GENA: A Case-Based Approach to the Generation of Audio-visual Narratives

Santiago Ontañón[3], Josep Lluís Arcos[1], Josep Puyol-Gruart[1], Eusebio Carasusán[2], Daniel Giribet[2], David de la Cruz[1], Ismel Brito[1], and Carlos Lopez del Toro[1]

[1] IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra (Spain)
{arcos,puyol,davdela,ismel,clopez}@iiia.csic.es
[2] Televisió de Catalunya
Carrer de Jacint Verdaguer,
Sant Joan Despí, 08970 (Spain)
{ecarasusan.q,dgiribet.g}@tv3.cat
[3] Computer Science Department
Drexel University
Philadelphia, PA, 19104 (USA)
santi@cs.drexel.edu

Abstract. This paper presents GENA, a case-based reasoning system capable of generating audio-visual narratives by drawing from previously annotated content. Broadcast networks spend a large amount of resources in covering many events and many different types of audiences. However, it is not reasonable for them to cover smaller events or audiences, for which the cost would be greater than the potential benefits. For that reason, it is interesting to design systems that could automatically generate summaries, or personalized news shows for these smaller events or audiences. GENA was designed in collaboration with *Televisió de Catalunya* (the public Catalan broadcaster) precisely to address this problem. This paper describes GENA, and the techniques that were designed to address the complexities of the problem of generating audio-visual narrative. We also present an experimental evaluation in the domain of sports.

1 Introduction

Broadcast networks spend a large amount of resources in covering many events such as soccer matches or Formula 1 races, and personalizing content for many different types of audiences. In order to cover such events, editors spend lots of hours performing repetitive tasks, such as annotating and selecting video segments to create event reports. Additionally, broadcast networks also have to create many specific reports and news shows for different regions, which include local highlights amongst the general interest ones. The work presented in this paper aims at automating some of those tasks, in order to bring down the cost, and freeing editors from mechanical tasks, allowing them to focus in the more creative ones.

The GENA (Generation of Audio-visual NArrative) system was designed in collaboration with *Televisió de Catalunya* (TVC) (the public Catalan broadcast network) precisely to address this problem. Specifically, GENA was designed to generate different kinds of sport event summaries, and localized news shows, although the experiments reported in this paper only cover the sports domains. GENA generates an audio-visual narrative, which in this context means content that can be broadcast through a TV channel, IPTV, or the web.

In order to address this problem GENA uses a CBR approach. TVC kindly provided us with a collection of audio-visual narratives (soccer game summaries, Formula 1 reports, news shows) generated by professional reporters, together with the complete original assets from where the narratives were generated. Each of these narratives was captured as a case. For example, in the case of soccer summaries, a case consists of the complete original soccer game (complete audio-visual content plus metadata), a description of the type of summary desired for the game, and the actual summary. Given a new request to generate a summary, GENA retrieves similar cases of summaries generated by professional reporters, and generates a new candidate summary ready to be directly broadcast.

In this paper, we present GENA, including the knowledge representation used in order to capture audio-visual narratives, and the specific retrieval and adaptation procedures used to deal with such complex data. From an application point of view, GENA is a novel CBR solution to a real life problem. From a theoretical point of view the main contributions of GENA are a new similarity measure specially designed for narrative, which is a generalization of the Jaccard similarity [9], and a new adaptation approach based on generating solutions that exhibit similar statistical properties as the solutions in the retrieved cases.

The remainder of this paper is organized as follows. Section 2 describes in detail the specific problem that GENA tries to address. Then, Section 3 describes the GENA system, including retrieval and adaptation. Section 4 reports our experimental results in the domains of Soccer, and Formula 1. Finally, Section 5 compares GENA with existing work in the literature.

2 Audio-visual Narrative Generation

The main goal behind GENA is to generate audio-visual narratives, like sport summaries, or personalized news shows, from the existing data in the repositories of a broadcast network, while requiring minimal additional manual intervention. For that reason, in this section we will first describe the structure of the data already available in the repositories of a broadcast network, and then formulate the specific problem that GENA was designed to solve.

2.1 Available Data

Information in the repositories of Broadcast networks is stored as a series of layers, some of them containing the original audio-visual form (videos of sport

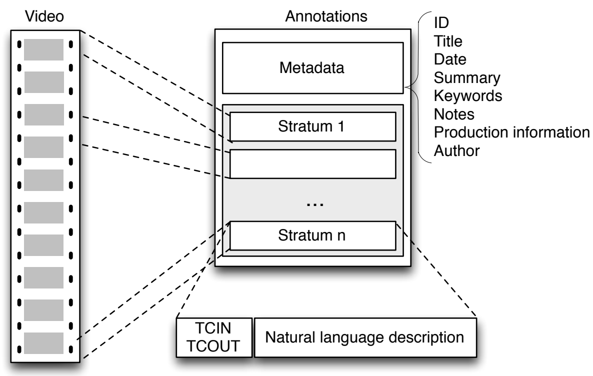

Fig. 1. Structure of the available information for sport events.

events, news clips, etc.), and some others contain annotations like keywords, natural language descriptions and other metadata in order to make the information available when reporters need to look for it. When a reporter or an editor wants to create, for example, a report on the best moments of a given sport event, they query the repositories of information using the annotations (such as keywords) to access the audio-visual information, and then they splice parts of the audio-visual content together in order to form a good summary. This is precisely the task GENA is designed to perform.

Figure 1 shows the structure of the available information from which GENA needs to generate audio-visual narratives. Specifically, we have applied GENA to two different domains: sport events (soccer and Formula 1) and news shows, but this paper only reports experiments concerning the sports domain. Figure 1 shows the structure of the information that TVC stores for sport events, as composed of two main parts: video and annotations. The current version of GENA is not equipped with any video processing reasoning capabilities (although a video processing module is being worked on). Thus, all the reasoning performed by GENA is done at the level of the annotations.

In the case of sports, Figure 1 shows that the video of a sport event is annotated with two main kinds of information: metadata and *strata* list. The metadata contain general information such as the ID, title, date and keywords of the video. The strata information is the most useful for GENA and consists of a list of individual records, called strata (commonly also known as "Time Segment Annotations"). Each of these strata contains a natural language description of a fragment of video (specified by a start time, TCIN, and an end time, TCOUT). For example, a typical stratum in the soccer domain could contain the sentence: "Leo Messi scores a goal from midfield", and correspond to some seconds of video that capture the moment in which Leo Messi scored a goal. The list of strata can be used by GENA and by the human editors to identify and search over the information contained in the video. Moreover, the list of strata might not be sorted, and strata can totally or partial overlap. For example, there might be a stratum labelled “Replay of the best moments of the first half of the game”, and each of those individual moments will be described by their respective strata.

Specifically, during the development of the GENA project, TVC provided (only for the specific purposes of this project) 18 soccer games (from the 2009 - 2010 Spanish national league) and the whole set of Formula 1 races from the 2010 championship. In the soccer domain, TVC also made available all the different summaries that had been generated from those games (long summaries, short summaries, best goals, etc.). In the Formula 1 domain, we had general summaries as well as summaries focused on specific drivers.

2.2 Problem Statement

The problem that GENA is designed to solve is the following: given the existing data from sport events and news stories described in the previous section, is it possible to create an artificial intelligence system that can generate different types of sport summaries or personalized news shows automatically?

In order to address that problem, we decided to use a case-based reasoning approach, and captured the available information in the form of cases. A case in GENA captures an example of how did a human expert generate a summary or a news show from a given original asset (from a soccer game, Formula 1 race, etc.). Therefore, a case in GENA is composed of the following parts:

- Problem Description:
  - *Original asset*: the original soccer game or Formula 1 race
  - *Generation parameters*: what kind of narrative had to be generated from the original asset (a summary, a report of a specific player/driver, a summary for a specific audience, etc.).
- Solution:
  The *Generated narrative*, represented as the subset of strata that were selected from the original asset, and the particular order in which they were sequenced.

The generation parameters depend on the domain. For example, in soccer, they consist of a “summary type” from the following list: long summary, short summary, first part summary, second part summary, goals, and summary focused on a specific player. In case the summary type is ”summary focused on a specific player”, then there is a second parameter specifying which player to focus on. Analogously, the Formula 1 domain has similar narrative types.

Moreover, notice that this case representation makes the assumption that human reporters and editors generate summaries or news shows by just selecting and sequencing strata from the original assets. In reality, humans tend to sometimes trim the strata (remove some parts of it), or extend them (add additional video footage not in the strata) in order to achieve desired effects. Since GENA cannot reason about the content in the video, this part was left out as future work. However, the strata granularity level is enough to generate interesting and useful summaries, and news shows to be broadcast directly.

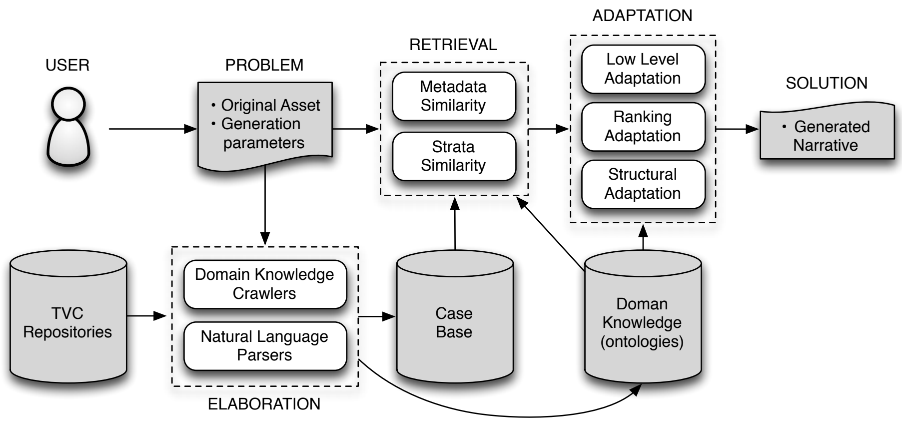

Fig. 2. High-level view of the GENA narrative generation architecture.

GENA contains a case-base with 59 soccer cases (59 different summaries generated from the 18 soccer games we had available), and 47 Formula 1 cases (47 different summaries generated from 15 races).

Problems contain only two parts: an original asset, and the generation parameters. The goal of GENA upon receiving a problem is to generate a narrative from the original asset constrained by the generation parameters. An example problem would be:

- Original asset: soccer game "Barça - Mallorca (4-2)".
- Generation parameters: "summary focused on a specific player", "Leo Messi" (meaning that GENA needs to generate a summary focusing on Leo Messi).

Upon receiving a problem, GENA would retrieve one or more cases containing similar types of summaries from the soccer domain, and then generate a narrative of the "Barça - Mallorca (4-2)" game in the same way as the narrative was generated in the retrieved case. The next section describes the retrieval and adaptation techniques we designed for the GENA system.

3 GENA

Figure 2 shows a high level overview of the GENA system architecture, with its three main components: elaboration, retrieval and adaptation. The role of the elaboration module is to 1) process the data in the broadcast network repositories (TVC repositories) and turn them into cases and additional domain knowledge (such as ontologies for the different domains), and 2) process problems coming from the user so GENA can work with them (e.g. perform any kind of natural language analysis needed). Once problems have been elaborated, the retrieval module retrieves one or several cases that are relevant to the problem at hand by combining several similarity measures. Finally, the adaptation module reuses the solution(s) in the retrieved cases to generate the desired narrative (e.g. a summary of a soccer game). We have implemented three different adaptation modules of increasing complexity, which will be compared in Section 4.

The following subsections describe each one of these three modules in detail.

3.1 Elaboration

GENA contains two main explicit knowledge containers: the case-base and the domain knowledge repository. The former is a plain list of cases, and the second one contains a collection of ontologies for the different domains GENA can deal with. In particular, in the version reported in this paper, GENA has three ontologies: a generic one, one for soccer, and one for Formula 1.

The *elaboration* module preprocesses the data coming into GENA to feed these two knowledge containers, and so that retrieval and adaptation can be performer effectively. Therefore, the elaboration module has two main goals:

- Transform all the available knowledge into a unified representation so GENA can reason about it (domain knowledge). This goal is mainly achieved by the *domain knowledge crawlers* in GENA.
- Transform all the suitable available knowledge into episodic knowledge (problems and cases). In order to achieve this goal, the most complex process is to perform natural language processing in the strata descriptions of the assets.

Given that the data available in the broadcast network repositories was through for human consumption, rather than for being used by an artificial intelligence system, there are no formal domain ontologies available for GENA to use. However, there are glossaries of terms and thesauri used by the documentarists in the network to classify all the assets, containing thousands of terms in a semi-structured way. The first domain knowledge crawler in GENA contrasts these lists of terms with the words appearing in the natural language description of the strata in the available assets. Those terms appearing with enough frequency in the assets, are terms that are likely to be useful to GENA. Thus, the crawler compiles a list of the useful terms for each of the three domains GENA was applied to. Specifically, the resulting ontologies for Soccer and Formula 1 contain 390 and 85 unique (i.e. non synonymous) concepts respectively.

In order to organize the set of terms obtained by the previous crawler, we defined a generic ontology with the basic concepts of narrative (story, discourse, existents, events, actions, happenings, characters and props) as identified by Chatman [2], and later semi-automatically classified all the terms obtained by the first crawler into the concepts of this generic ontology.

A final crawler mined information about the different teams, soccer players, Formula 1 drivers, cities, countries, politicians, etc. available in the different data repositories of TVC in order to populate the resulting ontologies.

Once the knowledge crawlers have populated all the ontologies in GENA, the next step in the elaboration process is to process the soccer games, Formula 1

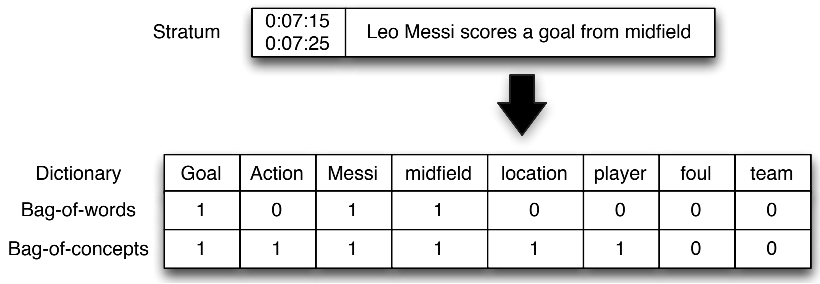

Fig. 3. Illustration of the BoW and BoC generation for the description of a stratum.

races and news shows available in order to create cases. The most complex part of this process is to turn the natural language descriptions in the strata into some computer-understandable representation that GENA can reason with. In particular, we opted for a *bag-of-concepts* (BOC) [11] representation.

Given a predefined dictionary of words, the standard *bag-of-words* (BOW) [4] representation, represents a stratum as a vector with one position per word in the dictionary. Each position is 1 or 0 depending on whether the corresponding word appears in the stratum or not. A bag-of-concepts representation extends this representation in a simple but significant way: rather than having a dictionary of words, the dictionary contains *concepts* organized in an *is-a* hierarchy. If a given concept c appears in a stratum, then, not only c, but all the super-concepts of c are also added to the bag-of-concepts. This is illustrated in Figure 3, where the bag-of-words and bag-of-concepts for a stratum with the description “Leo Messi scores a goal from midfield” using a small dictionary are shown.

We used the FreeLing [7] natural language parser to analyze the natural language text from each stratum. We then cross the result of the analysis with the available ontologies in order to determine which of the concepts in the ontologies of GENA is present in each stratum, and build the bag-of-concepts of each stratum. Once the BOC of each stratum in the cases is generated, they are added to the case-base of GENA.

Each time a new problem arrives, the same natural language analysis is performed, before GENA attempts to solve the problem.

3.2 Retrieval

Retrieval in GENA works as a 2 step process: In the first *filtering* process, a subset of candidate cases of the case-base is selected as those cases that belong to the appropriate domain (soccer, Formula 1 or news) are annotated with the same generation parameters as the problem (e.g. same kind of summary). Then, in a second step, GENA follows a standard k-nearest neighbor algorithm using a specially designed similarity measure to retrieve the k most similar cases to the problem at hand. This section focuses on the similarity measure used for the second step of the retrieval process.

The most basic similarity measure used in GENA is similarity between two strata, for which we experimented with several options. In the experiments reported in this paper, we defined the similarity between two strata, a and b as the *cosine* similarity between their two bag-of-concepts, $BOC(a)$, and $BOC(b)$:

$$S_{cos}(a,b) = cos(\theta) = \frac{BOC(a) \cdot BOC(b)}{|BOC(a)||BOC(b)|}$$

For example, consider the two strata: $a =$ "Leo Messi scores a goal from midfield" and $b =$ "Messi scores a goal". We compute their bag-of-concepts using the dictionary shown in Figure 3, and we obtain $BOC(a) = (1,1,1,1,1,1,0,0)$, $BOC(b) = (1,1,1,0,0,1,0,0)$. Their cosine similarity is:

$$S_{cos}(a,b) = \frac{(1,1,1,1,1,1,0,0) \cdot (1,1,1,0,0,1,0,0)}{|(1,1,1,1,1,1,0,0)||(1,1,1,0,0,1,0,0)|} = \frac{4}{\sqrt{6}\sqrt{4}} = 0.816$$

GENA can use the strata similarity to assess similarity between complete narratives (e.g. between soccer matches, F1 races, etc.). Since the problem of comparing complete narratives is very complex, GENA uses two simplification assumptions in order to make case retrieval feasible: a) narratives can be represented as sets of strata, b) an approximation of the similarity measure will be computed, rather than the exact measure.

The first assumption is that narratives can be represented as sets of strata, i.e. GENA ignores the order in which the strata are sequenced in a narrative for the purposes of case retrieval (since order is important in the formation of narratives, the order is indeed taken into account during adaptation, but not during retrieval). Similarity between sets of elements can be approximated using the Jaccard similarity [9], which estimates the similarity between two sets A and B as follows:

$$S_{Jaccard}(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

The Jaccard similarity returns 1 when the two sets are identical, and 0 when they are disjoint. In general, the larger the intersection, the higher the similarity. Moreover, in the case of GENA, each strata is practically unique[4], and thus, the intersection between the sets of strata of two narratives is likely to be empty. For that reason, we defined a generalization of the Jaccard similarity, as follows (assuming $|A| < |B|$):

$$S_{GJ}(A,B) = \max_{m \in M} \frac{\sum_{a \in A} S_{cos}(a, m(a))}{|A| + |B| - \sum_{a \in A} S_{cos}(a, m(a))}$$

Intuitively, this new measure works as follows. Assume m is an injective mapping from the elements in A to the elements in B, so that if $a \in A$ then

[4] Journalists annotate events in natural language plus some common tags, it is very unlikely that two strata have exactly the same tags and natural language annotation.

$m(a) \in B$. The numerator is the sum of similarities of the elements in A with their corresponding elements in B (which is an approximation of their intersection), and the denominator is just the sum of elements from A and B minus the approximation of their intersection. In this way, the numerator is bounded between 0 and $|A|$, and the denominator between $|B|$ and $|A| + |B|$, since we assumed that $|A| \leq |B|$, the similarity is always in the interval $[0, 1]$. Moreover, the more similar the strata in A to their corresponding strata in B, the higher the similarity. The final step is to select the mapping m that maximizes this similarity from the set M of all possible injective mappings from A to B.

It is easy to prove that if the strata similarity metric used is the identity function, the previous measure corresponds exactly to the Jaccard similarity, and thus, the proposed measure is a generalization of it. Moreover, since computing all the possible mappings is an expensive operation, we will only approximate it by the following measure:

$$S_{SGJ}(A, B) = \frac{\sum_{a \in A} \max_{b \in B} S_{cos}(a, b)}{|A| + |B| - \sum_{a \in A} \max_{b \in B} S_{cos}(a, b)}$$

This measure has a polynomial time complexity and is the one used for case retrieval in GENA.

Finally, using the previously defined narrative similarity measure, GENA defines the similarity between a problem and a case as follows. A problem defines an original asset from which we want to generate a narrative, and a case contains both an asset and a generated narrative. Thus, there are two similarities that can be assessed:

1. The similarity between the asset in the problem and the original asset in the case: which gives as a measure of how similar was the problem solved in the case to the problem at hand.
2. The similarity between the asset in the problem and the generated narrative in the case: which gives us a measure of how similar are the strata that were selected to form the target narrative in the case to the ones in the problem at hand. In other words, this gives us a measure of how easy would it be to generate a target narrative similar to the one in the case using the strata in the problem at hand, i.e. this is a measure of adaptability.

Following ideas from *adaptation-guided retrieval* [12], GENA combines the previous two similarities to obtain a final score for each case. Thus, given a problem with original asset A, and a case with an original asset C and generated narrative G, the score the GENA assigns to each case is defined as:

$$S(A, (C, G) = \alpha \times S_{SGJ}(A, C) + (1 - \alpha) \times S_{SGJ}(A, G)$$

where α is a constant that can take values in the interval $[0, 1]$.

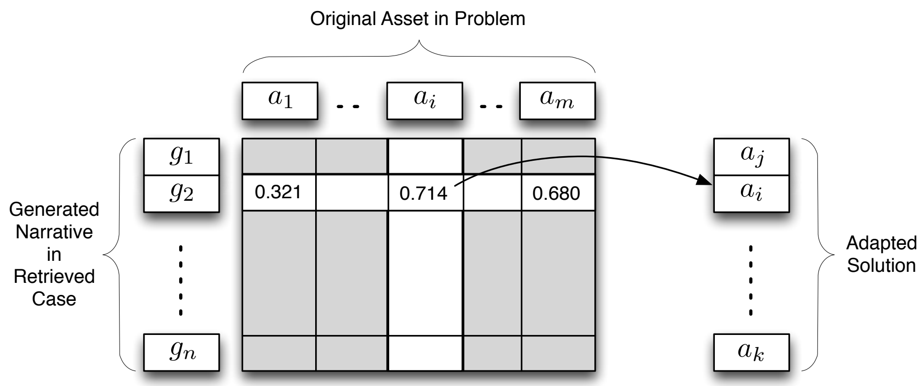

Fig. 4. Illustration of the *low-level-adaptation* module, where the numbers in the matrix represent the similarities between the strata in the generated narrative in the retrieved case and the strata in the original asset in the problem.

3.3 Adaptation

GENA implements three different adaptation modules, of increasing complexity: the *low-level adaptation* (LLA) module, the *ranking adaptation* (RA) module, and the *structural adaptation* (SA) module. Let us describe each one of them in detail.

The low-level adaptation module, or LLA, adapts the solution from a single retrieved case in the following way. The solution in a case is an ordered list of strata: $[s_1, ..., s_n]$. Now, let us call $m(s_i) = a$ to the most similar stratum a in the problem at hand to the strata s_i. The solution generated by the LLA module is: $[m(s_1), ..., m(s_n)]$. In other words, the LLA generates a solution for the problem at hand, by taking the solution in the retrieved case and replacing each strata by the most similar one in the problem at hand. This idea is illustrated in Figure 4, where it can be see that the LLA just needs to compute a similarity matrix with the similarities between all the strata in the solution in the retrieved case and all the strata in the problem at hand.

The ranking adaptation module, or RA, performs a similar process to the LLA, but considering a set of k retrieved cases, rather than a single retrieved case. Given a set of cases $C_1, ..., C_k$, the RA proceeds as follows:

1. Let $S_1, ..., S_k$ be the solutions that the LLA would generate from each of the cases in $C_1, ..., C_k$ respectively.
2. Let the score of a stratum $score(a) = |\{S \in S_1, ..., S_k | a \in S\}|$ be the number of solutions from $S_1, ..., S_k$ in which a given strata a from the problem at hand appears.
3. The RA computes N as the average size of $S_1, ..., S_k$ (in number of strata), and selects the N strata from the problem at hand that have highest score.
4. The set of N strata selected in the previous step are ordered according to their average positions in $S_1, ..., S_k$ in order to form the final solution S.

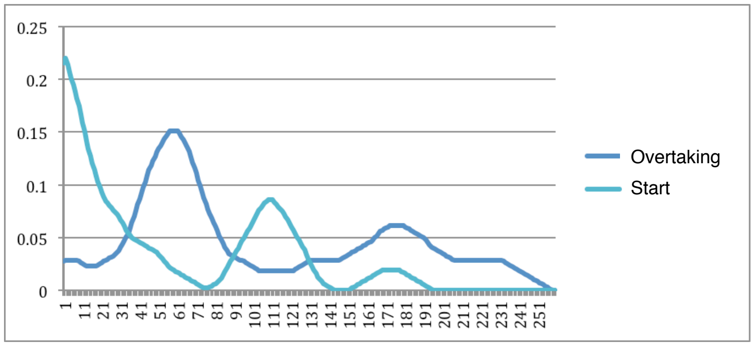

Fig. 5. Illustration of the the temporal distribution statistical indicator computed by the TDC module for the two concepts *overtaking* and *start* in the Formula 1 domain. The vertical axis represents probability of a stratum with the concept to appear, and the horizontal axis represents time.

The advantage of the RA module is that it is less brittle than the LLA to exceptional events, or to problems that require parts from more than one case. If the problem has some exceptional event, it is unlikely that the retrieved case contains it, but more likely that at least one case in the ranking has it.

Finally, the structural adaptation module, or SA, is the most complex of the adaptation modules in GENA and consists of three main processes (for the sake of space, we only provide a high-level view of this module):

1. A *Term Relevance* module assesses a relevance factor (a real number in the interval $[0, 1]$) for each concept in the ontology, by measuring how important is the fact that a given concept appears in a stratum for the strata to be selected as part of the solution of the cases with the same generation parameters as the problem at hand. For example, soccer concepts such as *goal* have a high relevance, whereas terms, like *midfield* are not very relevant.
2. A *Target Distribution Computation* (TDC) module, computes, for each concept in the ontology, a collection of statistical indicators that all cases with the given generation parameters satisfy. In particular, for each concept, it computes three indicators: percentage of all the strata with the concept that appear in the solution, percentage of all the strata not belonging to a replay and with the concept that appear in the solution, and temporal distribution of strata in the solution with a given concept. This last indicator is illustrated in Figure 5, where we can see the distribution of strata containing two given concepts in summaries of Formula 1 races. In this example, we can see that the indicator represents that in Formula 1 summaries, it is more common to have the strata that talk about the start of the race at the beginning, and that immediately after we should have strata that talk about overtaking moves.
3. Finally, the SA module generates a solution as follows: it generates an initial solution by using LLA or RA. Then, using a hill-climbing search process, it

modifies this solution by adding/removing/reordering strata trying to maximize the fit of the solution to the statistical indicators computed by the TDC module. When the solution cannot be transformed in any further way to improve the fit to the statistical indicators, it is returned to the user.

The advantages of the structural adaptation approach over the low-level or ranking adaptation approaches can be seen with this simple example. Imagine that a user asks GENA to generate the summary of a soccer game that had 5 goals. Imagine GENA retrieves a case that contained only 3 goals. Since goals in summaries are only shown once, the solution generated by the low-level adaptation will only contain three strata with goals, and will not show all 5 goals in the problem at hand. However, the structural adaptation module would realize (through the statistical indicators) that all the strata marked as goal in the original asset were selected for the solution. Therefore, it will include all 5 goals from the problem at hand in the final solution.

4 Experimental Evaluation

In order to evaluate the performance of GENA we performed a *leave-one-out* evaluation in two domains: Soccer and Formula 1. The case-bases used for the two domains had 59 and 47 cases respectively. Different cases contain different types of generation parameters (whole game summaries, summaries of the first part, reports of the performance of a specific player or driver, etc.). For each case in the leave-one-out evaluation, we asked GENA to generate a narrative and compared it against the narrative contained in the case, that had been authored by a professional editor or journalist.

In order to compare the output of GENA with the human-generated narrative, we used the following metrics:

- The Jaccard similarity, as described in Section 3.2, which measures the proportion of strata that GENA selected that were also selected by the human professional.
- Since the Jaccard similarity penalizes severely the fact that GENA might not select the exact strata the expert did, but one that is very similar, we also used the proposed generalization of the Jaccard similarity S_{SGJ}, as described in Section 3.2, which measures how similar are the strata selected by GENA to those selected by the human professional.
- A measure of whether GENA sequences the strata in the same order as the human does, defined as:

$$O(S, S') = \frac{\sum_{a \in C} \sum_{b \in (C \setminus a)} o(a, b, S, S')}{|C|(|C| - 1)}$$

  where $C = S \cap S'$, and $o(a, b, S, S') = 1$ when a and b appear in the same order in S and S' (i.e. if a appears before b in S, they also must appear in that order in S', regardless of any other strata that can be in between them), and 0 otherwise.

Table 1. Experimental results in the soccer domain. All measures are normalized between 0 and 1, and higher is better.

	$\alpha = 1$			$\alpha = 0$		
Adaptation	$S_{Jaccard}$	S_{SGJ}	O	$S_{Jaccard}$	S_{SGJ}	O
Low-Level Adaptation	0.105	0.860	0.480	0.067	0.846	0.382
Ranking Adaptation	0.157	0.935	0.467	0.155	0.932	0.455
Structural Adaptation	0.147	0.906	0.550	0.143	0.896	0.541

Table 2. Experimental results in the Formula 1 domain. All measures are normalized between 0 and 1, and higher is better.

	$\alpha = 1$			$\alpha = 0$		
Adaptation	$S_{Jaccard}$	S_{SGJ}	O	$S_{Jaccard}$	S_{SGJ}	O
Low-Level Adaptation	0.107	0.883	0.233	0.079	0.884	0.175
Ranking Adaptation	0.197	0.947	0.404	0.198	0.949	0.355
Structural Adaptation	0.145	0.913	0.420	0.140	0.899	0.343

Moreover, we report results with $\alpha = 1$ (the retrieval module only considers the similarity between the problem at hand and the original asset in the case) and with $\alpha = 0$ (the retrieval module only considers the similarity between the problem at hand and the generated narrative in the case), experiments with intermediate values of the α parameter are part of our future work.

Tables 1 and 2 show the results for Soccer and Formula 1 respectively. Considering soccer, the first thing we observe is that the Jaccard similarity severely penalizes GENA for not selecting exactly the same strata that the expert selected. However, as S_{SGJ} shows, the strata selected by GENA are very similar to the ones selected by the human expert. For example, with $\alpha = 1$, the average similarity between the strata of the output of GENA and those of the narrative by the human expert is 0.935 using the ranking adaptation method. Thus, we can conclude that GENA selects strata that are very similar to those selected by a human expert. Considering the order in which GENA sequences the strata, we see that the O measure varies from 0.382 using low-level adaptation and $\alpha = 0$ and up to 0.550 with structural adaptation and $\alpha = 1$. This means that about half of the strata are ordered in the same way as those generated by the human expert. We can also see that the choice of the adaptation procedure has a very strong impact in all measures, with both ranking adaptation and structural adaptation obtaining better results than the low-level adaptation measure. Concerning Formula 1, Table 2 shows similar trends as for soccer.

In summary, GENA can generate narratives that contain strata that are very similar to those generated by human experts. The order of the strata is highly improved by the complex adaptation procedures used in GENA, but improving it even further is part of our future work. Additionally, an evaluation of the output of GENA by human experts, manually inspecting each answer produced by GENA is also ongoing.

5 Related Work

Three areas of work are relevant for the work presented in this paper: narrative generation, textual CBR and AI applications to sports and news domains.

CBR approaches to narrative generation date back to the Minstrel system [13]. Minstrel generated King Arthur styled narratives by retrieving and adapting past story snippets. A key difference between Minstrel and GENA is that Minstrel's goal was to generate original and creative narratives, whereas in GENA, the goal is to generate summaries, or personalized news. Therefore, the content in narratives generated by GENA is selected from an original asset, while in Minstrel it is generated by adapting the stories form the retrieved cases. Similar to Minstrel, other more recent work like Mexica [8] and Riu [6] also use CBR to generate narratives, but none of them focus on generating narratives by selecting and sequencing content from an original asset, as GENA does.

GENA's case retrieval mechanism is related to work on textual CBR [3], where typically, the goal is to retrieve textual documents from a case-base that are relevant or similar to a given query document. This has been explored in depth in the CBR community; representative examples are the CR2N [1] system for identification of reusable pieces of text, and the work on the jCOLIBRI system [10] for generic textual case retrieval.

Finally, there is a recent interest in artificial intelligence applications to generate sport event reports or personalized news. For example, the *News at Seven* [5] system generates personalized news given a set of user preferences. The main difference between GENA and News at Seven, is that GENA generates content exclusively based on the input audio-visual asset in the problem at hand, while News at Seven starts with a set of preferences and crawls the Internet for relevant material. For example, News at Seven would not be capable of performing GENA's task of, given a Formula 1 race, generate a summary of the race with the best moments sequenced in the appropriate way.

6 Conclusions and Future Work

This paper has presented GENA, a case-based reasoning system capable of generating audio-visual narratives by drawing from previously annotated content generated by human experts. We described a new similarity measure for narratives composed of lists of strata and a collection of adaptation procedures to adapt narratives. GENA was evaluated in two different domains: soccer and Formula 1, demonstrating the generality of the approach.

Our experimental results show that GENA succeeds in generating narratives that are similar to those generated by human experts in that they contain strata that are very similar. However, there is still room for improvement in the order in which these strata are sequenced by GENA. For example, GENA currently doesn't fully exploit semantic relations between entities and actions.

As part of our future work, we are currently applying GENA to the news shows domain, where the main challenge is a very large vocabulary and the requirement of better natural language processing. Additionally, we are also exploring new forms of adaptation that help GENA in generating narratives that resemble even more closely those generated by humans. As part of our ongoing work, we plan to give the output of GENA to human experts in order to have their subjective impression on the quality of the generated narratives.

Acknowledgements. This research was supported by the project CENIT-BUSCAMEDIA 2009 ref. CEN20091026.

References

[1] Adeyanju, I., Wiratunga, N., Lothian, R., Sripada, S., Lamontagne, L.: Case retrieval reuse net (cr2n): An architecture for reuse of textual solutions. In: ICCBR. pp. 14–28 (2009)

[2] Chatman, S.: Story and Discourse: Narrative Structure in Fiction and Film. Cornell University Press (June 1978)

[3] Lenz, M., Hübner, A., Kunze, M.: Textual cbr. In: Lenz, M., Burkhard, H.D., Bartsch-Spörl, B., Wess, S. (eds.) Case-Based Reasoning Technology, Lecture Notes in Computer Science, vol. 1400, pp. 115–137. Springer Berlin / Heidelberg (1998)

[4] Lewis, D.D.: Naive (bayes) at forty: The independence assumption in information retrieval. In: Proceedings of the 10th European Conference on Machine Learning. pp. 4–15. ECML '98, Springer-Verlag, London, UK, UK (1998)

[5] Nichols, N., Hammond, K.: Machine-generated multimedia content. In: Proceedings of the 2009 Second International Conferences on Advances in Computer-Human Interactions. pp. 336–341. ACHI '09, IEEE Computer Society, Washington, DC, USA (2009)

[6] Ontañón, S., Zhu, J.: Story and Text Generation through Computational Analogy in the Riu System. In: AIIDE. pp. 51–56. The AAAI Press (2010)

[7] Padró, L., Collado, M., Reese, S., Lloberes, M., Castelln, I.: Freeling 2.1: Five years of open-source language processing tools. In: Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (may 2010)

[8] Pérez y Pérez, R., Sharples, M.: Mexica: A computer model of a cognitive account of creative writing. Journal of Experimental and Theoretical Artificial Intelligence 13(2), 119–139 (2001)

[9] Real, R., Vargas, J.M.: The probabilistic basis of jaccards index of similarity. Systematic Biology 45(3), 380–385 (1996)

[10] Recio, J.A., Díaz-agudo, B., Gómez-martín, M.A., Wiratunga, N.: Extending jcolibri for textual cbr. In: In Procs. of 6th International Conference on CBR, ICCBR 2005, volume 3620 of LNCS. pp. 421–435. SpringerVerlang (2005)

[11] Sahlgren, M., Cöster, R.: Using bag-of-concepts to improve the performance of support vector machines in text categorization. In: Proceedings of the 20th international conference on Computational Linguistics. COLING '04, Association for Computational Linguistics, Stroudsburg, PA, USA (2004)

[12] Smyth, B., Keane, M.T.: Adaptation-guided retrieval: questioning the similarity assumption in reasoning. Artificial intelligence 102, 249–293 (1998)

[13] Turner, S.R.: Minstrel: a computer model of creativity and storytelling. Ph.D. thesis, University of California at Los Angeles, Los Angeles, CA, USA (1993)