Open access • Proceedings Article • DOI:10.1109/MMCS.1999.779303

On the automated interpretation and indexing of American Football

Mihai Lazarescu, Svetha Venkatesh, Geoff West, Terry Caelli

Institutions: Curtin University, Ohio State University

Published on: 07 Jun 1999 - International Conference on Multimedia Computing and Systems

Topics: Domain knowledge, Knowledge base, Knowledge representation and reasoning, Natural language and Natural language understanding

On the Automated Interpretation and Indexing of American Football

Mihai Lazarescu, Svetha Venkatesh, Geoff West
Curtin University of Technology
GPO Box U1987, Perth 6001, W.A.
lazaresc@cs.curtin.edu.au

Terry Caelli
Ohio State University
1216 Kinnear Rd., Columbus, Ohio 43212
caelli@cfm.ohio-state.edu

Abstract

This work combines natural language understanding and image processing with incremental learning to develop a system that can automatically interpret and index American Football. We have developed a model for representing spatio-temporal characteristics of multiple objects in dynamic scenes in this domain. Our representation combines expert knowledge, domain knowledge, spatial knowledge and temporal knowledge. We also present an incremental learning algorithm to improve the knowledge base as well as to keep previously developed concepts consistent with new data. The advantages of the incremental learning algorithm are that is that it does not split concepts and it generates a compact conceptual hierarchy which does not store instances.

1 Introduction

Natural language understanding, image understanding and machine learning are three important areas of research in artificial intelligence. In this paper we present a system that combines natural language and image understanding to *recognise* American Football plays. The system dynamically *expands* its knowledge base through incremental learning. The main benefit of combining image information from the American Football tapes and natural language commentary is that it allows for information to be fused from both sources and therefore video is no longer treated as a silent movie.

This enables a rich information set to be derived from the text which, when combined with the analysis of the player positions in space and time from the video, can be used to understand the video at a higher semantic level than would be possible by processing the video data alone.

We further explore how machine learning can be used to dynamically update the system's knowledge base, by developing new models or by modifying existing ones, and hence improve the system's performance.

From the relatively small number of attempts that have been made to connect image processing and natural language, of great relevance to our work are the VITRA and News-on-Demand systems.

The VITRA (VIsual TRANslator) [5] is one of the projects which tries to integrate computer vision and the generation of natural language expressions for the description of image sequences. The domains investigated in the VITRA project have been traffic scenes and *soccer sequences*.

Another project which combines natural language understanding with image processing is Informedia: News-on-Demand [11]. The Informedia [1] digital video library project at Carnegie Mellon University is creating a digital library of text, images, video and audio data available for full content search and retrieval. News-on-Demand [4] is a fully-automatic system that monitors TV, radio and text of news radio material and allows selective retrieval of news stories based on spoken queries.

Several methods of learning have been developed but the one that is best suited for real world situations is *incremental learning*. Human learning is incremental. There are several important issues in incremental learning which have been identified: bias, concept drift, memory size and forgetting. Several systems (GEM [9], COBWEB [2] and UNIMEM [8]) have been developed to address these issues.

The objective of our work is to use natural language understanding and image processing to query and learn American Football plays from video tapes.

The paper is organised as follows. In section two, we describe in detail the knowledge structure that is used in the recognition routine. In the third section we look at the data analysed by the system, while in section four we discuss the aspects of learning in our recognition system. Section five describes the results whilst the conclusions are presented in section six.

0-7695-0253-9/99 $10.00 © 1999 IEEE

2 Play Model

The play model description we propose to represent an American Football play has the following features (for a detailed description see Lazarescu *et al* [7]):

The **play class** — there are three classes of plays in American Football which are defined by the main action that takes place in the play. The play classes possible are defined by set A where: A = {*offensive, defensive,special*}.

The **significant players and actions in the play** — each play involves significant players performing predefined tasks. We define as significant the players whose actions are very likely to have a major impact on the main action and outcome of the play. The player significance is defined by set B where: B = {*low, average, high, very-high*}.

In an analogous manner, the player position, significance and action is defined for all plays. The possible actions are defined by set C where: C = {*pass, toss, give, hand, catch, get, receive, fake, tackle, block, push, grab, intercept*}.

The **player combination** — each play requires a specific combination of players. For example an offensive pass play might have the following player combination: 4 blockers, 2 corner-backs, 2 receivers, 1 guard, 1 snap and 1 quarterback.

The **temporal sequencing** of the actions in the play — in each play certain actions take place at specific times and in a certain order. The temporal representation of the play model is based on a set of simple movements combined with a play action. The *play action* is the main action in the play already defined in the first part of the play model (see significant players and actions definitions). The *movement* describes the type of movement of a given player *at an instance in time*. A player can either move to a position or make a turn. The movements *turn-left* and *turn-right* are relative to the way in which the football player is facing. The possible movements of a player are defined by set D where: D = {*move, turn-left, turn-right*}.

The order and timing of the combined movement of the players is described with the help of five of Allen's temporal primitives [3]. The set of primitives used in the temporal representation is defined by the set E where: E = {*before, after, during, meets, equals*}.

To derive the temporal relationships between the players, the spatial model of a play (all spatial models are extracted from coaching books) is considered and the movement of the players is described as a series of moves and turns. The time and sequencing of the moves by the players are also recorded. Then using the temporal primitives we derive relations such as *player1(move position2) equals player5(move position 8)*.

The **movement of individual players in the play** — players generally have specific movement patterns in specific plays. The movement of the player *over the entire play period* is described in terms of *shape* and *distance*. The shape of the movement indicates whether the movement is **straight** or **contains turns**. The distance covered by the player during the play can be either short or long. To determine the type of distance, the system uses the player's starting and finishing positions and the position of the defensive line. If the player started *behind* or *in the same line* with the defensive line and finished *in-front* of the defensive line *then* the distance covered is considered to be *long*. Otherwise the distance is *short*.

The possible overall movements are defined by set F where: F ={*short-and-straight, short-and-turn, short-and-many-turns, long-and-straight, long-and-turn, long-and-many-turns*}.

The **relationships between a player and the other teammates** at different instances throughout the duration of play are described symbolically and derived from four reference points which are assumed to be known at all times. These are the left boundary, the right boundary, the opponent's goal and the team's own goal line and they produce the relationships: *in-front, behind, to-the-left* and *to-the-right*. Two more relationships can be further defined: *in-line-with-horizontal* and *in-line-with-vertical*. The references are dependant on which way the attacking team is *facing*. The possible relationships are defined by the set G where: G = {*in-front, behind, to-the-left, to-the-right, in-line-with-horizontal, in-line-with-vertical*}.

To build the play models we have created a database of play information that includes data about the player combinations, play actions, player significance and player movement over the duration of a "by the book" play. The system uses a well defined semi-dynamic hierarchical structure to store the models of the American Football plays. The structure was designed specifically to enable the system to recognise and learn (when there is evidence that suggests learning from the new data is justified) from the data it receives. In the following section we describe in detail the type of data received by the system and the processing applied to convert it to a meaningful form.

3 Data Inputs

We use three sources of information: the natural language commentary of the football game, geometrical information about the play and domain knowledge to determine the type of play.

The natural language commentary plays a crucial role in recognition because it is an expert description of the action in the video and therefore can be used to obtain much information about the play. The transcript of the commentary provides three clues about the play in the video. The first, is the type of action that takes place in the game, such

as "the ball was passed". The second clue gives the name and the type of the players that have been involved in the play action. For example "PlayerX throws the ball high to receiver PlayerY" describes a play "pass", in which PlayerY is a receiver. The third clue that can be obtained from the natural language text is the game statistics. The system searches the input commentary for a set of predefined keywords. The keywords found are passed onto a module which attempts to assemble the keywords into predefined patterns that describe different aspects of the game ranging from player positions to play actions. The patterns extracted from the commentary text are then used to fill two types of frames: a player frame and a play frame.

The geometrical information extracted from the video is made up of a list of unlabelled player coordinates from the frames in the video sequence. This information is used to determine direction of movement and acceleration of players in the video as well as the initial and final player setups and positions.

Using the information from the video, three symbolic descriptions of the player movement are derived. The first description details the movement of the player of the entire play as a series of *moves* and *turns*. The description is used in the temporal part of the play model. The second description is used to keep track of the player's movement at each instance of time and is used to reconstruct the dynamics of the play. The third description details the player movement over the entire play in terms of shape and distance.

The information from the video is assumed to be incomplete and can possibly contain noisy data about the player coordinates at different frames in the video.

When the natural language commentary and the video data processing is completed, the resulting information frames and sets of coordinates are passed on to the search module which attempts to find a match for the query play. A detailed description of the processing involved in finding the best match for the query is presented in Lazarescu *et al* [7].

4 The Incremental Learning Algorithm

The model representation of the American Football plays is complex as it includes the relationships between the objects over time as well as the temporal sequencing of these relationships. Each play model consists of a large number of features and as a result the data structure used to store the models has been designed to minimise data overlap and to enable forgetting.

Overall we have combined selected parts of two well known conceptual clustering systems – UNIMEM and COBWEB – with our own style of forgetting and data representation to develop the algorithm.

4.1 Data Structure

The concepts developed by our algorithm are stored in a hierarchy in which all nodes share all the features observed in the training instances. Unlike the case of UNIMEM and COBWEB, the nodes in our structure do not store the individual instances. Instead the range of values of any feature is defined with the help of a set which covers all the values encountered in the training instances which were used in the generalisation process. There are several reasons for choosing this type of representation. The reason for not storing the instances is that by retaining them, the concept tends to become fragmented when dealing with several distinct variants and therefore the system is no longer able to recognise the actual concept or track its drift. The second reason is that by using all observed features to build the concepts, the system is able to handle cases of missing or noisy data. The third reason is that it substantially reduces the size of the concept and as a result the amount of memory required to store the hierarchy.

The way in which the system sorts the instances is as follows. Each new instance is compared with the current node in the hierarchy to determine if there is enough evidence to justify the update of the current node. Each node produces an evidence score which determines whether the instance does or does not match the current node. The score is computed as a function of age. In our concept representation, each feature has a set of values associated with it and each one of the values in the set has an age associated with it. Multiple concepts/nodes can be updated (provided enough evidence was found) by the same instance. While this procedure results in the system updating nodes which should not be updated when one considers the overall set of instances, results show that over time the unnecessary modifications are "aged out" of the concepts. The main reason for choosing to update multiple nodes is that it is a simple way of representing a fuzzy match between the concept and the instance which is more appropriate than an absolute match.

4.2 Forgetting

Forgetting offers the following advantages: it helps restructure the acquired knowledge, it shortens the retrieval time and it helps to reduce the memory size required for processing [6].

Research in cognitive psychology [10, 12] suggests that human beings have an *immediate memory* used to store information for only a couple of seconds, a *short term memory* which is used to store information (in a ordered manner) for time intervals ranging from minutes to hours and a *long term memory* which is used to store information in a associative and highly structured manner for very long peri-

ods of time (years). Hence we have implemented an ageing procedure which simulates short-term memory (set of facts just observed and which can be discarded if not reinforced), medium-term memory (concepts already reinforced but not up to the level justifying permanent storage and which can still be forgotten) and permanent memory (concepts which have been reinforced so many times that it justifies permanent storage and which cannot be forgotten).

The algorithm uses ageing at two levels: the *data level* and the *concept level*. The data level refers to the data which is contained in the concept. Each attribute in the model has one or more generalised symbolic descriptions and each description has an age value associated with it. Similarly the concepts in the hierarchy have an age value associated with each of them.

There are three stages at the data level: *system-seen* (equivalent to the human *immediate memory*), *system-learned* (equivalent to the human *short-term memory*) and *expert-learned* (equivalent to the human *long-term memory*). The difference between the three stages is the way in which the age of the data is updated. Any data that has reached the *expert-learned* stage no longer ages and therefore it cannot be forgotten. The difference between *system-seen* and *system-learned* is the amount by which the age of the data increases or decreases.

The age of the data in a given concept is updated only when the system has determined that there is enough evidence for the concept to be modified. To determine whether enough evidence exists to justify any concept update, the system computes a similarity score.

If the new instance is found to be similar to the model, then the age of each of the attribute values (provided it is at the *computer-seen* or *computer-learned* stage) is reassessed. If there is an attribute value in the concept that matches its corresponding value in the new instance, then that attribute value is reinforced and its age is decreased. If the attribute value in the concept does not match its corresponding value in the instance, then the age of the value is increased.

The algorithm we have developed offers several advantages. The first advantage is that we use a refined (and we believe more appropriate) type of forgetting which is guided by the evidence computed from the training data. Our *ageing* procedure *is applied locally and only when there is enough evidence to justify feature value update.*

The second major advantage is that the actual hierarchy is compact and *it does not store any instance that it observes*. This is a major benefit especially in cases which involve dealing with large datasets with large numbers of features.

Since the hierarchy is more compact, the system requires less time to access and update the concepts (when necessary).

Finally *by using concepts which do not store instances, our algorithm is able to better handle the problem of concept drift* (the problem of concepts becoming fragmented has been eliminated).

5 Results

At the beginning of the experiment the system had a training session in which instances of 50 basic plays were inputed. The play models developed covered 5 basic formations (Pro-formation, I-formation, Single-Back-formation, Goal-line-formation and Far-formation) performing a combination of 43 running and passing plays. At the end of the training stage the system's memory contained 43 play models.

The test data consisted of queries relating to 3, 4 or 5 players. Also the queries were arranged into two sets. The first set contained "easy" queries with only one of the players in the query having a different movement to that in the play model. The second set contained "hard" queries with one, two or three players having different movements. A detailed run of what the system learns from a simple query is shown next.

A test consisted of a query play (which could be either easy or hard) which the system would attempt to match against the play models in the system's memory. The best five matches are returned as possible solutions to the query. A total of 300 different queries were used for the tests (103 easy queries and 200 hard queries). When the classification was carried out without any incremental learning we obtained the results shown in Tables 1 and 2. In the first set the recognition success rate was 65% while with the second set, the success rate was only 49%. In most cases where it failed the system was still able to identify the correct basic formation. When the classification was carried out with incre-

Formation	Tests Run	Correct	Incorrect
Pro	40	28	12
I	40	26	14
Single-Back	23	13	10

Table 1. First set of tests — No Learning (Easy Queries).

Formation	Tests Run	Correct	Incorrect
Pro	80	45	35
I	60	27	33
Single-Back	40	18	22
Goal-line	20	8	12

Table 2. Second set of tests — No Learning (Hard Queries).

mental learning we obtained the results shown in Tables 3 and 4. For the first set the recognition success rate was 80% while for the second set the success rate was 64%. The ac-

Formation	Tests Run	Correct	Incorrect
Pro	40	36	4
I	40	33	7
Single-Back	23	14	9

Table 3. First set of tests — with Incremental Learning (Easy Queries).

Formation	Tests Run	Correct	Incorrect
Pro	80	52	28
I	60	42	18
Single-Back	40	21	19
Goal-line	20	13	7

Table 4. Second set of tests — with Incremental Learning (Hard Queries).

curacy of the classification was significantly affected by the hypothesis generated by the system. In many of the cases where it failed to accurately classify the query, the system generated a hypothesis which made it difficult to distinguish between the basic formations. This was due to the fact that the formations *Far-formation* and *Pro-formation*, as well as *I-formation* and *Goal-line-formation* are very *similar*. The main difference between the two cases (with *vs* without learning) is that the system with learning is able to handle the situations where the variations between the query and model are significant (hard queries involving 5 players) more successfully.

6 Conclusions

We have developed a model for representing multiple labelled objects in a dynamic scene. Our representation combines several types of knowledge: expert knowledge, domain knowledge, spatial knowledge and temporal knowledge. This is used in a system that is able to recognise American Football plays. The system has the advantages that it is not dependent on accurate low level data and its search is driven by expert input which involves cross checking its input from the several sources it has available. We also present an incremental learning algorithm to improve the knowledge base as well as to keep previously developed concepts consistent with new data. The advantages of the incremental learning algorithm are that is that it does not split concepts and it generates a compact conceptual hierarchy which does not store instances.

References

[1] The CMU Informedia Project. http://www.informedia.cs.cmu.edu.

[2] D. H. Fisher. Knowledge aquisition via incremental conceptual clustering. In J. W. Shavlik and T. G. Diettrich, editors, *Readings in Machine Learning*, pages 267–284. Morgan Kauffman, 1990.

[3] H. W. Geusgen. Spatial reasoning based on allen temporal logic. Technical report, International Computer Science Institute, Berkley, Aug. 1989.

[4] A. G. Hauptmann, M. J. Witbrock, A. I. Rudnicky, and S. Reed. Speech for multimedia information retrieval. In *UIST-95 Eighth Annual Symposium for User Interface Software and Techonology*, Nov. 1995.

[5] G. Herzog and K. Rohr. Integrating vision and language: Towards automatic description of human movements. In *Proceedings of the 19th Annual German Conference on Artificial Intelligence (KI-95)*, June 1995.

[6] M. Kubat and I. Krizakova. Forgetting and aging of knowledge in concept formation. *Applied Artificial Intelligence*, 6:195–206, 1992.

[7] M. Lazarescu, S. Venkatesh, G. West, and T. Caelli. Classifying and incrementally learning American Football plays using natural language and video processing. Technical Report 2, Curtin University of Technology, May 1998.

[8] M. Lebowitz. Experiments with incremental concept formation : Unimem. *Machine Learning*, 2(2):103–138, 1987.

[9] R. E. Reinke and R. S. Michalski. Incremental learning of concept descriptions : A method and experimental results. *Machine Learning*, pages 263–289, 1984.

[10] G. Tiberghien. Natural and artificial memory systems. In J. Demongeot, T. Herve, V. Rialle, and C. Roche, editors, *Artificial Intelligence and Cognitive Sciences*, chapter 16, pages 221–237. Manchester University Press, 1988.

[11] M. Witbrock, A. Hauptmann, and M. Cristel. News-on-demand - an application of informedia technology. *D-LIB magazine*, 1995.

[12] J. Z. Young. *Programs of the Brain.* Oxford University Press, 1978.