Automated Story Selection for Color Commentary in Sports

Greg Lee, Vadim Bulitko and Elliot A. Ludvig

Abstract—Automated sports commentary is a form of automated narrative. Sports commentary exists to keep the viewer informed and entertained. One way to entertain the viewer is by telling brief stories relevant to the game in progress. We introduce a system called the Sports Commentary Recommendation System (SCoReS) that can automatically suggest stories for commentators to tell during games. Through several user studies, we compared commentary using SCoReS to three other types of commentary and show that SCoReS adds significantly to the broadcast across several enjoyment metrics. We also collected interview data from professional sports commentators who positively evaluated a demonstration of the system. We conclude that SCoReS can be a useful broadcast tool, effective at selecting stories that add to the enjoyment and watchability of sports. SCoReS is a step toward automating sports commentary and, thus, automating narrative.

Index Terms—Artificial Intelligence, Information Retrieval, Automated Narrative.

I. INTRODUCTION

Sports broadcasting is a billion-dollar industry. Most professional sports are broadcast to the public on television, reaching millions of homes. The television experience differs in many ways from the live viewing experience, most significantly through its commentary.

Much research has been done into the importance of commentary during sports broadcasting. When watching a game on television, "...the words of the commentator are often given most attention" [9]. The commentary has the effect of drawing the attention of the viewer to the parts of the picture that merit closer attention [15], an effect called italicizing [2]. Commentary can also set a mood during a broadcast. A commentator who creates a hostile atmosphere during a broadcast often makes the viewing experience more enjoyable for the viewer [5]. The descriptions given in a broadcast are so useful that fans often bring radios to live games in order to listen to the interpretations of the commentators [23]. Also, some sporting venues now support a handheld video device that provides the spectator with in-game commentary [29].

The purpose of commentators is to help the viewer follow the game and to add to its entertainment value. One way to add entertainment to a broadcast is to tell interesting, relevant stories from the sport’s past. The sport of baseball is particularly suited to storytelling. Baseball is one of the oldest professional sports in North America, existing since 1876. This longevity provides a rich history from which to draw interesting stories. The more popular commentators are known as “storytellers” [32], as they augment the games they call by adding stories that connect baseball’s past to its present. One of these “storytellers”, Vin Scully, has been commentating for Brooklyn/Los Angeles Dodgers games since 1950 and has been voted the “most memorable personality in the history of the franchise”.

A typical baseball game lasts about three hours, but contains only ten minutes of action, where the ball is live and something is happening on the playing field. This leaves two hours and fifty minutes of time where little is happening on the playing field, and it is the job of the commentators to entertain the viewer. This downtime is a good time to tell stories [30]. Baseball is also known for being a statistically dense league, and being able to match statistics from the current game state to a past situation in baseball adds to the commentators’ ability to entertain the viewer.

To illustrate, consider the case where one baseball team is trailing by four runs in the ninth inning. As this is not a particularly interesting situation, it may be a good time for a story. An appropriate story might be that of the Los Angeles Dodgers, who on September 18, 2006, were also trailing by four runs in the bottom of the ninth inning. The Dodgers hit four consecutive home runs to tie the game. The broadcast team could tell this story to the viewer, because the situations are similar. Thus, what is needed is a mapping from a game state (a particular point in a game) to an appropriate story.

Sports storytelling is a form of narrative discourse. Narrative discourse is a creative activity that involves relaying a series of events in an interesting and entertaining manner. It is a recounting of contingent events from the past with one or more main characters. Automating narrative discourse is a challenging problem for Artificial Intelligence (AI) and a subject of much recent research [35]. Our hypothesis is that sports story selection can be automated with Artificial Intelligence. Specifically, we set out to test whether an AI approach can be developed that maps game states to relevant stories, thereby significantly increasing audiences’ enjoyment of the broadcast. To achieve this goal, we develop an AI system that tells stories in the context of baseball. The Sports Commentary Recommendation System (SCoReS) learns offline to connect sports stories to game states, provided with some scored examples of story-game state pairs. This learned mapping is then used during baseball games to suggest relevant stories to a (human) broadcast team or, in the absence of a broadcast team (e.g., in a sports video game), to autonomously output a relevant story to the audience. In the case of suggesting stories to commentators, SCoReS is an example of human-computer...
interaction [8], as it is a computer system that communicates to humans information that will improve their performance.

This paper makes the following contributions to Computational Narrative. First, we formalize story-based sports commentary as a mathematical problem. Second, we use machine-learning methods and information retrieval techniques to solve the problem. This solution is for the specific domain of sports story selection, but is also general enough to be used in other domains involving story selection. Third, we implement the approach in the domain of baseball, evaluate the resulting AI system by observing feedback from human participants and show that it is effective in performing two separate tasks: i) automating sports commentary, and thus automating narrative in a special case, and ii) assisting human commentators. That is, we show that our combination of information retrieval techniques is able to map previously unseen baseball game states to stories in a sufficiently effective manner to improve the enjoyment of baseball broadcasts, increase interest in watching baseball and suggest stories to professional commentators that they would tell during a live game.

The rest of the paper is organized as follows. We first describe colour commentary in detail, and formulate the problem of mapping sports game states to interesting stories. This is followed by a review of research related to story-based commentary. Next, we describe information retrieval techniques in detail and describe our approach to mapping game states to stories. This approach combines information retrieval techniques designed to rank stories based on a given game state, and others that ensure the higher-ranked stories are indeed relevant to said game state. We then describe empirical work performed to choose a story ranker and use this ranker to select stories for baseball broadcasts. The quality of the story mapping is evaluated with user studies and demonstrations in a special case, and ii) assisting human commentators. That is, we show that our combination of information retrieval techniques is able to map previously unseen baseball game states to stories in a sufficiently effective manner to improve the enjoyment of baseball broadcasts, increase interest in watching baseball and suggest stories to professional commentators that they would tell during a live game.

Another manner in which the colour commentator adds to a broadcast is by giving background information on the players involved in the game. Whereas play-by-play commentators can provide a player’s statistics, colour commentators tend to add more personal information about a player, possibly including their own interactions with that player. Passing on their expertise in the sport at hand is one more way colour commentators add to the broadcast.

Another rich aspect of colour commentary is storytelling, which is our focus. Effective storytelling in sports broadcasts involves telling a story that is interesting to the audience, and that is related to what is actually happening in the game being broadcast. While play-by-play commentators are generally trained journalists, colour commentators are typically former professional athletes or coaches. As former members of the league being shown, they are thought to bring a level of expertise to the broadcast. Because they actually played or coached in the league, colour commentators tend to tell stories from their own experiences in the game. This gives the audience a first-hand account of snippets of baseball history, which can be quite entertaining. Unfortunately, colour commentators do not have first-hand knowledge of most of baseball history. Their knowledge of stories, however vast for human beings, is still limited relative to the total set of baseball stories available and they cannot necessarily connect the stories they do know to the game state at hand. This is where AI can be of assistance. Computers can both store many more stories than a human brain as well as quickly compute the quality of a match between each story in the library and the game state at hand. The problem we are addressing in this paper is to tell interesting stories as live commentary to a sports game.

III. RELATED WORK

In this section, we review existing research relevant to the problem we are solving. Some approaches deliver stories to non-sports video games (Section III-A), and others deliver live play-by-play commentary, with some added colour (Section III-B). None of the existing work delivers story-based colour commentary to a live game.

A. Storytelling in an Unpredictable Environment

Narrative generation explicitly attempts to automate storytelling by having computers create a story with an entertaining plot and characters. The ability to generate effective narrative is important in different applications, such as entertainment, training and education [27], and this area has been studied for over 20 years [13]. There has been much work on automated storytelling in non-sports video games [28]. Some systems generate or adapt a story due to actions taken by the player, creating a so-called interactive drama. Automated Story Director (ASD) [26] is an experience manager that accepts two inputs: an exemplar narrative that encodes all the desired experiences for the player and domain theory encoding the rules of the environment. As the player can influence the narrative, ASD adapts the exemplar narrative to ensure coherence of the story and a feeling of agency for the player. These adaptations are pre-planned based on the domain theory.
Systems such as ASD actively change the course of the game based on user actions. As a commentary system cannot alter what happens in a live sports game (real or video game), these systems are not directly applicable to our problem.

Human commentators try to weave a narrative with a coherent plot through unpredictable sports games [30]. Plot is a difficult narrative dimension for the broadcast team, as they do not know what the end result of the game will be until it concludes. They can, however, attempt to predict what will happen and begin to build up a plot that fits with the predicted ending. This brings into play different themes that the broadcaster can choose to follow, and he or she can follow more than one at a time (hedging his or her bets, so to speak).

In the broad scheme of narratives, the range of themes spans all of human experience. In baseball, these themes are limited, and they include: Spirit, Human Interest, Pity, Gamesmanship, Old-College-Try, and Glory [30], [6]. Matching game states and stories thematically (or categorically) can be used to help gauge similarity between the two and we investigate this in our approach.

B. Automated Commentating in Sports

Statsheet [1] and Narrative Science [10] automatically write previews for sports games that have not yet happened, and summaries about sports games that have already happened. For summaries, they are provided with statistics from a completed game and compose a narrative about the game, with the goal being to provide an interesting summary of game events. For previews, they are provided with statistics from past games, and compose a preview for the game that should entice the reader to watch said game. Neither Statsheet nor Narrative Science operate with live game data and both are unable to solve our problem of providing live stories during a game. They may, however, provide an additional potential database of stories about past games to augment our current system.

Modern commercial sports video games typically employ professional broadcast teams from television to provide commentary for the action in the game. This involves pre-recording large amounts of voice data that will be reusable for many different games. As an example, a clip to the effect of “Now that was a big out!” could be recorded and used in multiple situations where one team was in dire need of retiring a batter or runner. Recorded clips often use pronouns (i.e., “he” and “they”) so that they are not specific to any particular player or team. This makes the commentary generic, which reduces the amount of recording required, as recording voice data can be time consuming and expensive. Unfortunately, generic commentary is less exciting to the audience. We would like our AI system to deliver colourful commentary tailored to the current game by mapping the current game state to a story chosen specifically for said game state.

The MLB: The Show [31] suite of games is often considered to be at the leading edge of baseball video games [14]. Recorded clips of Matt Vasgersian of the Major League Baseball (MLB) network act as the play-by-play commentary while Dave Campbell of ESPN’s Baseball Tonight is the colour commentator. Most of the colour commentary involves analysis of what has happened on the field and indirect suggestions to the player as to how to improve their play. As far as we have seen, there is no storytelling in these games. The problem we are trying to solve, automated storytelling in sports, could provide an additional angle to the colour commentary in these games.

In MLB 2K7, another leading baseball video game, Jon Miller and Joe Morgan of ESPN: Sunday Night Baseball provide the play-by-play and colour commentary, respectively. There is at least one instance of relating a game state to a story from baseball’s past, occurring when a third strike is dropped by the catcher and he has to throw to first base to record the out. Here, Miller reminds us of a 1941 World Series game where Mickey Owens committed an error on a similar play, changing the course of the series. There appear to be a very limited number of stories told in the game, however, and a single story is used repeatedly. The mapping of the game state to the story appears to be hand-coded and based on just one feature (a dropped third strike). This limitation suggests that an important consideration for building an automated story selection system is to avoid repetition.

Robot World-Cup Soccer (RoboCup) is a research testbed involving robots playing soccer [16]. There is also a RoboCup simulation league, where the games are not physically played, but are simulated on a computer. Both the physical and simulation leagues provide researchers with a standard testbed in which to evaluate their AI strategies for various goals. In this context, previous work in automated commentary has focused primarily on automated play-by-play commentary. Byrne, Rocco and MIKE [4] are three systems that produce automated play-by-play commentary for RoboCup simulator league games. The three systems obtain their data from the Soccer Server [16], which summarizes the gameplay’s main features – the player locations and orientations, the ball location and the score. Each system generates natural language templates, filling in player and team names where appropriate, then uses text-to-speech software to verbalize the derived commentary. Dynamic Engaging Intelligent Reporter Agent (DEIRA) is a similar system to Byrne, Rocco, and MIKE, as it performs the same task, but in the sport of horse racing.

There are some attempts within these systems to provide colour commentary, but none go as far as to try to incorporate storytelling. That is, these systems tackle a problem different from what our work tackles – they automate factual commentary with some bias added, but do not implement colour commentary via stories. Our system could be used in conjunction with each of these automated play-by-play systems to create fully automated commentary, featuring both play-by-play and colour.

Within its live online game summaries, Major League Baseball uses a system called SCOUT [17] that provides textual analysis of the current game. The viewer is shown information such as the type of pitches thrown during an at-bat and the tendencies of the batter with respect to the pitcher. While SCOUT provides some colour, it is mostly a statistical summary and currently does not tell stories. Our system could extend SCOUT by adding stories to MLB online game summaries.
Freytag [12] identifies a five-stage pyramid of dramatic structure: exposition, rising action, falling action and denouement. Rhodes [25] describes a system that follows this pyramid and adds dramatic commentary to sports video games by using language with different levels of emotional connotation depending upon the game situation. Values for the different motifs (themes) [6] are stored, and a vocabulary for each that changes as their level of intensity increases. Each theme has a set of actions that can occur in the game that either increase its intensity (Freytag’s rising action) or decrease its intensity (Freytag’s falling action). One theme within the system is “Urgency”, with some level 1 phrases being “looking shaky” and “fortune not on their side”, and level 3 phrases being “doomed” and “beyond salvation”. Lexicalised Tree-Adjoining Grammar is used to generate comments. While drama is added to the commentary with this program, it is an augmentation to the play-by-play commentary more so than an addition of colour commentary. Our system could further augment the dramatic commentary by adding relevant stories.

IV. PROPOSED APPROACH

In this section, we present an AI approach to solving the problem of delivering story-based colour commentary to a live baseball game. We start by framing the problem as an information retrieval problem (Section IV-A). We then describe the machine-learning techniques we used (Sections IV-B – IV-C) and finally combine these techniques in Section IV-D.

A. Information Retrieval Framework

We approach the problem of automated story selection in sports as an information retrieval problem. In the context of sports, the game state for which we are seeking an appropriate story is treated as the query, while the candidate stories returned by the system are the documents. Our system’s goal then, is given a game state, to return to the user a ranked list of stories, based on how appropriate they are for the game state. We assume that the game state is available live, such as is the case for Major League Baseball [22]. We also assume that a database of stories has previously been collected.

Thus, the problem is to retrieve stories most appropriate for game states during live sports broadcasts. Once a story database has been obtained, a system must learn to match the stories to game states. The broader the story database, the more likely an appropriate story can be found that matches any given game state. As “being interesting” is an informal and subjective measure, we evaluate the quality of the mapping by incorporating the selected stories into a simulated broadcast and test the enjoyment of viewers and the interest of professional commentators.

To compare stories to game states, we extract features from both, such as the score, the teams involved and what type of action is happening on the field (i.e., a home run). Formally, the game state is a vector of $n$ numeric features: $\vec{g} = (g_1, g_2, \ldots g_n)$. To illustrate: binary feature $g_1$ may be 1 if in game state $\vec{g}$ there is a runner on first base. Integer feature $g_2$ can be the current inning. Similarly, a baseball story can be described with a vector of $p$ numeric features: $\vec{s} = (s_1, s_2, \ldots s_p)$. Binary feature $s_1$ can be 1 if the story involves a runner on first base and integer feature $s_2$ can be the inning number mentioned in the story. The task is then to map $\vec{g}$ to a relevant and interesting $\vec{s}$.

The match quality $D(\vec{g}, \vec{s})$ between a game state $\vec{g} = (g_1, g_2, \ldots g_n)$ and story $\vec{s} = (s_1, s_2, \ldots s_p)$ is scored on a 5-point integer scale ranging from 0 for a completely inappropriate match to 4 for a perfect match. Thus, the problem is given a game state $\vec{g}$, to retrieve a story $\vec{s}$ of the highest possible match quality $D(\vec{g}, \vec{s})$.

B. Training Data

We solved this problem by using IR techniques and machine learning. Rather than separately feed the game state and story features to IR algorithms, we made the connection between corresponding features more explicit. A similarity vector $\vec{c}$ was computed for a game state specified by feature vector $\vec{g}$ and a story specified by feature vector $\vec{s}$. Each component of vector $\vec{c}$ is the result of comparing one or more features of $\vec{g}$ to one or more relevant features of $\vec{s}$. Logical connectives are used to compare binary features. For instance, to match the runner on first base features, the biconditional over the corresponding features is used: $c = g \leftrightarrow s = 1 \leftrightarrow 1 = 1$. For non-binary features feature-specific functions are used. For instance, when comparing the current inning number $g$ and the inning number involved in a story $s$, the similarity feature $c$ is calculated as $(8 - |g - s|)/8$, where values closer to 1 indicate a closer pairing of inning features. Another example is the marquee matchup feature valued between 0 and 1. It indicates how well the story and game state match in terms of a strong hitter and a strong pitcher being involved. This feature is calculated by combining several statistical features for the batter and pitcher in $\vec{g}$ with a story category feature in $\vec{s}$ [19].

The similarity vector $\vec{c}$ indicates how related $\vec{g}$ and $\vec{s}$ are, but does not provide a scalar value. What is needed is a way to map $\vec{c}$ to $D(\vec{g}, \vec{s})$ — the 5-point-scale quality of the match between $\vec{g}$ and $\vec{s}$. Machine-learning techniques are used to create this mapping from training data $T$.

To build this training data set, a set of $m$ game state vectors $G = \{\vec{g}_1, \ldots, \vec{g}_m\}$ is taken to form a set of $m \times p$ similarity vectors $\vec{c}$ for all $\vec{g}_i \in G$ and all $p$ story vectors from our story vector library $S = \{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_p\}$. Each similarity vector is then labeled with the ground-truth value of the quality of match between the corresponding game state and the story. Mathematically: $T = \{\langle \vec{c}, D(\vec{g}, \vec{s}) \rangle | \vec{g} \in G, \vec{s} \in S, \vec{c} \text{ is the similarity vector for } \vec{s} \text{ and } \vec{g} \}$. For simplicity’s sake in the rest of the paper, we refer to $S$ as the story library rather than the story vector library and $G$ as the game state library, rather than the game state vector library. Also, as game states and stories are the equivalent of queries and documents in this work, we will use the former terms in the rest of the paper.

C. SCoReS Offline

IR algorithms are generally divided into three groups – pointwise, pairwise and listwise [21]. Pointwise algorithms are regression and classification algorithms, with mean-squared error typically used as an error function. Pairwise algorithms
perform an incomplete ordering on the data. Listwise algorithms make direct use of IR metrics to search for a good ranking of documents.

Our approach to selecting stories for game states is a hybrid approach that uses a listwise algorithm to rank stories and a pointwise algorithm to evaluate the top-ranked story. This two-step process involves a Ranker and an Evaluator, as shown in Figure 1. In this section, we describe the offline training and creation of the two elements of SCoReS, and then describe the online operation in detail (Section IV-D).

1) Machine learning a Ranker: For the ranking algorithm, we adapted AdaRank [34], a listwise algorithm based on AdaBoost [11]. AdaRank forms a “strong” ranker by iteratively selecting and combining “weak” rankers. A weak ranker uses a single component of the similarity vector $\vec{c}$ to rank (i.e., sort) the training data $T$. Each weak ranker has a “vote” on the final ranking, based on how its ordering of $S$ over the training data was scored according to a chosen IR scoring metric.

SCoReS AdaRank (Algorithm 1) accepts as input a set of training data $T$, an IR scoring function $F$, the number of game states $m$ in $T$, the number of weak rankers to compose the strong ranker $k$, and the number of tie-breaking features to use $y$. In line 1, SCoReS AdaRank first partitions $T$ by its $m$ constituent games states. This is done because stories can be meaningfully sorted by the match quality values $D$ only for a given game state. The ground truth rankings $\mathcal{T}$ are then calculated (line 2) for possible use in evaluating weak rankers (line 12). All weights are initialized to $1/m$ (line 3) as all game states are initially equally important. The main ranker $r$ and its corresponding confidence values $A$ are initialized to be empty sets (line 4). The set of feature combinations to be considered for use in weak rankers, $B$, is calculated based on the number of features in each $\vec{c}$ in $T$, and the number of features to use for tie-breaking $y$ (line 5).

At each iteration of SCoReS AdaRank, elements $b$ of $B$ whose first elements have not yet been used in $R$ are considered as possible weak rankers (lines 6-15). Thus, each feature of $\vec{c}$ may only be used once as the main sorter in a weak ranker. For each game state, the weighted score $v$ of sorting $T_i$ by $b$ (with any remaining ties broken randomly) is calculated, using the scoring function $F$ and current weights $w$ (lines 10-12). The arguments to $F$ vary based on which scoring metric is used, but all metrics we consider accept $\theta_i$, the match qualities for $T_i$ as input. If the mean weighted score $v$ for $b$ is greater than the maximum encountered so far in this iteration, the feature combination to be used in the weak ranker $r$ for this iteration is set to $b$ (lines 13-15). After evaluating all valid elements of $B$, the best weak ranker for this iteration is added to the main ranker $R$ (line 16) and $A$ and $w$ are updated (lines 18-19) as in [34]. The data is re-weighted after each iteration so that game states for which stories have been poorly ordered are given more weight (line 19).

As a small, concrete example, let $F$ be Normalized Discounted Cumulative Gain (NDCG) [34], $m = 3$, $k = 2$, and $y = 1$. Let the similarity vectors $\vec{c}$ in $T$ consist of the following features: the runner on first base feature ($c_1$), the strikeout feature ($c_2$), the inning feature ($c_3$), and the marquee matchup feature ($c_4$). The first two are binary: they are 1 if both the story and game state involve a runner on first base (or both involve a strikeout), and 0 otherwise. The marquee matchup and inning features are computed as previously described.
Table I

<table>
<thead>
<tr>
<th>Game State 1</th>
<th>(w_1 = 1/3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story</td>
<td>(c_1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Game State 2</th>
<th>(w_2 = 1/3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story</td>
<td>(c_1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Game State 3</th>
<th>(w_3 = 1/3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story</td>
<td>(c_1)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

The ordering of \(\mathbf{T}\) after being sorted by feature combination \((c_3, c_4)\). Random tie-breaking was necessary for game state 2 (between stories 1 and 2).

Table I shows possible training data split into its 3 constituent game states. The weight of each game is initially set to 1/3. The best feature combination for the first iteration is \((c_3, c_4)\) — the runner on first base feature to be used as the main sortor, with the strikeout feature as a tiebreaker. Sorting by this feature combination yields the orderings shown in Table II. Assuming we consider the top 3 ranking positions relevant, the NDCG scores for each game state in this ordering would be \((0.97, 0.59, 0.89)\), with weighted mean \(\mu = 0.81\) (as all weights are equal).

The weak ranker \(r\) for this iteration would thus be \((c_3, c_4)\) and \(c_3\) would not be considered as a main sortor in weak rankers in later iterations (as it is the main sortor here). \(\alpha\) is calculated with the formula (from [34]):

\[
\alpha = \frac{1}{2} \cdot \ln \frac{\sum_{i=1}^{m} w(i) \left(1 + F(\theta_i, 3, \theta_i') \right)}{\sum_{i=1}^{m} w(i) \left(1 - F(\theta_i, 3, \theta_i') \right)}
\]

where \(\theta_i\) is the vector of match qualities \(D\) for the ordering of game state \(i\) by \(r\), and \(\theta_i'\) is the vector of match qualities \(D\) for the ground truth. \(\alpha\) for this iteration would thus be 1.13. The weights for each game state are then updated with the formula (from [34]):

\[
w(i) = \frac{e^{-F(\theta_i, 3, \theta_i')}}{\sum_{j=1}^{m} e^{-F(\theta_j, 3, \theta_j')}}
\]

giving the new \(w = (0.28, 0.41, 0.32)\).

On the second iteration, game state 2 has more weight, as the weak (and main) ranker \((c_3, c_4)\) failed to sort it as well as it did game states 1 and 3. This leads to \(r = (c_1, c_4)\), \(\mu = 0.73\), \(\alpha = 1.01\), and thus main ranker \(R = ((c_3, c_4), (c_1, c_4))\) and \(A = (1.13, 1.01)\). After normalizing, \(A = (0.53, 0.47)\).

This means the weak ranker using the inning feature to sort and the marquee matchup feature to break ties gets 53% of the vote for ranking \(S\), and the runner on first base feature combined with the marquee matchup feature tiebreaker gets 47% of the vote for the ranking output by SCoReS AdaRank.

2) Machine learning an Evaluator: Though the Ranker \(R\) output by SCoReS AdaRank provides a ranked list of stories, it does not provide a value for \(D\) for these stories. Thus, there is always a top-ranked story for a game state, but SCoReS AdaRank provides no indication as to how “good” the top ranked story is. We added an Evaluator to provide an estimate of \(D\). The Evaluator can then be used as a threshold to ensure the top-ranked SCoReS AdaRank story is worth telling.

In principle, it is possible to use the Evaluator on its own to rank stories. The reason we do not do this is because of its pointwise nature; it uses mean squared error (MSE) as a scoring metric, which treats each datum equally – there is no preference with respect to accuracy towards the top of a ranked list. Thus the accuracy at the bottom of the ranked list is regarded as being as important as the top of the list, making using the Evaluator as a ranker less desirable than the Ranker – Evaluator combination.

Algorithm 2 SCoReS. Input: \(G'\): game states for a live game, \(S\): our story library, \(R\): ranker from SCoReS AdaRank, \(A\): weak ranker confidences for \(R\), \(E\): evaluator, \(t\) threshold for evaluator.

1: for each game state \(g\) in \(G'\) do
2: \(\text{if should tell a story in } g\) then
3: \(\text{create } \{\vec{e}\}, \text{ comparing } g\) to \(\vec{s} \in S\)
4: \(\text{rank } \{\vec{e}\} \text{ with } R \text{ and } A\)
5: \(s_{\vec{e}} \leftarrow \text{top ranked story}\)
6: \(\text{if } E(s_{\vec{e}}) \geq t\)
7: \(\text{output } s_{\vec{e}} \text{ to broadcast team (or viewer)}\)

D. SCoReS Online

Our SCoReS system thus consists of a Ranker (learned by SCoReS AdaRank) and an Evaluator (learned by a pointwise algorithm) as shown in Algorithm 2. SCoReS processes each game state in a given (novel) game \(G'\) (line 1). If the current game state \(g\) is appropriate for a story we create similarity vectors of \(g\) and each story in \(S\) (line 3). Generally, any game state is appropriate to suggest a story to human commentators, but in the case of SCoReS operating autonomously (as in video games), we limit the number of stories output to the
viewer by only allowing stories to be told after an appropriate “downtime” (typically 10 – 20 pitches). The similarity vectors are sorted with the ranker $R$ and confidences $A$ learned by SCoReS AdaRank offline (line 4). The top ranked story is extracted in line 5 and then scored by the Evaluator $E$ in line 6. If the story passes the provided threshold $t$, and SCoReS is operating autonomously, the story is output to the viewer (line 7). If a broadcast team is using SCoReS, the top few stories can be output.

V. Empirical Evaluation

To evaluate the quality of SCoReS, we conducted a series of empirical studies where we asked potential users to evaluate the system. We examined several possible applications of SCoReS: providing generic commentary, adding stories to existing commentary, picking context appropriate stories, and assisting professional commentators to select stories. These results were presented in [20].

We conducted several user studies to evaluate whether SCoReS improved broadcast quality in any of these applications. The first user study was conducted to ensure commentary was beneficial within our game library before we even tried to improve said commentary. Subsequent user studies tested whether adding stories to a broadcast within our game library helps and whether intelligently selecting stories based on the game context works better than simply adding any story to the broadcast. A demonstration of the SCoReS system to professional commentators was also carried out. Their feedback is an estimate of whether SCoReS can be successfully deployed in a professional broadcast setting.

These experiments evaluated SCoReS in both its modes of operation: as an autonomous commentary tool within the user studies, and as an assistant to a human colour commentator in the interviews with professional commentators. In the user studies, we inserted SCoReS’ top-ranked story into video clips from actual baseball games. We chose to use actual games because watching clips from a video game would likely not be interesting to the participants, inducing boredom. Note that improving commentary from actual games appears to be a harder problem than improving sports video game commentary, as commentary in video games is pre-recorded and generic, whereas commentary during an actual game is tailored to that particular game.

In order to build training data for SCoReS, we first downloaded MLB game statistics from MLB’s XML site [22], with permission from MLB. For all experiments, the game state and story features were kept constant. The feature vector for each game state included statistics such as the score, the inning, the date, the teams involved and various statistics for the current batter and pitcher (such as home runs and wins). Statistics for the previous pitch’s batter and pitcher were also present because stories are often told in reference to an event preceding the current game state. Details are found in [19].

One hundred and ten stories were gathered from “Rob Neyer’s Big Book of Baseball Legends” [24], “Baseball Eccentrics” [18], and Wikipedia. Stories ranged in year from 1903 to 2005. Feature selection and categorization of the stories were done by hand. Story features were similar to game state features, but also included the category. Using our baseball expertise, and based on the features available from MLB’s XML site, we chose the 10 categories listed in Table III. While these categories are not those used in other work, we thought they were appropriate given the available data (i.e., it would be difficult to categorize a game state as “the Lucky Break Victory” as in [30] without video of the game, using only selected statistics).

Building similarity vectors $\vec{c}$ created a new feature set, consisting mostly of binary features (as described in Section IV-B), which were home run, sacrifice, single, double, triple, double play, strikeout, fly out, pop out, ground out, walk, intentional walk, hit by pitch, substitution, runner on first, runner on second, runner on third, one team (do $\vec{g}$ and $\vec{s}$ have at least one team in common), and two teams. Confidence similarity features (how well $\vec{g}$ and $\vec{s}$ match in terms of a category) included marquee matchup, great statistics for batter or pitcher, bad statistics for batter, bad statistics for pitcher, opening of inning, important games from history, big finish, blow out and home run hitter in a one-run game. Other similarity features scaled between 0 and 1 included balls, strikes, outs, inning, run difference and month.

A. Choosing a Ranker

In order to choose a Ranker and an Evaluator for SCoReS, we performed a leave-one-out cross-validation experiment. Training data consisted of 40 randomly selected game states from the 2008 MLB season and the 110 stories gathered from various sources. At each cross-validation fold, 4290 data (39 game states $\times$ 110 stories) were used to build training data $T$, while 110 (1 game state $\times$ 110 stories) were used for testing data. The same story database was used in training and testing. Candidate Evaluators were trained on $T$ and then used within SCoReS at each fold.

For IR scoring metrics, $F$, we considered the following: NDCG, Mean Average Precision (MAP), Winner Takes All (WTA) [34], Expected Reciprocal Rank (ERR) [7], and Return Score (RS) a metric that simply returns the match quality of the top ranked story. We tested the cross product of this set of scoring metrics, $k \in \{1, 2, 3\}$ and $y \in \{0, 1, \ldots, 25\}$. We also tested higher values of $k$ with a reduced set $y$, including the cross product of the full set of scoring metrics, $k \in \{4, 5\}$.

### Table III

<table>
<thead>
<tr>
<th>ID</th>
<th>Story Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Opening of Inning</td>
</tr>
<tr>
<td>2</td>
<td>Bad Statistics for Batter</td>
</tr>
<tr>
<td>3</td>
<td>Marquee Matchup</td>
</tr>
<tr>
<td>4</td>
<td>Great Statistics for One Player</td>
</tr>
<tr>
<td>5</td>
<td>Bad Statistics for either Hitter or Pitcher</td>
</tr>
<tr>
<td>6</td>
<td>Important Games from History</td>
</tr>
<tr>
<td>7</td>
<td>Big Finish</td>
</tr>
<tr>
<td>8</td>
<td>Blowout or Comeback</td>
</tr>
<tr>
<td>9</td>
<td>One-run Game, Home Run Hitter Batting</td>
</tr>
<tr>
<td>10</td>
<td>Human Interest</td>
</tr>
</tbody>
</table>

The story and game state categories used in our experiments.
and \( y \in \{0, 1, \ldots, 4\} \), and the cross product of the full set of scoring metrics, \( k \in \{6, 7\} \) and \( y \in \{0, 1\} \).

Cross-validation chooses the set of parameters that is best at predicting the score of unseen game state-story pairs. The best performing instantiation of SCoReS had a Decision Tree as the Evaluator. The best set of parameters for AdaRank found using this process was \((F = \text{NDCG}, k = 7, y = 0)\). To choose the actual Ranker for SCoReS we provided all 4400 training data \( T \) to SCoReS AdaRank. This produced the Ranker shown in Table IV.

This combination of Ranker and Evaluator output stories that averaged a 3.35 match quality. Stories were only output for 23 of the 40 folds, because in 17 folds, the Evaluator (Decision Tree) deemed the story chosen by SCoReS AdaRank to be of insufficient quality to output. As a comparison, a perfect selector (based on the ground truth labelings, done by hand by a domain expert) would output stories with a 3.7 average match quality, outputting a story for all 40 folds. The Decision Tree operating on its own output stories that averaged a 2.3 match quality, outputting a story for all 40 folds. Thus it does not provide as good a ranking as the Ranker, but does provide a scalar value within the hybrid approach that is used as a threshold to determine which stories are output by SCoReS.

While these experiments show that SCoReS performs well with respect to outputting a story with a high match quality, what we are actually interested in is the enjoyment of viewers watching games containing commentary augmented by SCoReS, as well as how useful professional commentators think SCoReS would be to them while they are commenting on games. We also are interested in how well SCoReS chooses stories for game states beyond the 40 used in cross-validation. We thus made the transition from cross-validation experiments to user studies and interviews with professional commentators, using previously unseen game states.

### B. User Studies

The true measure of success for SCoReS is how viewers perceive the stories it selected in the context of the game. We conducted six user studies to test whether commentary improves a broadcast, whether inserting stories into a broadcast makes it more enjoyable, and whether the added stories need to be in the proper context to add to the broadcast. Each user study involved participants watching video clips from two AAA (minor league) baseball games: the July 15, 2009 AAA All-Star game between the International League and the Pacific Coast League, and the April 7, 2011 game between the Buffalo Bisons and Syracuse Chiefs. Each study involved three different types of commentary, depending upon which hypothesis we were testing. Each video clip was between three and six minutes in length, and they were always shown in chronological order. The order of the commentary, however, varied. After each clip, participants answered questions related to their enjoyment of the clip. At the end of the session, participants completed a background questionnaire.

#### 1) User Study I – The Need for Commentary:

In the first user study, we compared SCoReS Commentary to two different types of commentary. For No Commentary, we removed the commentary from the broadcast, and left the crowd noise. The Original Commentary had voiceovers for the original professional commentary, with no stories inserted or present in the original commentary. Voicing over the commentary was necessary as we needed to insert stories in some types of commentary, and needed the stories read in the same voice as the other commentary. We used voiceovers in all video clips for consistency. The SCoReS Commentary had a story selected by our system.

We recruited 16 participants from the local community. To measure the performance of the different types of commentary, we evaluated participants’ answers to the eight questions listed in Table V. For both games, each participant saw one clip each with Original Commentary, SCoReS Commentary, and No Commentary. Thus, each participant saw six video clips in total. For this experiment, SCoReS was trained on 35 game states and 88 stories, as we did not yet have our full training data. SCoReS output stories that averaged a 1.89 match quality, for 19 of the 35 cross-validation folds. As a comparison, a perfect selector would output stories with a 2.6 average match quality, outputting a story for all 35 folds.

The parameters for this experiment are shown in Table VI, with the Ranker being that shown in Table VII. Jim Prime, who did the play-by-play for this study, is an author of several baseball books, including Ted Williams’ Hit List, a book he co-authored with Ted Williams of the Boston Red Sox [33]. SCoReS had a database of 88 stories from which to choose stories for each game state. Figure 2 shows that SCoReS Commentary ranked significantly higher than No Commentary across all metrics. A one-tailed test was used to check for significance in the results of all experiments. 

The crowd noise was artificial as we could not remove commentary without removing the crowd noise. After removing all the sound from the original broadcast, we added an audio file of crowd noise. This held for all types of commentary.
After correcting for multiple comparisons with a Holm-Sidak adjustment, SCoReS Commentary ranked significantly higher than No Commentary across all metrics except “Viewing this clip made me more interested in participating in sports.” There were no significant differences between SCoReS Commentary and Original Commentary in this experiment.

2) User Study II - Baseball Fans prefer SCoReS: Having collected evidence that commentary itself adds entertainment to a game broadcast, we changed the user study format, replacing No Commentary with Mismatch Commentary. For Mismatch Commentary, we inserted a story in the same place as we would with SCoReS, but instead of the story being selected for that game state, it was actually a story chosen by SCoReS for a different game state, in the other game the participant saw. This manipulation kept the story pool consistent across the conditions thereby controlling for the overall level of interest.

User Study II differed further from User Study I in the following ways. An extra clip was added before the clips of interest because pilot data indicated that the first clip in a sequence was always rated more poorly. Participants were screened with the question “Are you a baseball fan? As baseball fans actually enjoy baseball games, we hypothesized that they would be likely to notice differences in commentary. Also, SCoReS is built to improve the enjoyment of a sport for a fan. If someone does not enjoy a sport to begin with, it may be difficult to change his or her mind. SCoReS had access to the full 4400 training data, and output the Ranker shown in Table IV. Parameters for this study are given in Table IX. Len Hawley, the play-by-play commentator for the Acadia University Varsity Hockey team did the play-by-play reading for these experiments. Clip ordering was counterbalanced across subjects, ensuring that each type of commentary appeared in each chronological position the same number of times for each game.

We recruited 39 students from the University of Alberta who were self-described baseball fans. For each of the two games, each participant saw one clip each with Original Commentary, SCoReS Commentary, and Mismatch Commentary. Figure 3 shows the mean difference between SCoReS Commentary and both Original Commentary and Mismatch Commentary for the second user study. SCoReS Commentary was ranked higher than the Original Commentary for the “Viewing this clip made me more interested in watching baseball” metric, with $p < 0.001$. This shows that adding stories to commentary can improve a broadcast. SCoReS Commentary was ranked higher than Mismatch Commentary for the “I found this viewing experience enjoyable” metric with $p < 0.01$. This shows that intelligently adding stories to commentary can be more enjoyable to the viewer than adding random stories. After correcting for multiple comparisons with a Holm-Sidak adjustment, $p < 0.05$ for both of these comparisons. Questions 6 – 8 from Table VIII are omitted from the graph as the questions were irrelevant to the Original Commentary case. Note that while there were no significant differences between Original Commentary and SCoReS Commentary in User Study I where there was no screening for baseball fans, there were significant differences in User Study II, where baseball fans were targeted.
C. Interviews with Commentators

In this experiment, we demonstrated SCoReS to professional commentators. To evaluate the potential usefulness of SCoReS, we first asked them, “Would you be interested in a system that suggests interesting stories during a game?” Then we demonstrated SCoReS delivering three stories for four different clips to the commentators. After each clip, we asked, “Would you tell any of the suggested stories?” The commentators could answer based on a synopsis, or choose to see the full story text. After the full demonstration we asked, “Would you be interested in this particular system?” The Ranker used within SCoReS to choose stories for the demonstration was the same as the Ranker used in the second user study (Table IV).

The four commentators were Mark Lee and Kevin Weekes, a play-by-play and colour commentator team for the Canadian Broadcasting Corporations’s (CBC’s) Hockey Night in Canada programme, Dan Robertson, play-by-play commentator for various sports (including baseball) for Eastlink Television, and Len Hawley.

All four commentators said they thought a system such as SCoReS would be a useful tool to have at their disposal. When asked about SCoReS itself, they all answered that it would be a great tool not only for baseball, but also for other sports, with a few tweaks. In particular, stories would need to be kept short for hockey broadcasts, which is a faster-paced sport.

Among the four clips shown to each commentator, a story suggested in two of them would have been told by Len Hawley and Dan Robertson with some minor changes. One of these was a story about a player having a successful day batting, the day after his divorce in late April, 1988. The game state where the story was told was during the April 7th Buffalo Bisons game, just after a player had his fourth hit and fifth run batted in of the game, thus, a successful game in his own right. Both commentators said they would have told the story if it had happened in early April. The Ranker within SCoReS did not make use of a day match feature, for this experiment (or have access to said feature). Thus, the story happening earlier in April would have had no effect on SCoReS, and this story could still have been output.

Mark Lee and Kevin Weekes also said in two of the clips shown that they would have told one of the stories presented by SCoReS, with no conditions on their answers. They offered several other insights as well. First, even if the stories suggested are not appropriate at the time of suggestion, once commentators have seen them, they can keep the stories in mind in case they are relevant later. As Mark Lee pointed out, on the two occasions he did not choose to tell a story, SCoReS suggested stories with a borderline match quality to the current game state, but if the game state were to change slightly, the match quality could become high enough for the story to warrant telling. Secondly, a system such as SCoReS would need to be completely integrated into a broadcast, and not simply used by the commenting team. This would allow the producers to add imagery to the broadcast, relevant to the story that the commentators choose to tell. Thirdly, while commentators do know many stories about the sport being broadcast, there are too many different parts of the game they must monitor and process in order to properly broadcast the game that they often cannot think up stories on their own. Thus SCoReS can be beneficial to the broadcasters not only in terms of suggesting stories they do not know, but also in terms of stories they do know, but would not have thought of and connected to the current game state. Lastly, while older stories may seem less relevant to younger viewers, they are actually quite relevant to viewers who watched games at the time the stories took place. Thus, SCoReS can help keep the interest of several generations of sports fans by connecting a sport’s past to its present.

VI. DISCUSSION AND CONCLUSION

We have shown that SCoReS has a statistically significant positive influence on the sports viewing experience across several metrics. National commentators Mark Lee and Kevin Weekes were particularly positive about the system, suggesting its appeal for a broad audience. This indicates that implementing SCoReS in professional commentating may lead to a better viewing experience.

A. Lessons Learned

Participants in the user studies were recruited from subject pools, and study participation made up a part of their grade. Thus, they did not watch the video clips at a time of their choosing, as opposed to a fan watching on television. A more effective way of evaluating SCoReS would be to have baseball fans evaluate the system at a time of their choosing, when they are interested in watching baseball.

Viewing three to six minute clips of games can make it difficult for a participant to gain context in the game, and thus make it difficult to appreciate an appropriately placed story. An ideal setup for SCoReS evaluation would be to have participants watch longer stretches of baseball games, so that they could better gauge whether a particular story should be told at a particular time.
Professional commentators generally state why stories they are telling are relevant, to give context to the story. This did not happen during the user studies, because we believed this would have biased participants’ answers to some of the questions. In hindsight, it may have been possible to state why a story was being told by mentioning the game features that led to the story’s selection, for both the SCoReS Commentary stories and the Mismatch Commentary stories.

Despite these challenges, SCoReS was able to achieve significant improvements in overall enjoyment and increasing interest in watching baseball, and we surmise that in a more realistic future deployment, SCoReS would further improve the entertainment value of sports broadcasts. Identifying these challenges and the ways to overcome them is one of the contributions of this project and we hope these will be used by future researchers.

B. Future Research

In the future, we would like to augment SCoReS with baseball facts, in addition to stories. These facts would be shorter bits of information from baseball’s past, that could be told quickly. Also, the system would benefit from an automated bot to perform several tasks. The bot could mine the web looking for relevant stories, providing SCoReS with an ever-growing set of stories from which to choose. The bot could also automatically extract features from the stories it finds, eliminating the need to do so by hand.

Adding more training data (both game states and stories) should also benefit SCoReS, by helping to create a more intelligent learner. A larger story database for SCoReS to choose from online should provide for a system with more flexibility, making it more likely there is a story in the database relevant to each game situation encountered. Data labelling is a time-consuming process that could be performed by multiple individuals (rather than just one domain expert), possibly making use of such services as Mechanical Turk [3]. Another source of training data is stories that were actually told by professional commentators. These matches provide one “good” example of a story for a game state, but no negative examples, as the commentator may have considered other stories relevant as well, even if he or she did not tell them.

C. Future Applications

SCoReS offers many possible future applications along the lines of fully automated commentary. Combining SCoReS with the systems described in Section III (such as Byrne) would yield a completely automated commentating system. Sports video games could increase their story databases, and then use SCoReS to select between these stories during gameplay. Automated storytelling systems such as Statsheet and Narrative Science could use SCoReS to automatically add stories to their own automatic recaps of games.

SCoReS could also be used to created personalized colour commentary. The viewing experience could be tailored to not just groups, but individuals, through web broadcasts. Features describing the viewer could be input into the system, and stories could be selected partly based on these features. Finally, SCoReS could be used for story selection in real-time strategy games such as StarCraft as they have their own sets of features and stories, and often contain commentary.

D. Conclusion

Storytelling is believed to be a cognitively rich and creative task. In order for humans to excel in storytelling, an innate aptitude and training are required. Skilled storytellers including writers, poets and colour commentators are recognized and famed. In this work, we took a step towards automating this task by building the first AI story selector for colour commentary in any sport. Its implementation in baseball was positively evaluated in user studies and by national level professional commentators. We believe this to be a contribution to the field of Artificial Intelligence with immediate practical applications.

REFERENCES