Unsupervised Text Classification for Natural Language Interactive Narratives

Jenna Bellassai,[1] Andrew S. Gordon,[2] Melissa Roemmele,[2]
Margaret Cychosz,[3] Obiageli Odimegwu,[2] Olivia Connolly[2]
[1]Oberlin College, Oberlin, Ohio USA
[2]University of Southern California, Los Angeles, California USA
[3]University of California Berkeley, Berkeley, California USA
jbellass@oberlin.edu, {gordon, roemmele}@ict.usc.edu, mcychosz@berkeley.edu, {odimegwu, oconnoll}@usc.edu

Abstract

Natural language interactive narratives are a variant of traditional branching storylines where player actions are expressed in natural language rather than by selecting among choices. Previous efforts have handled the richness of natural language input using machine learning technologies for text classification, bootstrapping supervised machine learning approaches with human-in-the-loop data acquisition or by using expected player input as fake training data. This paper explores a third alternative, where unsupervised text classifiers are used to automatically route player input to the most appropriate storyline branch. We describe the Data-driven Interactive Narrative Engine (DINE), a web-based tool for authoring and deploying natural language interactive narratives. To compare the performance of different algorithms for unsupervised text classification, we collected thousands of user inputs from hundreds of crowdsourced participants playing 25 different scenarios, and hand-annotated them to create a gold-standard test set. Through comparative evaluations, we identified an unsupervised algorithm for narrative text classification that approaches the performance of supervised text classification algorithms. We discuss how this technology supports authors in the rapid creation and deployment of interactive narrative experiences, with authorial burdens similar to that of traditional branching storylines.

Introduction

Among the most common forms of interactive narrative is the branching storyline, exemplified by the first choose-your-own-adventure book, *Cave of Time* (Packard 1979). In this work, as in the hypertext fiction works that would follow, players are presented with a short narrative context and a list of possible actions for the player-character, where the choice determines how the storyline unfolds. Although works of this type have seen commercial and artistic success over the years, the genre is often maligned in interactive narrative research because the agency of players is very limited, the authoring burden grows exponentially with the number choices in each playthrough, and the static content does not promote replay. Indeed, these problems have spurred much of the research in the field of intelligent narrative technologies, where radically different approaches to interactive narrative are typically pursued.

Instead of abandoning branching storylines, other research has sought to transform them into something qualitatively different by changing the mode of player interaction. In natural language interactive narratives, players take actions by typing their intentions as free text input, which is analyzed by software to route the player down one storyline branch or another in a branching storyline. By obscuring storyline possibilities, natural language interactive narratives provide some of the same play style as Interactive Fiction (Montfort 2003), but using unconstrained language rather than a restricted vocabulary of commands. To handle the richness of natural language player input, previous work has turned to text classification techniques using supervised machine learning (Gordon et al. 2004; Traum et al. 2015). Here, the task is to classify the player input into classes linked to storyline branches, each designed to coherently continue the storyline given the actions expressed by the class instances. Well-designed scenarios anticipate a wide range of player input, and provide outcomes that are responsive to a diversity of player intentions.

The biggest challenge in developing natural language interactive narratives has been in acquiring and annotating input from actual players. In this framework, acquisition of player input serves two purposes in the development process. First, it provides the scenario authors with an indication of the breadth of player input that should be anticipated, allowing authors to craft setups that best manage this scope and sufficient branches to be responsive to it. Second, this input serves as training data for supervised text classification, where machine learning algorithms learn to classify player input in the same way as a human annotator. However, these two purposes work against each other in several ways. Before sufficient training data has been acquired, the performance of a system may be unacceptably poor for players, leading them to quit or provide inputs that are unrepresentative of what they would type if the system was performing as intended. As authors make changes to setups and outcomes, they potentially invalidate the annotations of input collected previously, forcing additional data collection and annotation. The result is a Chicken-and-Egg problem, where either a robust scenario design or large amounts of annotated data are required to obtain the other.

Two solutions to this problem have been pursued in previous research. First is the use of so-called “Wizard-of-Oz” data-collection methods, where a human-in-the-loop executes the classification task in real-time interactions with players, e.g., (Gandhe and Traum 2014). This approach affords real-time annotation of input and provides human-level classification performance to the player, but requires the participation of a wizard in every play test. A second solution is to train the classification algorithms with fake data, i.e., seeding it with many labeled examples of expected player input in place of actual input, e.g., (Hill et al. 2003). This approach allows subsequent data-collection with actual players to be completely automated, but may not provide classification performance that is sufficiently high to provide a coherent experience for the players.

In this paper we pursue a third approach to this problem, where *unsupervised* text classification algorithms are used to route player input to the most appropriate of the available outcomes. In unsupervised text classification, an algorithm selects the most appropriate outcome for a given player input using information gleaned from some other data source, e.g., statistical information about words in a large text corpus, rather than from hand-labeled training data. We hypothesized that contemporary algorithms for unsupervised text classification are sufficient to provide coherent experiences to players of natural language interactive narratives, without requiring the collection and annotation of training data. We describe a new web-based platform for authoring and deploying interactive narratives, the Data-driven Interactive Narrative Engine (DINE), which uses unsupervised text classification to enable natural language interaction. We used this platform to author dozens of short interactive narratives across a wide variety of fictional genres, and to collect thousands of text inputs from hundreds of crowdsourced players. We annotated each input to create a gold-standard test set, which we used to compare the performance of different algorithms for unsupervised text classification, as well as supervised algorithms that use either actual or fake training data. Our results demonstrate that unsupervised text classification is a viable alternative to Wizard-of-Oz data collection or the use of fake training data, and affords the rapid development and deployment of interactive narrative experiences.

Related Work

Natural language processing in interactive narratives has historically shared many of the methods and technologies of research in natural language dialogue systems. Early examples such as *Facade* (Mateas and Stern 2003) and *Mission Rehearsal Exercise* (Rickel et al. 2002) advanced storylines through dialogue with virtual characters, supported by knowledge-based parsers and formal models of the story world. Supervised text classification was used in the *TLAC-XL* (Hill et al. 2003) and *Leaders* (Gordon et al. 2004) systems, where fake training data was used to scaffold the data-collection process. Researchers have worked to improve the coherence of these systems by inserting bridging statements between player inputs and outcomes (Gandhe, Gordon, and Traum 2006), or by producing huge libraries of possible narrative responses (Traum et al. 2015).

We are not aware of other work on unsupervised text classification in interactive narrative applications. However, our approach shares much in common with previous work on case-based interactive narrative (Swanson and Gordon 2012; Roemmele and Gordon 2015). As in these previous works, we use large corpora of narrative text to coherently respond to player input. Instead of inserting excerpts directly into the human-computer interaction, however, the corpus is used as a source for word co-occurrence statistics.

Data-driven Interactive Narrative Engine

The Data-driven Interactive Narrative Engine (DINE) is a technology for building branching storylines where player actions are expressed via natural-language input. In DINE scenarios, a writer hand-authors all of the textual content that players see, structured into sequences of *pages* consisting of a *setup* and a set of possible *outcomes*. A DINE page is analogous to a single page of a classic choose-your-own-adventure book (Packard 1979), where the setup describes a situation in which the player must make a decision about what to do next. Instead of being presented with a list of choices, however, players are given an empty text box to describe their actions. The DINE system analyzes this text and automatically selects the appropriate outcome to display in response. At the discretion of the author, some of these outcomes can direct the player to different DINE pages, while others simply elaborate the situation presented on the current page. To author effective DINE pages, writers need only to design setups that encourage a narrow range of player inputs, and write enough outcomes to cover this range.

The core technology used in DINE is a text classifier whose job is to select the most appropriate outcome given the player's input text. Instead of a traditional supervised text classifier, which requires copious amounts of hand-annotated training data, DINE uses an unsupervised text classifier. These models encode the word-level statistical regularities that exist between passages in large text corpora, and use this information to automatically select the most coherent outcome of a player's input from among those the scenario author provides. By removing the laborious task of collecting and annotating training data, DINE aims to support the creation of natural language interactive narratives with authorial burdens that are closer to that of traditional branching storylines. The evaluation section of this paper describes the unsupervised approaches that we have investigated in this research, and compares their performance with traditional supervised classification methods.

We developed a web-based platform for authoring and deploying DINE scenarios, facilitating the evaluation of our approach with crowdsourced players.[1] Authors can create new pages by entering the text of the page's setup and possible outcomes. Each outcome consists of the following parts:

1. An (optional) list of example player inputs that should evoke the outcome,
2. the text displayed to the player when the outcome is selected, and
3. an (optional) identifier of a page that should be displayed immediately following the outcome. Left unspecified, the interaction remains on the current page. Alternatively,

[1] http://dine.ict.usc.edu

the author can indicate that the outcome is terminal, and should end the interaction.

When a player launches a DINE scenario on this website, the setup text of the first page is presented with a free-text input box. When the player enters a sentence, the DINE system ranks the possible outcomes, and displays the highest-ranked outcome that has not already been shown during the player's interaction on the current page. Where the author has linked the outcome to another page, the player is then presented with a *continue* button that advances to that page. Where the author has indicated a terminal outcome, the player is then presented with an end-of-scenario message, "The End." Otherwise, a new free-text input box appears, and the interaction on the page continues as before.

Example Scenario

(setup) I had to get my rig to Albuquerque before the shop opened at 8AM the next morning. Mendleton was already riding me hard for late deliveries. One more bad turnover and I was out of a job. I'd lose my rig, my paycheck and any hope in hell of making rent next month. But I was so exhausted. I'd been driving for 15 hours straight and had 8 more in front of me. My eyes were heavy, my head drooping, as I contemplated my options. If I didn't pull over, I could fall asleep at the wheel but if I did: nothing, but the tip of Mendleton's finger as he showed me the door, would be waiting for me when I woke. I looked around my truck to find things to occupy me. There was an almost empty pack of cigarettes, an energy drink, half of a sad sandwich from earlier in the day, a sole stick of chewing gum and that was it. That was all I had to prevent my 18-wheeler from driving off the road. Or of course I could pull over and– I didn't know what to do.

In this DINE scenario, entitled *Pull Over / Sleep Under*, the player takes on the unenviable role of a tired truck driver trying to stay awake on a highway. The situation poses a choice: should the player pull over and get some sleep, or push on through the night to make the delivery on time? Alternatively, a resourceful player may try several different ways to stay awake while driving. The author of this scenario must anticipate the sorts of player actions that the situation affords, and craft coherent outcomes—without knowing ahead of time how these actions will be articulated in natural language. Different outcomes can serve different functions in the interaction. Consider the following exchanges:

(input) `> I rolled down the window`

(outcome) As I rolled down the windows to usher in the night air, I felt disheartened. It was muggy out. Lukewarm and goopy like soup left untended and allowed to congeal. It settled around me, heavy, making me even more drowsy than before. The road swam, slow and sluggish, before my eyes.

(input) `> I drank my energy drink`

(outcome) As I gulped the energy drink down in long, greedy pulls, I felt hopeless. I had tried this before. Coffee. Five-hour Energy. Even caffeine pills once. It would work but only for an hour or two. I had eight sitting between me and the drop off point. The caffeine boost wouldn't hold that long. No way. No how. After a certain point, nothing would replicate the effect of a good night's sleep on my state of mind and in an hour, maybe two, I'd be faced with the same problem. Only it would be compounded this time by the comedown from caffeine burning itself off and tiring me out.

(input) `> I finished my sandwich`

(outcome) Eating the sandwich only made me more drowsy. I felt it sitting heavy at the bottom of my belly. My mouth tasted like soggy bread and stale cheese. And my eyes, flickering shut and open, a signal, warning me that I had to choose and choose soon before my body gave over and the choice was made for me.

(input) `> I turned on the radio`

(outcome) The radio blared country music into the silence of the car. Something by Chesney something Parton something Swift something or the other. It didn't matter. I couldn't tell the difference. It was all just a dull ringing in the air around me. My eyes kept easing shut and before snapping open again.

In these four examples, the selected outcome does not actually move the scenario forward; the player is shown the outcome text and immediately prompted for another input sentence. The DINE system considers each subsequent input just as the first, selecting from among the same set of outcomes minus those already shown. In contrast, other page outcomes can advance the storyline, either by transitioning to a different DINE page or by ending the scenario. For example, this page has two outcomes in response to the primary decision posed to the player. First, they may choose to pull over for the night:

(input) `> I pulled off the highway at the next exit`

(outcome) As I pulled over to sleep for the night, I felt a sense of trepidation. Sure, sleeping would ensure that I didn't harm anyone but it also meant that there was no way I'd get the contents of my truck to Albuquerque on time. Mendleton wasn't going to be happy about that. No matter what the rationale behind my decision was, Mendleton wasn't going to be happy... THE END

Or alternatively:

(input) `> I kept going and hoped for the best`

(outcome) As I decided to keep driving, I felt a sense of trepidation. Even now my eyes were slipping closed and sliding shut. Even now the road was a blurry, wavering snake before me. Now a curve. Then a straight line. Now a curve. Then a straight line. And was that a bend ahead? The miles between my lone truck, on this ever changing road, and Albuquerque seemed infinite. THE END

In DINE scenarios such as this, there are several distinct ways that the system can fail to provide a coherent interaction with the player. First, the player may type an intention that was not anticipated by the author, i.e., where no existing outcome would be a coherent response. In the extreme case, an uncooperative player can easily force an incoherent response by offering a bizarre input.

(input) `> I pulled a rabbit out of my hat`

(outcome) Chewing gum was useless. All it did was make my jaw as tired as my drooping eyes. I still had to decide. Pull over and maybe forfeit my job, my rig, or keep on driving and maybe forfeit my life.

Unsupervised text classification can be very robust even when given rare vocabulary, but will fail when presented with out-of-vocabulary input.

(input) `> J'ai ouvert la fenêtre`

(outcome) Playing car games wasn't helping any. I had counted sheep. Spied nothing with my tired eyes. And there were no license plates to count. All there was to occupy me was the gravity of the decision looming before me. Pull over, sleep and potentially lose my job or keep driving and risk falling asleep at the wheel.

DINE displays the highest-rank outcome that has not already been shown to the player, which causes incoherence when players try an action repeatedly.

(input) `> I turned up the radio even louder this time`

(outcome) The dial tone from my phone rang loud in the silence of the car. There was no response. Of course there wasn't. It was 3AM in the morning, everyone, friend and otherwise was tucked in, tight, to bed. Without the distraction a phone call would provide, my dilemma remained the same: pull over and sleep for the night or keep on driving.

Finally, even when a player's input is expected by an author, the underlying classification model can fail to select the right outcome. In this paper, we focus only on this source of incoherence, seeking to maximize the accuracy of the text classification task when a coherent outcome for user input is among those written by the scenario author. Whereas the other sources of incoherence may be mitigated by improvements in scenario and interface design, automatically matching user input to appropriate outcomes requires a robust mechanism for handling unconstrained natural language input. In this paper, we explore how AI advances in natural language processing can meet this need.

Evaluation

To compare the performance of different unsupervised classification models, we created a gold-standard test set by collecting and annotating input from crowdsourced participants playing many different DINE scenarios. To facilitate data collection, we first hired two accomplished fiction writers to craft 25 new DINE scenarios using the online authoring tool. Most of these scenarios were authored as single-page scenarios, for a total of 30 pages with an average of 9.1 outcomes per page. The topics and genres of these fictional scenarios were left up to the authors, and included science-fiction adventures, psychological thrillers, romantic comedies, and skills-training situations. We encouraged the authors to explore the space of possible interaction styles afforded by the DINE software, and conducted daily discussions of the merits of design decisions. To support evaluations of supervised approaches using fake training data, both writers authored four examples of expected player input for every outcome on every page.

We recruited 393 participants using an online crowdsourcing service.[2] Each participant completed one of the 25 interactive DINE scenarios, and was compensated $1.00 USD for their time, which was no more than 8 minutes. During this data collection phase, a single algorithm was used to select outcomes across all scenarios (PMI 1M$_6$, described below). At the end of each scenario, we asked participants to rate "How coherent was your story?" on a five-point Likert scale, with five as the highest rating. The mean coherence score was 3.55 out of 5 (SD=1.20), indicating that the subjective quality of these scenarios, paired with the PMI 1M$_6$ unsupervised model, was sufficiently high to provide largely-coherent experiences to crowdsourced participants.

A total of 2368 user inputs were collected across the 25 scenarios. Each user input was hand-labeled with the most appropriate outcome by the original author of the scenario. The author was also given the option to either annotate a user input as garbage to be ignored (8.83% of input) or lacking an appropriate outcome (24.6% of input). Additionally, we investigated the inter-rater agreement (Cohen's Kappa) between our two writers on nine scenarios (910 additional annotations), finding moderate agreement on the best outcome where both agreed one was available (κ=0.637), and moderate agreement on a three-way decision on whether an input was garbage, lacking an appropriate outcome, or should have been assigned to one of the outcomes available (κ=0.651).

We compared the performance of a variety of unsupervised, supervised, and mixed approaches, described below. Table 1 summarizes the performance of each approach, listing both the raw accuracy (percent agreement) and the mean chance-corrected agreement (Cohen's Kappa) to normalize for the varying number of outcomes across the 25 scenarios.

[2] http://www.crowdflower.com

In all reported results, the expected chance agreement is estimated as one divided by the number of possible outcomes for a given scenario. For each of the supervised approaches that use the annotated data as training data, results are reported using leave-one-out cross-validation. The statistical significance of observed performance gains were evaluated using the compute-intensive randomized test with stratified shuffling (Noreen 1989).

PMI 1M Models Our first unsupervised classifier (a) was a direct implementation of an approach to ranking the causal strength between two adjacent sentences (Gordon, Bejan, and Sagae 2011). The measure computes the mean pairwise association between words in each sentence, which in DINE are the words in the user input and the first six words of a possible outcome. Association is computed using an asymmetric variant of Pointwise Mutual Information (PMI) (Church and Harrison 1990), where a corpus co-occurrence is tabulated only if a word appears *after* another within the word window, here set to 25 words to better capture associations across adjacent phrases and sentences. As a corpus for co-occurrence statistics, we used the text of one million personal stories automatically filtered from public weblogs (Gordon and Swanson 2009).

Our interest in this model stemmed from its success in benchmark evaluations for commonsense causal reasoning over natural language text, specifically the Choice Of Plausible Alternatives (COPA) evaluation (Roemmele, Bejan, and Gordon 2011). COPA questions pose a textual premise and ask which of two textual alternatives is more plausibly the causal consequence (or antecedent). We view DINE pages as variants of COPA questions, where the premise text is provided as input by players, and the task is to select the most plausible causal consequence among the outcomes written by the page's author.

The PMI 1M model was the first one implemented for the DINE system, and was sufficiently accurate to facilitate the collection of real player input from crowdsourced participants. We considered only the first six words of each outcome when using this model (a) during data collection, based on our early informal tests. We subsequently tuned this parameter using the gold-standard annotations, and found that using seven words slightly improved performance (b).

AvgMaxSim Model We observed that users' input was often semantically similar to the beginning of the text of the best outcome, not only causally associated. An additional unsupervised algorithm (c) ranks outcomes according to the *average maximum similarity* between all words in the user input I and the first W words of each outcome O.

$$AvgMaxSim_W(I, O) = \frac{\sum_{i=1}^{Len(I)} \max_{1 \le j \le W} Sim(I_i, O_j)}{Len(I)}$$

Here, $Sim(I_i, O_j)$ is computed as the cosine distance between each word's `Word2Vec` representation (Mikolov et al. 2013). In our work, we use pre-trained distributed word vectors computed from 100 billion words from a Google News dataset[3]. Tuning for performance, we found that this algorithm worked best when considering the first ten words of each outcome (W=10). This parameter-tuned model (c) significantly outperformed the best PMI-based algorithm (b).

We were surprised by the ability of this model to predict appropriate outcomes of player actions, and wondered whether the distributed vector representations were encoding some of the same long-distance statistical regularities between causally-related words as the PMI 1M models. To test this hypothesis, we applied the AvgMaxSim model to the COPA evaluation, hoping that it might outperform the PMI models on this task, as well. However, we found that the AvgMaxSim model performed only marginally better than the random baseline on COPA evaluation. This suggests that the Word2Vec representations used in the AvgMaxSim model are not capturing causal associations between words, which is required to perform well on the COPA task. Instead, we conclude that the strength of this model for DINE scenarios is due to substantial semantic overlap between the words in the player input and the first 10 words of authored outcomes. As seen earlier in the example scenario *Pull Over / Sleep Under*, the first sentence of each outcome can be stylistically written to include words that are closely related to those in expected user input, e.g., "I finished my sandwich," and "Eating the sandwich only made me more drowsy." Here, the semantic overlap between the identical word "sandwich" and the related words "finished" and "eating" seem to be more helpful in selecting appropriate outcomes than the causal associations between words like "sandwich" and "drowsy."

Averaged Perceptron For our supervised algorithms, we used the Averaged Perceptron machine learning algorithm trained on unigram, bigram, and trigram features from different sets of training data. For algorithm (d), we trained one classifier for each scenario page using the fake examples of expected user input that the author had written for each outcome (four examples per outcome). For algorithm (e), we trained classifiers using the gold-standard annotations of user input for each scenario page, and evaluated performance using leave-one-out cross-validation. For algorithm (f), we trained on the combination of author examples and gold-standard annotations, and evaluated performance using leave-one-out cross-validation on the gold-standard annotations only. The author's fake examples and the gold-standard annotations were complementary, yielding a classifier that significantly outperformed a supervised classifier trained on the gold-standard annotations alone (e).

Mixture Models We conducted a series of studies to see if further gains could be achieved by combining our best unsupervised classifier (c) with a supervised approach. In algorithm (g), we combined the scores produced by algorithms (c) and (d) using a weighted average, and tuned the mixture parameter α at 0.1 increments to identify the best-performing mixture. Using the author's fake examples in this manner did not significantly improve performance.

[3] https://code.google.com/archive/p/word2vec/

	Algorithm	All annotations		Cross-validation	
		acc	κ	acc	κ
(a)	PMI 1M$_6$	0.297	0.203		
(b)	PMI 1M$_7$	0.302	0.208		
(c)	AvgMaxSim$_{10}$	0.345	**0.258***		
(d)	Perceptron trained on author's examples	0.351	0.264		
(e)	Perceptron trained on author's annotations			0.419	**0.330***
(f)	Perceptron trained on both			0.472	**0.402*****
(g)	AvgMaxSim$_{10}$ and perceptron trained on author's examples ($\alpha = 0.1$)	0.360	0.275		
(h)	MaxAvgMaxSim$_{10}$ using author's examples	0.384	0.301		
(i)	AvgMaxSim$_{10}$ and perceptron trained on both ($\alpha = 0.3$)			0.494	0.427

Table 1: **Classification results.** Mean accuracy and mean Kappa on author's annotations. Cross-validation method is leave-one-out on author's annotations. Boldface results are significant over previous best at $p < 0.05$ (*) and $p < 0.001$ (***).

The highest-performing algorithm overall (i) was a mixture model that combined our best unsupervised classifier (c) with the best supervised classifier (f), using a weighted average of scores of each approach. After tuning the mixture parameter α to 0.3 (30% of weight to the unsupervised model), algorithm (i) outperformed the best supervised approach (f), but gains were not significant.

MaxAvgMaxSim Model Although the mixture models did not yield significant gains, they gave us the idea to try a simpler approach, listed as algorithm (h). Here, we start with the existing $AvgMaxSim_{10}$ model (c), comparing the player input to the first 10 words of a given outcome. Then we apply the same $AvgMaxSim_{10}$ model to each of the fake examples provided by the author for the given outcome, and take the Max value of all of these scores as the overall score. Conceptually, this becomes a type of nearest-neighbor classifier, where the targets include both the first 10 words of each outcome and the author's fabricated examples. The performance of this algorithm (h) is the highest we were able to achieve without the annotation of actual user input for use in a supervised classifier.

These experiments gave us clear guidance for the adoption of new algorithms for the DINE system, which we integrated into the online platform. The default classifier is now the $MaxAvgMaxSim_{10}$ model (h) that matches user input to both the first 10 words of each outcome and the author's fake examples of expected user input. This unsupervised model provides good performance even when the author does not provide examples (c). When the author does provide examples the performance improves (h), giving the author some control over the behavior of the classifier beyond the tailoring of the beginning text of outcomes. Where higher classification performance is required, the author can hand-annotate real user input as it is collected over time, which can then be used to train a high-performing supervised classifier, deployed as a mixture model (i).

Discussion

From the player's perspective, the natural language interaction of DINE scenarios is a novel twist somewhere between the familiar traditions of branching storylines and parser-based interactive fiction. Writing out actions is somewhat more demanding for players than selecting among fixed options, but writing them in unconstrained natural language is somewhat easier than learning a specialized grammar. The more significant contributions of the DINE approach are apparent from the author's perspective, specifically stemming from the use of unsupervised text classification as the means of processing user input. By removing the requirement to collect and annotate input from real players, the unsupervised classification approach allows authors to quickly deploy their works and gather design-level feedback. Seeing that players miss important storyline paths or try unexpected actions, authors can freely modify the quantity, content, and organization of DINE pages and their outcomes. This is in contrast to supervised text classification technologies, where any modification to context or the set of classes risks invalidating any training data that has been annotated. Likewise, authors need not have any skills or aptitude for computer programming or in the training of machine learning algorithms. This allows non-technical single-person teams to craft and deploy complete interactive narrative works, where their efforts are directed entirely toward the familiar task of writing engaging fiction.

The results presented in this paper quantify the differences in accuracy between unsupervised and supervised text classification methods. As expected, the unsupervised methods do not yet achieve the performance of supervised machine learning—a gap that will only increase as supervised classifiers are provided with additional labeled training data. However, high coherence ratings from crowdsourced participants demonstrate that unsupervised methods can be sufficiently accurate to support the collection of real user input, bootstrapping the development of more accurate supervised models. By progressively moving from unsupervised to supervised classification models, developers of natural language interactive narratives can deploy coherent experiences that are playable from the start, and incrementally improve over time.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1560426. The projects or efforts depicted were or are sponsored by the U. S. Army. The content or information presented does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Church, K., and Harrison, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16:22–29.

Gandhe, S., and Traum, D. 2014. SAWDUST: a semi-automated wizard dialogue utterance selection tool for domain-independent large-domain dialogue. In *Proceedings of the15th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL 2014), Philadelphia, PA, June 18-20, 2014*.

Gandhe, S.; Gordon, A.; and Traum, D. 2006. Improving question-answering with linking dialogues. In *2006 International Conference on Intelligent User Interfaces, Sydney, Australia, Jan 29 - Feb 1, 2006*.

Gordon, A. S., and Swanson, R. 2009. Identifying personal stories in millions of weblog entries. In *Proceedings of the International Conference on Weblogs and Social Media, Data Challenge Workshop, San Jose, CA*.

Gordon, A. S.; Bejan, C.; and Sagae, K. 2011. Commonsense causal reasoning using millions of personal stories. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AAAI-11), San Francisco, CA*.

Gordon, A.; van Lent, M.; van Velsen, M.; Carpenter, P.; and Jhala, A. 2004. Branching storylines in virtual reality environments for leadership development. In *Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI-04), San Jose, CA*.

Hill, R.; Douglas, J.; Gordon, A.; and van Velsen, M. 2003. Guided conversations about leadership: Mentoring with movies and interactive characters. In *Proceedings of the Fifteenth Innovative Applications of Artificial Intelligence Conference (IAAI-03) August 12-14, 2003, Acapulco, Mexico*.

Mateas, M., and Stern, A. 2003. Integrating plot, character and natural language processing in the interactive drama facade. In *Proceedings of Technologies for Interactive Digital Storytelling and Entertainment (TIDSE), Darmstadt, Germany*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, 3111–3119. USA: Curran Associates Inc.

Montfort, N. 2003. *Twisty Little Passages: An Approach to Interactive Fiction*. MIT Press.

Noreen, E. W. 1989. *Computer intensive methods for hypothesis testing: An introduction*. New York: Wiley.

Packard, E. 1979. *The Cave of Time*. New York: Bantum Books.

Rickel, J.; Marsella, S.; Gratch, J.; Hill, R.; Traum, D. R.; and Swartout, W. 2002. Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems* 17:32–38.

Roemmele, M., and Gordon, A. S. 2015. Creative help: A story writing assistant. In *Proceedings of the 8th International Conference on Interactive Digital Storytelling (ICIDS), Copenhagen, Denmark*.

Roemmele, M.; Bejan, C.; and Gordon, A. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proceedings of the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning, Stanford University*.

Swanson, R., and Gordon, A. S. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Trans. Interact. Intell. Syst.* 2(3):16:1–16:35.

Traum, D.; Jones, A.; Hays, K.; Maio, H.; Alexander, O.; Artstein, R.; Debevec, P.; Gainer, A.; Georgila, K.; Haase, K.; Jungblut, K.; Leuski, A.; Smith, S.; and Swartout, W. 2015. New dimensions in testimony: Digitally preserving a holocaust survivor's interactive storytelling. In *Proceedings of the 8th International Conference on Interactive Digital Storytelling (ICIDS), Copenhagen, Denmark*.