웹에서 축출된 정보를 이용한 축구 경기의 시맨틱 인덱싱

Issao Hirata[O],김명훈,설상훈
고려대학교 전자전기공학과
{issao,mhkim, sull}@mpeg.korea.ac.kr

Semantic Indexing for Soccer Videos Using Web-Extracted Information

Issao Hirata[O], Myeonghoon Kim,Sanghoon Sull
School of Electrical Engineering, Korea University

Abstract

The rapid growing of video content production leads to the necessity of developing more complex indexing systems in order to efficiently allow searching, retrieval and presentation of the desired segments of videos. This paper presents a method for indexing soccer video through automatic extraction of information from internet. The proposed paper defines a metadata structure to formally represent the knowledge of soccer matches and provides an automatic method to extract semantic information from web-sites. This approach improves the capability to extract more reliable and richer semantic information for soccer videos. Experimental results demonstrate that the proposed method provides an efficient performance.

1. Introduction

The recent development of the digital multimedia industry leaded to an explosion in the production of multimedia content, in special video files. One study performed by Berkeley University [1] estimated that only in the year 2003, almost 70,000 terabytes of television content were created. Due to this huge amount of recorded data, it is essential to develop efficient indexing systems which should allow searching, retrievals and presentations of only the necessary video segments.

The first researches in this field were related to the extraction of low-level visual features, such as detection of human-faces [2], color [3], camera motion [4] and texts on frames [5]. However, the usage of only these techniques is not good enough to obtain high-level description for the video segments, which brings the necessity of using semantics information in order to index the video content.

On the past recent years, it is observed a great effort in order to define a formal model to represent the semantics metadata. Basically, it is possible to divide these efforts in two different perspectives:

• from Knowledge Engineering and Artificial Intelligence domain, we obtain the concept of Ontology which has some formal representations, like the Resource Description Framework (RDF) [6] and Web Ontology Language (OWL) [7], widely used on Web-Semantics [8] applications.

• from the video indexing perspective, there are defined some standard representations to describe the media content, such as MPEG-7[9] and TV-Anytime [10].

More recently, it has been developed some works which shows that both approaches can be interoperable [11] and possibly integrated [12]. However, until now there is no work attempting to use techniques from Web-Semantics in order to extract information needed to index a video sequence.

This is exactly the purpose of this paper. We applied text extraction techniques from Web-Semantics in order to extract information from web-sites and used it to semantically index video sequences. We have chosen soccer matches as the domain of our work. Many previous works tried to extract semantic information from soccer video files, but most of them analyzed only visual features, which showed to be not an efficient approach to obtain high-level information.

Thus, in this paper we propose a method to index a soccer video sequence using semantic information automatically extracted from the Internet. To perform the task, it were defined: an ontology that represents the semantic knowledge necessary to index soccer videos; a method to automatically extract soccer match information (such as teams, score, players, goals and other events) from texts containing the description of the matches which are available on the internet; a method that organizes the extracted information storing it according to the defined ontology, turning it possible to be accessed by video indexing applications.

This paper is organized as follows: In section 2, it is presented the metadata structure defined to represent the semantic information. The extraction method used to obtain semantic information from the web-sites is described in section 3. Section 4 is

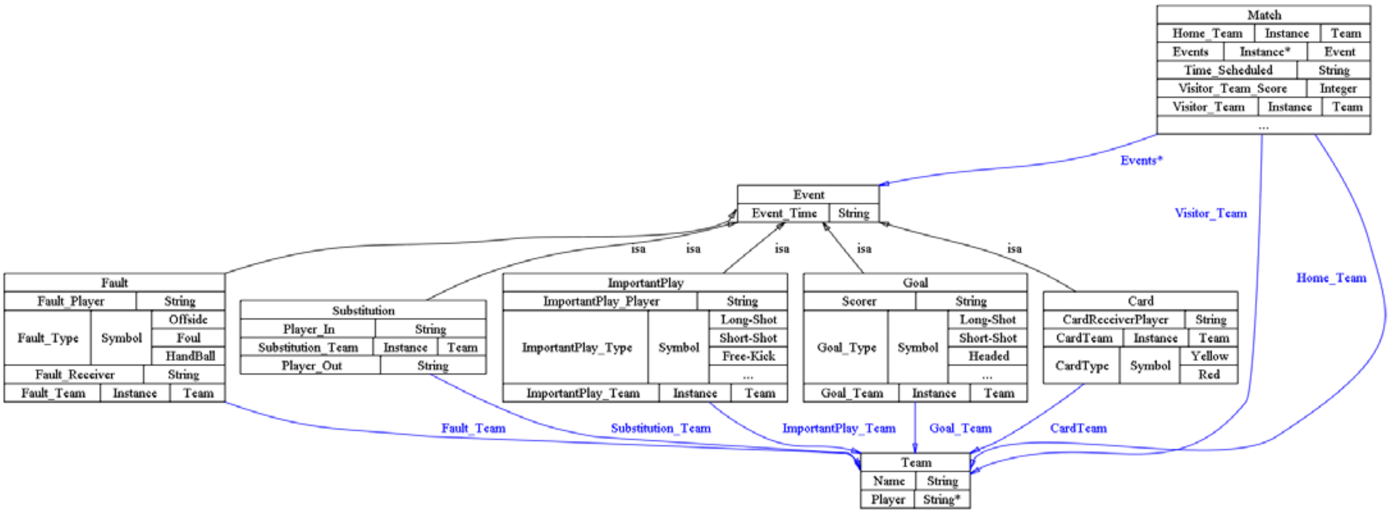

Fig. 1. Defined Ontology

related to the experiments realized in order to evaluate the performance of the method. Finally in section 5, some finals comments are made and some future work topics are suggested.

2. Metadata Structure Definition

In order to extract the semantic information, it is necessary first to define a knowledge structure in which the information of the domain will be stored. For that, we defined an ontology for soccer matches.

Searching on the scientific literature, it is possible to find some previous work which have already defined some ontologies related to the soccer domain [13][14][15]. However, most of these models or are too complex or they focus on different issues rather than video indexing. Thus, it was defined a new ontology model, which matches with our requirements.

This model, defined the classes and its attributes for a soccer match and it is shown on Fig.1. The class ***Match*** is the root class of the ontology and stores the information about the match itself, having attributes such as *home* and *visitor team* and *match score*. Also, the ***Match*** is formed by a set of instances of events. An ***Event*** can be a ***Fault***, ***Substitution***, ***Important Play***, ***Goal*** or a ***Card***. For each event, there is an attribute that describes the *time* the event happened. The other attributes of an ***Event*** depends on which kind of event it is. For instance, a ***Goal*** event has attributes like, *Goal Team* (team who scored the goal), *Scorer Player* and *Goal Type* (long-shot, header, penalty kick, etc). The other classes and attributes of the ontology are represented by the diagram of Fig. 1.

3. Text Extraction

In this section, it is presented the method used in order to extract the semantic information from the web-sites. First, it is shown the layout and format of the pages used for extraction. Second, it is presented the tools used for the extraction and how to use these tools in order to define extraction rules for detecting the desired patterns. Finally, it will be described the overall algorithm used to detect and extract the desired soccer information from HTML files.

3.1 WebPages used for Extraction

On the web, there are many sites describing in the events that happen on a soccer match. In this paper, we preferred to use on our experiments the “Live Text” pages from BBC SPORT website. Besides being a trustful source and providing detailed information about the matches, the portal was chosen also because its pages preserves a regular layout and text patterns, turning the information extraction more efficient.

The pages obey the same layout, which is shown in Fig. 2. On the top of the page it is specified the match title, containing the home and visitor teams and the score. On the right side of the page, there is information about the teams and players’ names. The events are described in small paragraphs and they are chronologically ordered on the body of the page.

3.2 Used Framework

In order to perform the text extraction from the web-sites, we used one Language Engineering tool called GATE (General Architecture for Text Engineering) [16]. GATE is an infrastructure for developing and deploying software components that process human language. GATE is formed by a set of different plug-ins and libraries. The most important one used in this project is called ANNIE (a Nearly-New Information Extraction System) [17]. ANNIE allows the definition of extraction rules which define the patterns used to extract the information from the text.

Fig. 2. Example of Web Site used for Extraction

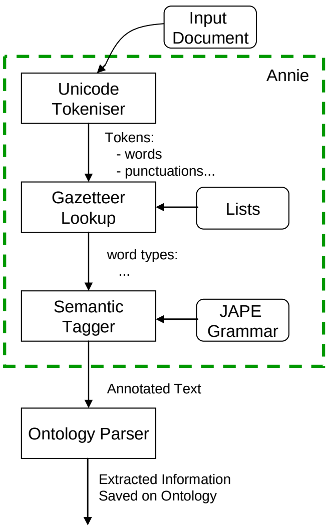

Fig. 3. Annie/GATE Pipeline for information extraction

To perform this task, ANNIE is formed by a set of components forming a pipeline. A simplified diagram of the most important components of ANNIE is shown on Fig. 3. The first component is called Tokenizer and its function is to read the input text (formed by a chain of characters) and splits in components called Tokens. A token can be a word, a number, a punctuation symbol or a space token. The second component (Gazetter Lookup) is the first component to give semantic meaning to the text. For that, the Gazetter will compare the obtained Tokens with lists of possible words for a class. For example, we defined a list containing the most famous soccer teams, like AC Milan, Arsenal, etc. Once one of these words are detected on the text (e.g. AC Milan), the Gazetter associates it to a class type (Soccer

```
Rule: MatchScore

Priority: 104

(
 {HomeTeam}
 {SpaceToken}
 ({Token.kind == number}):hometeamscore
 {Token.string == "-" }
 ({Token.kind == number}):visitorteamscore
 {SpaceToken}
 {VisitorTeam}

)

-->

:hometeamscore.HomeTeamScore = {kind = "HomeTeamScore",
                                rule = "MatchScore"},
:visitorteamscore.VisitorTeamScore = {kind = "VisitorTeamScore",
                                rule = "MatchScore"}
```

Fig. 4. Example of JAPE rule, extraction of the Match Score

Team). The third component is the core of ANNIE and it is where the most part of the information are extracted. For that, it is necessary to define grammar rules to detect patterns and associates it to an annotation class. ANNIE uses a language called JAPE described on section 2.3 to define these pattern/actions rules. The output of the Semantic Tagger is a text annotated with the extracted semantic information. Then, this annotated text is used by another GATE functionality called Ontology Parser which will basically store the extracted information on the ontology and save it on a file[18].

3.3 Definition of Grammar Rules

JAPE (Java Annotation Patterns Engine) is the language used to define grammar rules used to search for patterns on the text and to associate it to a semantic annotation. A JAPE grammar consists on a set of phases, each of them consists on a set of pattern/action rules. These rules are expressed through Regular Expressions and have 2 sides, the left-hand-side (LHS) which will define the patterns that should be detected, and the right-hand-side (RHS) which defines the semantic association to be performed when the pattern is matched. An example of define rule is shown on Fig. 4. These rules aim to extract the Score of the match.

When the Home Team appears in the text, followed by a number (with a space separation), the character "-"(separated by spaces), another number and by the Visitor Team, then this 2 numbers refers to the score of the match. It is possible to verify that this rule depends on the annotations {HomeTeam} and {VisitorTeam} which should have been already defined on another rule from a previous phase. The defined set of phases used on this paper is defined on next sub-section.

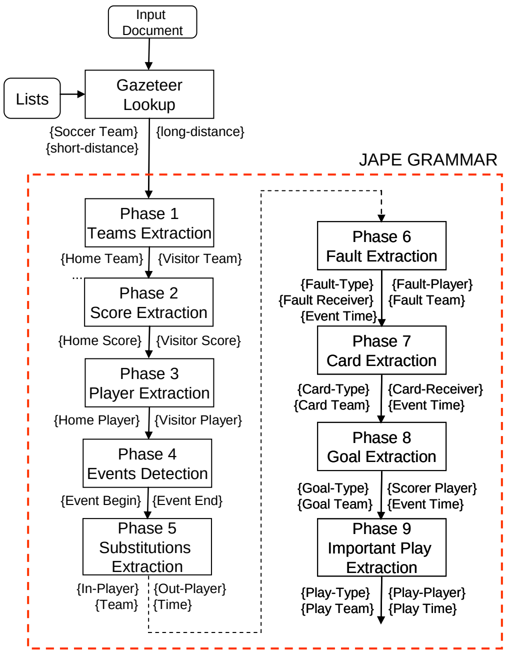

Fig. 5. Defined JAPE Grammar

3.4 Overall extraction steps

As mentioned previously, the grammar is formed by a set of phases, and the rules of a determined phase may depend on annotations obtained on previous phases. The defined JAPE grammar is divided in 9 phases, sequentially ordered as shown in the figure. Each phase aims to extract a different kind of semantic information. The output annotations produced in each of these phases are also described by the diagram of Fig. 5.

Before the JAPE grammar itself, some semantic information is obtained by the Gazetter Lookup. In this paper, the Gazetter uses a list of the known soccer teams and others auxiliary lists for temporary annotations, such as distance measurement. For example, the sub-string "20-feet" appears often in the text and the Gazetter associates this string to a {long distance} annotation.

After the Gazetter, the JAPE Grammar is executed in order to extract the desired information from the text. The firsts two phases perform the extraction of basic information of the Match, such as the team names and match score. On phase 3, it is defined a pattern in order to detect the name of the players in each team. This is relatively an easy task, since the squad of each team is available on the right side of the page (Fig. 1), and the team names were obtained on the previous phases.

After obtaining the team names, match score and players of each team, the events of the match are detected. In order to perform that, we use an auxiliary phase that will define the beginning and end position of an event in the text. Again, on Fig. 1 it is possible to see that the event description always starts with the time the event happened, and ends with a point, followed by a paragraph and the beginning of other event.

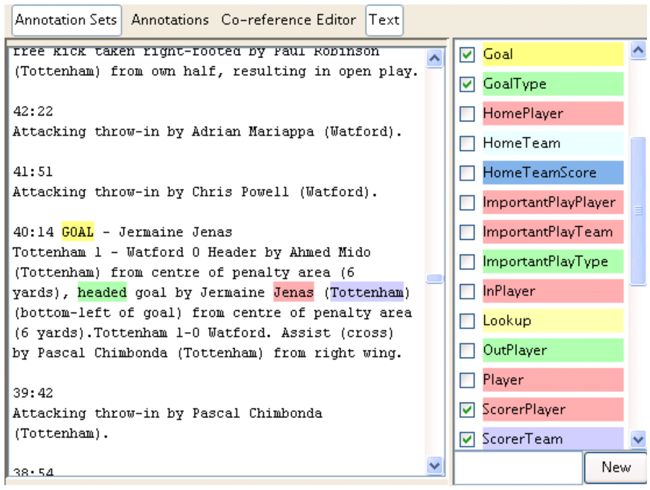

Fig. 6. GATE interface highlighting a Goal Event

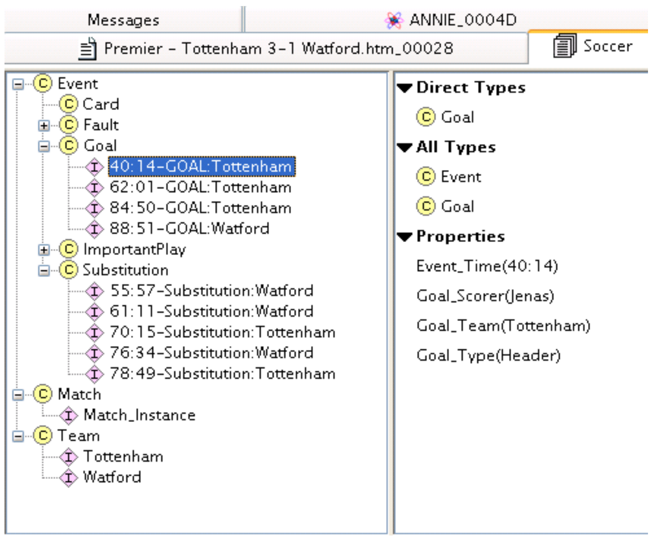

Fig. 7. GATE Interface to visualize the Extracted Information on the Ontology Structure

Once it is defined the beginning and end position of each event, the description information of the events are analyzed in order to obtain the semantic information about it. This analysis is performed by phase 5,6,7,8 and 9 which are respectively related to Substitution, Faults, Cards, Goals and Important Plays events.

After executing all the phases, the output will be an annotated text containing the necessary semantic information about the match and its events. To illustrate these annotations, figure 6 shows the GATE interface highlighting the extracted annotations for a Goal event.

Finally, after obtaining the annotations, we used the GATE Ontotext module in order to develop a method to organize this information and to store it on the defined ontology, which was presented in section 2.

The Fig. 7 shows how this information is organized in the ontology and how the ontology information can be viewed through a GATE interface. Note that, the highlighted annotations of Fig. 6 are stored as attributes of a Goal event on the ontology shown on Fig. 7. Finally the ontology and the extracted information can be saved on a XML file used the standard OWL notation, allowing the information to be accessed by other applications.

4. Experiments

In order to prove the efficiency of the proposed method, the system was tested on 10 soccer matches (from BBC Sports) from different tournaments like: Premier League, FA and UEFA Cup. The proposed method showed an efficient performance, from 1163 events, it recognized 1088 and mismatched only 5 of them, which represents an average ***recall*** of ***93.6%*** and ***precision*** of ***99.5%***. Also, it is possible to verify that for the most important events, such as the score, goals, cards and substitution, the performance was even superior, sometimes achieving 100% of both recall and precision. The results of the experiment are summarized on the table 1.

Table 1. Method performance for different events

	Correct Matches	Missed Matches	Wrong Matches	Recall	Precision
Match Information					
Score	10	0	0	**100.00%**	**100.00%**
Teams	20	0	0	**100.00%**	**100.00%**
Players	266	0	0	**100.00%**	**100.00%**
Events					
Goals	23	0	0	**100.00%**	**100.00%**
Substituitions	45	1	0	**97.83%**	**100.00%**
Cards	30	1	0	**96.77%**	**100.00%**
Fault	291	21	0	**93.27%**	**100.00%**
Important Plays	699	52	5	**93.08%**	**99.28%**

5. Conclusion and future work

Although many previous works aimed to detect events in soccer matches, by simply analyzing some video features it is not possible to extract high level semantic information such as event kind, teams, players involved, etc. However, this work showed that it is viable and efficient to extract this information from different sources, such as web-pages available on the Internet. In the case of sports events, the synchronization between the extracted information and video sequence is relatively easy, since timing information of the events are available. However, if techniques to synchronize the extracted information with the video sequence were developed, it would be possible to extend the same concept (using web-extracted information on video indexing) in other domains, such as movies, reality-shows, soap-operas, etc

6. References

[1] Lyman P, Varian H.R., "How Much Information", 2003. Retrieved from http://www.sims.berkeley.edu/how-much-info-2003. Accessed on 07/01/2007.

[2] Smith, M., and Kanade, T., "Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques," CVPR, 1997.

[3] Nagasaka, A., and Tanaka, Y., "Automatic video indexing and full-video search for object appearances," IPSJ, Vol.33, No.4, pp.543-550, 1992.

[4] Akutsu, A., Tonomura, Y., Hashimoto, H., and Ohba, Y., "Video indexing using motion vectors," In SPIE Proc. VCIP '92, pp.522-530, 1992.

[5] Nakamura, Y., and Kanade, T., "Semantic analysis for video contents extraction - spotting by association in news video," ACM Multimedia, pp.393-401, 1997.

[6] W3C Working Group. W3c resource description framework (rdf). Available at http://www.w3.org/RDF, February 2004.

[7] W3C Working Group. OWL Web Ontology Language Overview. Available at http://www.w3.org/TR/owl-features, February 2004.

[8] Berners-Lee T, Handler J, and Lassila O, The Semantic Web. Available at http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html, May 2001.

[9] MPEG, Introduction to MPEG-7. ISO / IEC JTC1 / SC29 / WG11 / N3751, MPEG, La Baule, October 2000.

[10] TV-Anytime Forum, Available at http:www.tv-anytime.org. Accessed on 07/04/2007.

[11] Tsinaraki C, Polydoros P, and Christodoulakis S, "Interoperability Support between MPEG-7/21 and OWL in DS-MIRF", IEEE Trans. Knowl. Data Eng. 19(2): 219-232, 2007

[12] Garcia R, Celma O, "Semantic integration and retrieval of multimedia meta-data", In: Proceedings of the Fifth International Workshop on Knowledge Markup and Semantic Annotation, Galway, Ireland, 2005.

[13] Bertini M, Cucchiara R, del Bimbo A, and Torniai C, "Video Annotation with Pictorially Enriched Ontologies". In: Proceedings of the Multimedia and Expo, pp. 1428 - 1431 July 2005

[14] Tsinaraki C, Polydoros P, Kazasis F, and Christodoulakis S, "Ontology-Based Semantic Indexing for MPEG-7 and TV-Anytime Audiovisual Content", In: Proceedings of the ACM Multimedia Tools and Applications, pp. 299- 325, 2005.

[15] The DARPA Agent Markup Language. DAML Ontology Library, Soccer Ontology. Available at http://www.daml.org/ontologies/273. Accessed on 05/07/2007.

[16] Natural Language Processing Group – Sheffield University. GATE – General Architecture Text Engineering. Available at http://gate.ac.uk/. Accessed on 01/08/2007.

[17] Cunningham H, Maynard D, Bontcheva K, Tablan V. "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications", In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.

[18] Bontcheva K, Tablan V, Maynard D, Cunningham H, "Evolving GATE to Meet New Challenges in Language Engineering", Natural Language Engineering. 10 (3/4), pp. 349-373. 2004.