. Note that there are two separate taggings for verbs – VVD for past-tense verbs and VV0 for present tense. Past-tense verbs are used for most actions, except for Goal statements, which use present tense verbs. A large portion of time was spent converting verbs from RASP's spend+ed format back into normal English tenses: spent. In some cases it's useful to know that we saw the past tense form of spend, but in most cases I'd rather just know that we saw spent, which matches the language model.

Most Likely vs Probabilistic: In many situations throughout the project, we have the opportunity to either take the maximum value from a counter, or probabilistically select a value with probabilities based on frequencies. For selected parts-of-speech, anything other than maximum likelihood was a huge disaster. If we are trying to see if a noun is ever followed by a past tense verb, but the noun appears as a transitive verb 10% of the time, 10% of the time we will not be able to link where we should be able to, and we'll descend into nonsense.

For linking words, when we used solely most likely bigrams, we ran aground because many verbs do not take objects as often as we try to force them to in this story generator. Verbs are often followed by conjunctions or end sentences, moreso than leading directly into nouns. This led to long, semi-coherent links between words when a smarter, shorter link was obviously there. This sentence is an unfortunate example, using He, Hammered, and Work:

He could find that is the two namesakes along by and presently as bright and his guards at all a good enough to to a weary at this time until the bags all those anklets she was the stables and **hammered** at at than she found them for their necks with solid iron castle must be some blazing fire till you been of the giant the youth turned and maiden then she very soon fell to **work**

But going the opposite route is a problem too – we will often miss obvious links and have even less contextual sense. There are noticeable phrases that make sense in that jumble, but it just took a while before we got to one with "hammered" in it. Going probabilistically there is less internal "sense": `Knew more while the table table did she not fear`. This sentence is just linking verb Knew with object Fear, but goes off until it matches table with itself. It is problems like these that led to my switch to the final algorithm, which matches sets of possible parts-of-speech in order to ensure a match when one exists. "Knew fear" is a perfectly acceptable construction that was passed up because of low probabilities.

Linking algorithms: The multiple parts-of-speech matching linking algorithm was actually the third one I used. After an initial attempt at something more deterministic (ie filling in Subject Verb Object with determiners and prepositions), I decided that I didn't want to limit the creativity or vocabulary by creating a small, select list of possible verb, subject, object

pairings. My second attempt also tried to work inwards from two words, forwards from word A and backwards from B. However, instead of matching highly likely parts of speech, I focused on matching a single part of speech – either the most likely or a probabilistically selected one. This had the effect of ignoring many acceptable constructions, as well as going off on long inane rambles between words (see above). Now we don't see things like `Fell on the bride in this he could not her hand and wide open and had looked not want of vain i ve got back of death reached above the little and cover or mortal to come from all through him she was dismissed the hill for it he has sent a reason twisted themselves up and and rain` anymore.

A lot of error analysis went into making sure that the character names I provided behaved correctly. Since most of them are not traditional fairy tale names (or do not appear in the corpus), they need to be modeled by some other words when they appear as a subject or object in a verb phrase. I was very frustrated by something simple like John found Matilda – this should have been acceptable, but instead we kept getting "John and found Matilda", which makes no sense. Some errors have been eliminated by the choice of "brother" as the word we use to model subjects on – it works much better than "man" or "King", two words that are surprisingly not used as subjects in fairy tales very often. In the end, most errors were reduced with brother being selected, and the annoying "and" errors were manually removed with a rule in the linking algorithm.

Results and Conclusion

A final run of the program produced 10 stories:

```
1. In a whole, and pretty town There once was a woman helen who was very open. There is also prince, her superior, john, her friend, and meredith, her open superior. Windows of the rattle and then he held near john outside the whole, and pretty town. It made helen so sad. Helen needed to know the way. Then helen and stopped at carriage. Fortunately, after everything that happened, helen was able to know the way. It made helen quite triumphant. Fin!

2. There once was a girl aimee who was poor, poor, and terrible. There is also aurora, her poor relative. She cut off aurora in a healthful house. It made aimee far angry. If aimee decided to part to aurora. Aimee was step lady in a dark, and long house. Unfortunately, after everything that happened, aimee was unable to part to aurora. It made aimee quite angry. Fin!

3. There once was a boy broderick who was earnest, and dear. He lived in a possible, and gloomy town. There is also aimee, his earnest relative. Him cut
```

off broderick. It made broderick very sad. Broderick must work for you hidden yourself all this time passed and congratulating for yourself any time. After that, broderick it touched the sister in a full, and my street. Fortunately, after everything that happened, broderick was able to work for you hidden yourself all this time passed and congratulating for yourself any time. It made broderick so happy. Fin!

4. In a powerful cottage There once was a girl helen who was too good-natured. There is also prince, her good-natured superior. He struck prince. It made helen very angry. If helen promised to retire into kill prince. After that, helen thought the men at the pretty, and lonely castle. Fortunately, after everything that happened, helen was able to retire into kill prince. It made helen so happy. Fin!

5. There once was a woman aurora who was noble, and good-natured. She lived in a whole, and ugly bridge. A mantle split to aurora at the ghostly, obscure, small, and large bank. It made aurora very angry. Aurora needed to Give him consent. Aurora set me and dogs. Fortunately, after everything that happened, aurora was able to Give him consent. It made aurora too determined. Fin!

6. In a secret barn There once was a Sir king who was very old. There is also sherwood, his old superior. A enemy cut off sherwood. It made king very angry. King decided to Call the children. King sighed the girl around the secret barn. Fortunately, after everything that happened, king was able to Call the children. It made king too determined. Fin!

7. In a whole, and royal place There once was a woman aurora who was eldest, and ague. There is also thor, her eldest enemy. Thor cut off aurora. It made aurora so scared. Aurora must Take us and sweet lute. After that, aurora received the letter. Unfortunately, after everything that happened, aurora was unable to Take us and sweet lute. It made aurora too angry. It made thor so happy. Fin!

8. There once was a Sir sherwood who was very ready. He lived in a wild town. There is also king, his relative, micah, his relative, and thomas, his ready enemy. Thomas cut off sherwood around the high hill. It made sherwood some angry. So sherwood decided to spoke a word. Sherwood begged leave. Next, sherwood sent thomas. Fortunately, after everything that happened, sherwood was able to spoke a word. It made sherwood too happy. It made thomas too angry. Fin!

9. There once was a Sir john who was seest, and tiny. He lived in a healthful Moor+s. There is also micah, his seest enemy. Micah shut john. It made john so angry. John promised to push the boat. John seemed to micah. Unfortunately, after everything that happened, john was unable to push the boat. It made john so angry. It made micah so triumphant. Fin!

10. In a great, and great harbor There once was a girl elisabeth who was so certain. There is also helen, her friend, and meredith, her certain enemy. Meredith were in dust clouds float both on land and the letter to be off with elisabeth. It made elisabeth very angry. When elisabeth decided to desire each of their acquaintance. Next, helen presented meredith outside the largest town. Unfortunately, after everything that happened, elisabeth was unable to desire

each of their acquaintance. It made elisabeth very angry. It made meredith too determined. Fin!

In a triumph of natural language generation, stories 1, 6, 7, and 8 are actually coherent! It's much better than anything produced at the beginning, and now 4 stories out of 10 have promise, as opposed to 0 stories out of 10. Unfortunately, this is the only ranking metric I've implemented – subjective coherency. Some papers describe a ranking method for "interesting" stories, but those programs all generate multiple stories from a single seed. There is still a lot of work that could be done, at the very least with formatting and capitalization, and beyond that working on better linking between words and choosing better subject-verb-object triplets.

A lot of the current work in the field works given seeds – main characters are inputted or basic nouns are chosen, and then a story is generated. In McIntyre and Lapata's story generation paper in the 2009 Proceedings of ACL, they give an example where they generate the following story from the sentence "The family has the baby": The family has the baby. The baby is to seat the lady at the back. The baby sees the lady in the family. The family marries a lady for the triumph. The family quickly wishes the lady vanishes. Clearly, this story does not make a lot of sense, but all of the sentences have a sensical structure. Their paper also used the Andrew Lang fairy tales and RASP, but they also did a lot more work in chaining verbs together – they encode a graph of verbs that appear near to each other in order to have more consistency in the story, as well as ranking subject-verb pairs and other type pairs using a mutual information metric.

Overall, I am quite pleased with the way the stories turned out in the end, especially after a disastrous start when first experimenting with linking words. I've only scratched the iceberg, but there does not seem to be a lot of work on generating stories without seeds, and with a more probabilistic approach to linking algorithms. Personally, I'll take Elisabeth decided to desire each of their acquaintance over The baby sees the lady in the family any day – I appreciate whimsy, even if it is pseudorandomly generated.