A Data-Driven Case-Based Reasoning Approach to Interactive Storytelling

Reid Swanson[1] and Andrew S. Gordon[2]

[1] Walt Disney Imagineering Research & Development, Glendale CA 91201, USA,
Reid.Swanson@disney.com
[2] The Institute for Creative Technologies, Marina Del Rey CA 90291, USA,
gordon@ict.usc.edu

Abstract. In this paper we describe a data-driven interactive storytelling system similar to previous work by Gordon & Swanson. We addresses some of the problems of their system, by combining information retrieval, machine learning and natural language processing. To evaluate our system, we leverage emerging crowd-sourcing communities to collect orders of magnitude more data and show statistical improvement over their system. The end result is a computer agent capable of contributing to stories that are nearly indistinguishable form entirely human written ones to outside observers.

Keywords: interactive storytelling, case-based reasoning, natural language processing, crowd-sourcing

1 Introduction

The primary allure of digital interactive storytelling has been the vision of completely immersive environments that present the participants with the same richness of interactions in the virtual world as they experience in the real world. Over the years, incredible progress has been made in modeling the physical characteristics and the graphical interfaces with which we are submerged into these virtual worlds. However, progress in developing narrative technologies that allow complex social interactions with non-player characters and free interaction with all aspects of the environment has been much more arduous and less fruitful.

One of the primary reasons why progress has been slow, and many existing interactive storytelling systems do not come to life in a truly dynamic way, is that the development of these systems has been remarkably similar to those of traditional media. A movie, for example, typically has a primary author and a small team of additional writers who refine and shape the screenplay into a single coherent narrative. Similarly, a group of writers for a typical interactive narrative also begins with a single core idea, but instead breaks it into multiple narrative threads, or beats [11], that enable different storylines based on narrative decisions inferred by the system. Through manual analysis and the use of specialized tools, an effort is made to identify places in these narratives that the user can move from one thread to another without breaking the coherence of the entire story.

This is both a natural and well understood approach, but there are several problems that are immediately apparent limiting its effectiveness. For example, a traditional movie takes many months or even years to author and yet is only the manifestation of one static narrative thread. An interactive narrative of the same magnitude requires an enormous amount of extra effort, not only because of the additional content but also because of the extra verification required. To further complicate this already monumental hurdle, there are very few authors trained for this genre and even fewer standardized tools to help facilitate their creation. Given these limitations, it is important to explore new architectures for digital interactive storytelling that do not have such a tight coupling between the content authoring process and the narrative decision process.

Swanson & Gordon [16] first proposed a large-scale data driven approach that decouples the authoring process from the narrative decision process by completely separating the content authoring from the mechanics of the narrative engine. Instead of using a relatively small knowledge-base of highly structured rules and content, they leveraged the massive amounts of natural language personal stories on the Web. Even though some control must be given up in directing the plot, the amount of content available enables players to explore a rich virtual world full of the complex social interactions and emotion that make traditional narratives so compelling.

Their previous research has shown that this is a promising direction to pursue, however, there are several shortcomings that we address in this paper. We will first give a brief overview of the system and work done to recreate the *Say Anything* system using a new story corpus and information retrieval toolkit. Sections 2.3 and 2.4 describe new work that addresses several shortcomings of the previous version of *Say Anything*. In the following section, we will show how to use emerging crowd-sourcing communities to obtain orders of magnitude more evaluation data for only a few hundred dollars. We will use this data to show significant improvements over the previous system and will be able to produce stories, like the one below, that raters have judged nearly indistinguishable from entirely human written personal stories found on ordinary people's weblogs, but where every other sentence (in bold) was authored by our computer agent.

> The baseball player swung at the pitch, but missed by a lot. **The home plate ump checked with the third base ump who said Casto had fouled the ball off.** With that call, the game was still going and Casto had another chance to win the game for his team. **The Giants had a young guy named Sanchez pitching who did well.** With Casto's repuation as a good hitter, this was an intriguing and exciting matchup - who would win the battle and therefore the game? **Who would get it?!!** The suspense seemed to grow with each second and each movement of the pitcher, catcher, or batter. **Cursing myself for not having a camera with me, the camera phone would have to do, and snapped the photo above.** That photo showed the elation on Casto's face after getting the game-winning hit off of the tough pitcher Sanchez.

2 System Architecture

The style of game-play and system architecture described in this paper follows the *Say Anything* system developed by Swanson & Gordon [16]. Similar to their system, our system is a textual, turn-based game in which a human participant and a computer agent alternate writing sentences of a developing narrative. The objective is to develop a system that can generate a coherent sentence during each turn and enable the human to experience an interactive story adventure, as a whole, that is driven by their interests and desires. Our system can be thought of as a return to early text adventure games such as Adventure [6] with two primary exceptions. First, the story is not known in advance and second, the user is not limited to the actions hard-coded and preauthored into the system, but rather the user is free do anything they feel is appropriate given what has happened so far. To this end we are not concerned with attaining long narrative experiences or artifacts, as long as the user is able to enjoy their interaction and the resulting stories are judged to be of sufficient quality.

To achieve these goals, our system breaks from the standard representation formalisms typically used for interactive storytelling and narrative generation and instead, we follow a data-driven philosophy inspired by Case-Based Reasoning (CBR) [13]. In Cased-Based Reasoning new problems are tackled by starting from a known solution to a different but similar problem and then adapting that solution to fit the unique particulars of the current problem at hand.

Predicting what happens next in a narrative chain of events is a complex task that requires, at least, a deep understanding of the causal and temporal relations implicit in the discourse. An ideal representation of these relations for our case library would be discriminative enough to operate at this level of event structure. Although annotated corpora modeling these types of relations exist, unfortunately the performance of automated systems on general web text is not adequate at this time.

Instead, we make two simplifying assumptions that enable a more practical solution. First, we assume that narratives are told in temporal order and second, that events are adequately represented as entire sentences. Although neither is entirely true, the consequences of their approximation are not detrimental to the success of our system. However, they enable us to use a large corpus of stories written in plain English that we treat as perfect solutions to our turn based game. When the user types a sentence during their turn, our system will find the most similar sentence and story in our corpus to what the user has written so far. This story will act as a proxy for the user's story and the next sentence after the one retrieved will be used as the basis for the computer's contribution.

The remainder of this section will describe the three main components of our case-based architecture, briefly outlined above. The first is the creation of the case library that acts as a repository of previously solved solutions to the problem. The second is a retrieval phase in which we find a similar case in our library to the one the user is currently working on. The final step is to adapt and change portions of the retrieved sentence from phase two so that it better fits the new sentence the user is actually writing.

2.1 The Case Library Story Corpus

The reasoning capabilities of any case-based system are critically determined by the composition of the case library and so its creation is extremely important to the success of our system. Traditionally, these case libraries have been built by hand, however this will not provide us with the depth of content we been arguing for. Instead, we follow a similar approach to Swanson & Gordon [16] and mine the social web for personal stories written by ordinary people on their weblogs. We roughly define a personal story to be a first person description of temporal and causally connected sequence of events that the author actively participated in. For a more complete definition see [16].

Our case library was constructed by mining the 25 million English language blog posts between August 1 and October 1 2008, provided by Spinn3r.com as part of the ICWSM Dataset Challenge [2]. Each of the extracted blog entries identified as a story was treated as if it were a perfect solution to the turn-based game and added to our library.

To automatically identify stories in the ICWSM corpus we followed a similar approach to Gordon & Ganesan [8], which takes a binary classification approach to the problem. However, in our work, entire blog posts are labeled as a positive or negative instance of a personal story, as opposed to extracting smaller segments from the text. We developed a new gold standard training corpus from a small randomly chosen subset of 5,270 weblog entries to train our classifier. Each of these entries was annotated using a similar definition to the one proposed in Gordon & Ganesan. In order to assess inter-annotator agreement and to obtain a high quality training corpus the subset of entries were hand labeled by both of the authors in an iterative process until all the examples were give a category both annotators could agree upon. On the first iteration, we found a similar level of agreement as Gordon & Ganesan's previous work (Cohen's κ=0.68). However, after two more iterations we were able to come to an agreement on all 4,985 entries[3], of which 267 were labeled as stories.

The completed training corpus was used to train a Confidence Weighted linear classifier [7], which was used for its efficiency and highly competitive performance. A wide variety of feature sets were explored to find the best combination with our classifier. These ranged from simple lexical based features and a set of hand crafted "narrative" words capturing the idiosyncrasies of this particular genre. In addition to simple lexical based features, several syntactic and discourse features that attempt to model implicit structural elements of the document were also compared. These included the relative frequency of grammatical dependency triples obtained using an automated dependency parser [14]. Finally, a set of features that capture structural information across sentences were also considered primarily based on Barzilay & Lapata's Entity-Grids [1].

Our training corpus was divided into a development set consisting of 250 weblog entries, a training set of 3,985 and a test set of 750. The highest performing feature set included a combination of all the features. The final performance of

[3] Entries no longer available on the web at the time of annotation were disregarded.

our system with this combination of features was a precision of 0.591, recall of 0.414 and an F_1-score of 0.487. Applying this classifier to the 25 million weblog entries in the entire Spinn3r.com dataset, and holding out about 5% for development data, produced a corpus of 1.5 million stories comprised of about 42 million sentences.

2.2 Retrieval

Given our corpus of stories, it is necessary to develop a method for identifying similar stories and sentences to the one the user is currently writing. Fortunately, finding information in large collections of data is a well studied area in Computer Science. Many efficient algorithms have been developed in the Information Retrieval (IR) community that solve large scale document retrieval problems in fractions of a second. Manning, Raghavan & Shutze [10] provide an overview of some of the fundamental algorithms that are widely used. Many toolkits exist that implement these and other more sophisticated techniques, and for this work we chose to the Terrier toolkit [12] because it was far more efficient than any other software package we tried.

This information retrieval mechanism provides the base algorithm for our generation method. First, the user's most recent sentence is passed into our system. This sentence is then preprocessed to remove stop words such as determiners and other low value tokens from the input. The remaining words are lowercased and used as keywords to the Terrier retrieval component, which returns a ranked list of sentences from our database and the top 10 are returned to the user in ranked order. We present a choice of sentences to the user for two primary reasons. First, we would like to give the user a reasonable chance of finding a suitable continuation for their story, so that they do not become frustrated early in the process. Second, every time the user chooses a sentence other than the top ranked candidate selected by default we gain a valuable piece of information that we can exploit to further improve the system.

2.3 Reranking

Although Swanson & Gordon [16] showed that even a simple technique like the one described in the previous section can be quite effective, many of its problems should be readily apparent. The first thing to notice is a limitation of the IR mechanism itself. Keyword based systems suffer from relatively poor recall, because many sentences in English can express the same thought or meaning using an entirely disparate set of words. The reliance on a single sentence to determine the similarity between a preauthored story and the user's developing narrative is also problematic. Swanson & Gordon tried combining two separate indexes to address this problem but found a simple combination is not sufficient because the proper weighting is not known *a priori*. Finally, regardless of how similar the retrieved stories and sentences are, so far there have not been any checks to ensure that what the system returns is actually coherent with what the user has written up to this point.

This remainder of this section will describe a unified framework that is capable of addressing each of these issues, while still maintaining nearly the same (real-time) latency as the simple retrieval approach described in the previous section. This new approach is nearly identical to the simple IR based algorithm, but adds one additional phase after retrieving the candidate sentences from Terrier, but before returning them to the user. In this second phase, the retrieved candidates (which there can now be more than 10) are reordered based on a new set of arbitrary features using a machine learning reranking algorithm.

For this work we chose an algorithm similar to the one described by Collins [4], because it supports online learning, is extremely efficient and usually performs competitively. Similar to supervised classification algorithms, our reranking algorithm requires a set of labeled training examples in order to learn a model that can be applied to new data. To collect training data for our ranking algorithm, we used the trace of our system from the simple IR based method described in the previous section. Every time the user selected an alternative sentence we obtained a training instance in which the (partial) ranking was known. Although we may not know the absolute ranking of all the sentences, we do know that the user preferred their selection more than the current best guess of the system.

During the first stage of our evaluation process we collected 1,168 stories using only the information retrieval method. During the process of writing these stories, people took a total of 5,310 turns writing sentences with the system. Of these 5,310 turns the user selected one of the other nine alternatives 4,395 times. Following a standard approach, a development, training and testing dataset were created to test our feature sets offline. The development set was composed of 100 stories and a total of 381 sentence pairs. 3,305 sentence pairs from 900 stories were included in the training set. The remaining 184 stories and 688 sentence pairs were used for the test set.

A wide range of features were explored for this task that cannot be fully explained in detail. These features tried to address the issues raised at the beginning of this section, such as using more contextual information. Discourse features developed in the coherence modeling community, such as Entity-Grids [1] and machine translation models [15] were also included. Additionally, several other novel feature sets were also investigated, such as variation in sentence length, coreference heuristics, and the transition of verb tenses across sentences.

Although all of the features in isolation showed some improvement over the baseline, the Entity-Grid based features were by far the most predictive. On average, the candidate sentence that the person actually chose for their story was the 4.7th sentence in the list (from 0 to 9). After applying the reranker trained using Entity-Grid features alone, it was able to reorder the candidate sentences so that the person's selection was almost always presented near the top of the list (0.33th sentence on average). Although this is an astounding improvement, we should not necessarily expect such a dramatic effect during actual online interaction with the complete system. Ultimately, we chose an aggregate feature set that performed nearly as well, but included several lexicalized features we

hypothesized would work well in combination with the adaptation component described in the next section.

2.4 Adaptation

There are many ways in which a retrieved sentence could fail to continue a user's story in a coherent way. The system could fail completely in finding an adequate sentence. There could also be more subtle errors due to an implicit mapping problem between similar semantic classes (e.g., football and baseball stories). Sometimes, however the errors are due to more surface level criteria, such as differences in the pronoun or verb agreement, such as third person versus first. Although the ultimate goal would be able to address all of the failure types, this work only considers the more surface level changes that could be performed.

The algorithm for fixing these types of shallow errors is a five step process. Step (1) begins by identifying all of the pronouns and proper names used in the subject or object position in the grammatical parse tree of the sentence. For each identified position, a set of valid replacement words is created using a replacement table for each type of word that could appear. The first set of tables correspond to five different classes of pronouns: subjective (e.g., I, he, she, we), objective, reflexive, possessive and possessive determiners. If the target word is contained within one of these tables, then this set is used for the candidate replacements. Proper names are handled similarly, but can also be replaced by any previously mentioned proper name (i.e. the cast of characters).

Step (2) involves generating a new sentence for every possible combination of the replacements in each target word set. Unfortunately, the number of combinations for sentences with more than a few target replacement candidates becomes prohibitively large. To prevent the set of candidates from exploding, a simple heuristic was used to limit the total number of possibilities. For any given target word, a maximum of two alternatives were selected as possible replacements. These alternatives were chosen by sampling the entire set of valid possibilities based on each word's relative frequency of occurrence in the entire story.

It is hoped that the characters participating in the events of one of these alternative sentences (or the original) will more closely adhere to the narrative intentions implied by the user's story. However, small changes to the subject of a verb can lead to ungrammatical agreement between the two. For example, when the subject of the sentence

I have more than one lemonade.

is changed to *He*, then the new sentence is no longer grammatical.

Step (3) combats this issue with a special dictionary [5] that provides, where applicable, the number, person, gender and tense for every lexical entry in the dictionary. The entry containing all lexical variations of any verb whose subject has been adapted is looked up in the dictionary using the number and person information available from the unaltered sentence. The lexical variation corresponding to this entry that matches the number and person information of the new adapted subject is then used to replace the previously ungrammatical verb.

In addition to ungrammatical verb agreement, altering a noun in the sentence can disrupt the coreference interpretation within the candidate sentence. Step (4) attempts to preserve the coreference interpretation of the unaltered sentence with the adapted one. A simple co-reference resolution algorithm, similar to the one proposed by Hobbs [9] was used to identify co-referring pronouns and proper names in the original, unaltered document. Once identified the number and gender of the appropriate pronouns was changed to reflect the alterations made in the previous steps. However, this may also cause problems with the verb agreement in the sentence and is addressed in the same way as step (3).

After applying these 4 steps to a candidate sentence a potentially large number of variations are produced. Step (5) uses a simple heuristic to prevent a candidate sentence template from dominating the results shown to the user.

3 Evaluation & Results

Evaluating interactive storytelling applications can be almost as challenging as developing the system itself. The success of a system is nearly always defined by a set of subjective criteria that requires human judges to evaluate. Without a fair amount of hand-waving, it is often difficult to show convincing results, because obtaining a large enough pool of independent testers to obtain clear and statistically significant data has typically been expensive and time consuming.

As a solution to a broader superset of problems, which are generally trivially easy for humans to solve, but still to difficult for automated systems to complete, Amazon created the Mechanical Turk website. In essence, it is a centralized cyber-location where *requesters* post work to be done (**H**uman **I**ntelligence **T**asks) and *workers* complete the jobs for a small monetary reward. Requesters are free to offer whatever they believe is a fair price for the task and workers can choose to work on HITs that interest them. The use of Mechanical Turk in academic studies has been steadily increasing and it has been shown with enough safeguards and redundancy, near expert quality annotation and data can be obtain from non-expert Mechanical Turk workers [3].

Mechanical Turk was used for both collecting a large sample of stories written with our system, and to collect subjective ratings from an independent set of users. Four different sets of stories were collected, during two phases, using variations of the generation mechanisms described above. To prevent Mechanical Turk workers from clicking through the HIT and ensuring some basic level of effort with the system, an 8 sentence (4 turn) minimum story length requirement was imposed on the workers. After each story was successfully completed the user was asked to answer several questions about their story on a scale from 1 (bad) to 5 (good) that tried to assess different aspects of the process:

1. *Does the story make sense? (Coherence)*
2. *Is the story believable? (Believability)*
3. *Did you have fun writing the story? (Entertainment)*
4. *How easy was the story to write? (Usability)*

Model	# Stories	Coherence	Believability	Usability	Entertainment
Unigram	601	3.46 ± 1.11	3.53 ± 1.16	3.08 ± 1.19	3.99 ± 1.05
Bigram	567	3.63 ± 1.11	3.59 ± 1.20	3.27 ± 1.19	4.14 ± 1.02
Reranking	443	3.51 ± 1.15	3.62 ± 1.20	3.85 ± 1.05	4.38 ± 0.83
Adaptation	429	3.46 ± 1.07	3.55 ± 1.19	3.90 ± 1.02	4.33 ± 0.85

Table 1. Author rating results from 1 (bad) to 5 (good).

Although other, potentially better, metrics for evaluating the narrative qualities of the story exist, these were chosen because they are not overwhelming and are easy for a non-technical person to understand.

In the first phase, only the retrieval mechanism was used to generate sentences, without reranking or adaptation. However, roughly half the stories were generated using query terms based solely on unigrams (601 stories) and the other half using bigram phrase queries (567 stories). In the second phase, a bigram retrieval model was used to collect 443 stories that included the reranking component and 429 stories that included the full adaptation component.

After the two story collection phases were complete a new batch of HITs were published to Mechanical Turk to obtain a separate set of independent judgments about the stories, using an analogous set of questions. Each story was rated by a maximum of 8 unique raters in order to provide more reliable statistics for each story and model. Additionally, as a baseline, a little more than one hundred stories from our held out weblog story corpus were included in the mix.

The results of the authors' subjective ratings on their own stories is presented in Table 1[4]. Considering only the usability and the entertainment ratings, these results are very encouraging. The usability of the system increases for each of the models tested and there is a dramatic increase when the reranker is introduced. Similarly, the introduction of the reranking system significantly improves the enjoyment the users have in writing their stories. Looking at the coherence and believability measures are not as clear cut, however, and there is actually a slight decrease in coherence from the Bigram to the Reranking model. The reasons for this are not fully understood but will be discussed later in this section.

In addition to subjective ratings obtained by the story authors, several objective statistics about the stories can also give us a better understanding of the characteristics of each model. Some of these statistics are presented in Table 2[5]. **Max** is the maximum story length (in total sentences) that was written with the model, while **Avg** is the average total number of sentences $\pm$ the standard deviation. Although the reranker was able to pick out the user's selected sentence after it had been selected (see section 2.3), the mean reciprocal rank **MRR** statistic shows that it is not able to predict which sentence the user will pick.

[4] $p < 0.05$ (χ^2) between the reranking and retrieval-only models for everything other than Coherence and Believability.

[5] $p < 0.05$ (t-test) between the the retrieval and the reranking models for the time based statistics.

Model	Max	Avg	MRR	Time(s)	(s)/Sen
Unigram	27	9.41 ± 2.31	0.36 ± 0.30	460.6 ± 411.8	44.9 ± 32.0
Bigram	25	9.50 ± 2.51	0.34 ± 0.29	492.4 ± 463.7	47.9 ± 35.6
Reranking	27	9.53 ± 2.68	0.28 ± 0.07	399.2 ± 294.3	40.1 ± 22.8
Adaptation	36	9.63 ± 3.07	0.23 ± 0.04	406.1 ± 286.5	39.3 ± 20.6

Table 2. Story authoring statistics.

The **Time(s)** is a measure of how long it took for the user to accept the HIT until they submitted their results and **(s)/Sen** is a similar measure normalized by the length of the story. Despite writing slightly longer stories, it actually took the users much less time to complete. Given that the quality of the stories from the authors perspective over all the models are at the very least on par with each other, this seems to suggest that the reranking and adaptation models are returning a set of sentences that are easier to choose from.

The subjective ratings from the authors' perspective and the objective statistics about the stories start to shape our understanding of these different models are operating. However, some doubts remain because of the somewhat contradictory results in which the user's rated the stories equally coherent, yet found the reranking based systems much more usable, entertaining and easy to write. Examining the judgments of independent raters can provide a different perspective and are presented in Table 1[6]. To account for a slight bias toward lower ratings introduced by a combination of dishonest workers and the layout of the rating form, the average rating for each of the models was performed in the following way. Each story only contributed one rating to the average over all stories in the model, which was determined by the rating with the most votes. If there was no single winner then an average of the tied values was used. On the left side of the graph is a baseline of completely human written stories. These are complete stories entirely written by a single human from a held out portion of our weblog corpus described above. Although it would also be interesting to compare our system to a human-to-human collaboration, our baseline provides an adequate upper-bound for coherence at a minimum. These human authored stories achieved a 3.9 Coherence, a 4.1 Believability and a 2.9 Entertainment value. In contrast, the readers found the stories written with the Unigram model much less coherent (3.4), believable (3.5) and entertaining (2.7). Although using bigram queries improves the situation a little bit, the reranking based model is less than 0.1 points away from human written stories in coherence, which has reduced the gap to the upper bound by 82%.

These results, along with several of the author ratings and story statistics clearly show the reranking based model generates a better set of candidates, which result in stories that are nearly indistinguishable from personal stories found on ordinary people's weblogs. However, this does not resolve why authors

[6] $p < 0.05$ between the retrieval-only models and the reranking model for Coherence. $p > 0.05$ between the human stories and reranking model except for Believability.

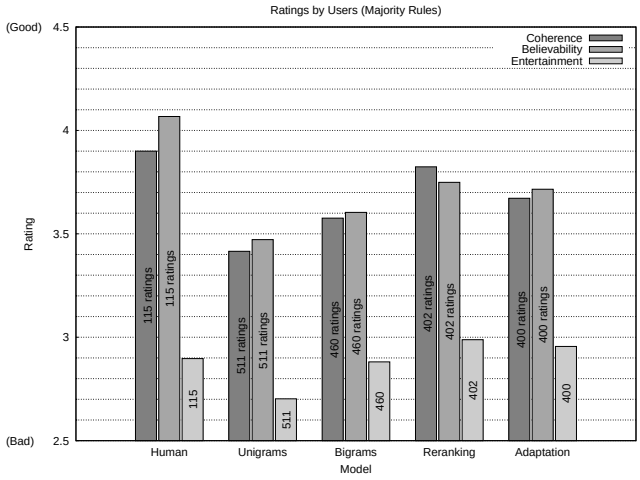

Fig. 1. Independent story rating results (1 to 5).

did not rate their own stories more coherent when using these models even though other indicators suggest they do. One possible explanation supported by this data is that the authors are using a different definition of coherence than was originally intended. Unlike independent readers, the authors enter into each turn with an agenda. Regardless as to whether any of the candidate sentences continue the story in a linguistically or logically coherent way, the user may still feel a disconnect between what they wanted to say and what their available options are and rate the coherence lower than an objective outside observer would do.

4 Conclusion

In this paper we have furthered the open-domain interactive storytelling paradigm developed by Swanson & Gordon by addressing some of the major shortcomings of their system. Unfortunately, the adaptation component was not able to provide additional performance, but its potential benefits are worth exploring in future work. However, the machine learning approach to reranking candidate sentences enabled us to more accurately balance the weight between local context (a sentence) and broader context (an entire story), which was not previously possible. Additionally, this approach supplies a unified and well founded way to include other arbitrary features to improve the quality, such as the discourse coherence features discussed in section 2.3. Using data obtained from a trace of the system provided enough training data for our reranker that it was able to allow users to author stories with our system that were nearly indistinguishable from entirely human written stories.

References

[1] Regina Barzilay and Mirella Lapata. Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 141–148, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

[2] Kevin Burton, Akshay Java, and Ian Soboroff. The ICWSM 2009 spinn3r dataset. In *Proceedings of the Third Annual Conference on Weblogs and Social Media*, San Jose, CA, May 2009.

[3] Chris Callison-Burch. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, pages 286–295, Singapore, 2009. Association for Computational Linguistics.

[4] Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 263–270, Philadelphia, Pennsylvania, 2001. Association for Computational Linguistics.

[5] Blandine Courtois and Max D. Silberztein. *Dictionnaires lectroniques du franais.* Langue Francaise. Larousse, Paris, 1989.

[6] Will Crowther and Don Woods. Adventure. 1977.

[7] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271, Helsinki, Finland, 2008. ACM.

[8] Andrew S. Gordon and Kavita Ganesan. Automated story capture from conversational speech. In *K-CAP*, pages 145–152, 2005.

[9] Jerry Hobbs. Pronoun resolution. Research Report 76-1, Department of Computer Sciences, City College, City University of New York, August 1976.

[10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval.* Cambridge University Press, July 2008.

[11] Michael Mateas and Andrew Stern. Towards integrating plot and character for interactive drama. *In Working Notes of the Social Intelligent Agents: The Human in the Loop Symposium. AAAI Fall Symposium Series. Menlo Park*, pages 113—118, 2000.

[12] I. Ounis. Research directions in terrier: a search engine for advanced retrieval on the web. http://eprints.gla.ac.uk/14096/, 2007.

[13] Christopher K. Riesbeck and Roger C. Schank. *Inside Case-Based Reasoning.* Lawrence Erlbaum, 1 edition, July 1989.

[14] Kenji Sagae and Alon Lavie. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 691–698, Sydney, Australia, 2006. Association for Computational Linguistics.

[15] Radu Soricut and Daniel Marcu. Discourse generation using utility-trained coherence models. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 803–810, Sydney, Australia, 2006. Association for Computational Linguistics.

[16] Reid Swanson and Andrew S. Gordon. Say anything: A massively collaborative open domain story writing companion. In *Interactive Storytelling*, pages 32–40. 2008.