Unsupervised Content-Based Indexing of Sports Video

Michael Fleischman
MIT Media Laboratory
mbf@mit.edu

Deb Roy
MIT Media Laboratory
dkroy@media.mit.edu

ABSTRACT

This paper presents a methodology for automatically indexing a large corpus of broadcast baseball games using an unsupervised content-based approach. The method relies on the learning of a grounded language model which maps query terms to the non-linguistic context to which they refer. Grounded language models are learned from a large, unlabeled corpus of video events. Events are represented using a codebook of automatically discovered temporal patterns of low level features extracted from the raw video. These patterns are associated with words extracted from the closed captioning text using a generalization of Latent Dirichlet Allocation. We evaluate the benefit of the grounded language model by extending a traditional language model based approach to information retrieval. Experimental results indicate that using a grounded language model nearly doubles performance on a held out test set.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding – *Video Analysis*

General Terms

Algorithms, Experimentation

Keywords

Video retrieval, grounded language models, sports video, Latent Dirichlet Allocation, temporal data mining, unsupervised content-based indexing.

1 INTRODUCTION

The decreasing cost of data storage and the increasing use of digital video recorders is driving the need for more advanced methods for video search. One popular proposal for facilitating search in video relies on using traditional information retrieval (IR) techniques to search the speech uttered during a video (e.g., [12]). Such methods are popular because of their scalability and the lack of human supervision required to index large corpora. However, applying such methods to searching sports video faces serious challenges, even when speech transcriptions are provided (for example, in the closed captioning stream).

Unlike the case with text documents, the occurrence of a query term in a video is often not enough to assume the video's relevance to that query. For example, when searching through video of baseball games, returning all clips in which the phrase "home run" occurs, results primarily in video of events where a home run does not actually occur. This follows from the fact that in sports, as in life, people often talk not about what is currently happening, but rather, they talk about what did, might, or will happen in the future.

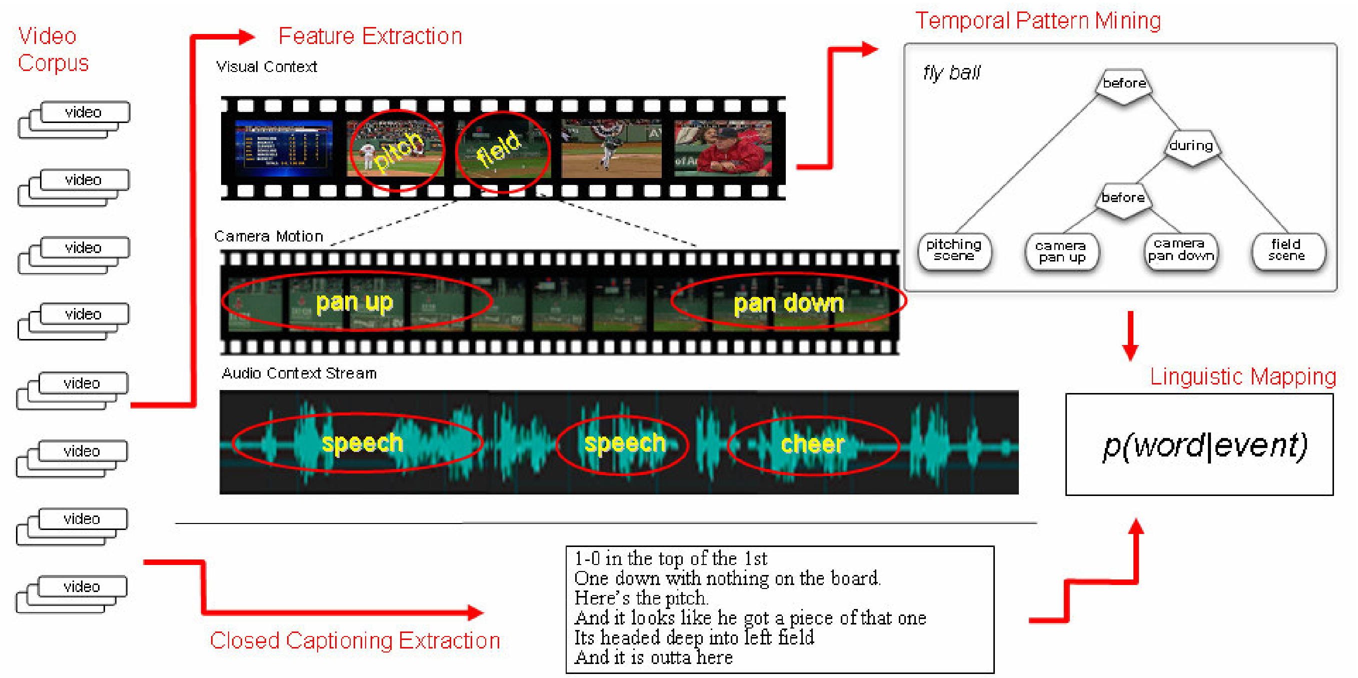
Traditional IR techniques cannot address such problems because they model the meaning of a query term strictly by that term's relationship to other terms. To build systems that successfully search video, IR techniques must exploit not just linguistic information but also elements of the non-linguistic context that surrounds language use. A great deal of research has addressed this issue by designing video search techniques that rely on supervised methods to classify events (see [22] for a review). The majority of these systems do not index events by natural language query terms (as traditional IR approaches do), but rather, categorize events using classifiers trained on hand labeled examples of predefined event types (e.g. *home runs*).[1] Although these approaches can be useful, such supervised approaches to video retrieval are labor intensive both for the system designers, who must label examples and train the concept classifiers, as well as for the system users, who may be required to re-write their queries to match the system's predefined event types.

In this paper, we present an unsupervised method for content-based video indexing of sports video. The method maintains the advantages of traditional IR approaches while incorporating contextual information in an unsupervised manner. The method is based on the learning of a grounded language model; a framework motivated by research on computational models of human verb learning [8]. We model the meaning of a word as a probabilistic mapping between words and representations of the non-linguistic events to which those words refer. To represent events in video, we follow recent work on video surveillance in which complex events are represented as temporal relations between lower level sub-events (e.g., [14]). While in the surveillance domain, hand crafted event representations have been used successfully, the greater variability of content in broadcast sports demands an automatic method for designing event representations.

Our approach operates in three phases: first, raw video data is abstracted into multiple streams of discrete features. Temporal data mining is then used to generate a codebook of temporal patterns used to represent video events. These temporal pattern

MIR'07, September 28-29, 2007, Augsburg, Bavaria, Germany.
Copyright 2007 ACM 978-1-59593-778-0/07/0009...$5.00.

[1] For an interesting exception see [23] in which hand chosen terms from the closed captioning stream are used to index a limited set of predefined events.



1-0 in the top of the 1st
One down with nothing on the board.
Here's the pitch.
And it looks like he got a piece of that one
Its headed deep into left field
And it is outta here

Figure 1. Learning grounded language models operates in three phases: first the raw video is abstracted into parallel streams corresponding to visual context, camera motion, and audio context features. Second, temporal data mining is used to discover a codebook of temporal patterns. Finally, words from the closed captioning are mapped onto the encoded event representations.

representations are then mapped to words in the closed captioning text using a generalization of Latent Dirichlet Allocation ([5],[17]). In the following sections we detail this approach and examine its effectiveness in retrieving video events from a held out test set of broadcast baseball games. Results indicate that performance of the system using the grounded language model is significantly better than traditional text based approaches.

2 GROUNDED LANGUAGE MODELING

Our framework for learning grounded language models operates in three phases (see Figure 1): first, raw video data is abstracted into multiple streams of discrete features. Temporal data mining techniques are then applied to these feature streams to discover hierarchical temporal patterns. These temporal patterns form a codebook that is used to generate event representations which are then mapped to words in the closed caption stream.

2.1 Feature Extraction

The first step in representing events in video is to abstract the very high dimensional raw video data into more semantically meaningful streams of information. Ideally, these streams would correspond to basic events that occur in sports video (e.g., hitting, throwing, catching, kicking, etc.). Due to the limitations of computer vision techniques, extracting such ideal features is often infeasible. However, by exploiting the "language of film" that is used to produce sports video, informative features can be extracted that are also easy to compute. Thus, although we cannot easily identify a player hitting the ball, we can easily detect features that correlate with hitting: e.g., when a scene focusing on the pitching mound immediately jumps to one zooming in on the field (see Figure 1). While such correlations are not perfect, pilot tests show that baseball events can be classified using such features [10]. Although many feature types can be extracted, we focus on only three: visual context, camera motion, and audio context.

2.1.1 Visual Context

Visual context features encode general properties of the visual scene in a video segment. Such features are relatively easy to extract in comparison to the classification of full events in video, requiring less training data and achieving higher performance. The first step in extracting such features is to split the raw video into "shots" based on changes in the visual scene due to editing (e.g., jumping from a close up of the pitcher to a wide angle of the field). Shot detection is a well studied problem in multimedia research; in this work, we use the method of Tardini et al. [18] because of its speed and proven performance on sports video.

After a game is segmented into shots, individual key frames are selected from the shot and represented as vectors of low level features (see Table 1 for the complete list of features used). Boosted decision tree classifiers are trained [19] to categorize each key frame into one of three categories: *pitching-scene, field-scene*, or *other*. For this three way classification, we achieved approximately 96% accuracy on a held out test set.

Given these categorizations, a second boosted decision tree classifier is then used to sub-classify the field shots into the following six categories: *infield, outfield, wall, base, running, and misc.* Performance on a held out test set showed approximately 90% accuracy.

Feature group	Type	Region/Statistic	Description
Camera motion stats	*median*	*pan/tilt/zoom*	Median camera motion during shot in each of three directions (pan/tilt/zoom).
Pixel stats	*grass, field*	*total, top, bottom, left, right*	Number of pixels with characteristic (of grass, field) color values in each of the names regions of the key frame.
Pixel ratios	*grass/grass, field/field, grass/field*	*total/total, top/bottom, left/right*	Ratio of pixels types by region of key frame (e.g., number of grass pixels in whole frame vs. number of soil pixels in whole frame)..
Pixel distributions	*grass*	*horizontal, vertical*	Distribution of pixel type projected onto region (from Pei and Chen, 2003).
Entropy stats	*entropy*	*total, center, top, bottom, quadrants*	Entropy of key frame by region.
Entropy ratios	*entropy*	*top/bottom, center/total*	Ratio of entropy by region.
Line stats	*lines*	*total lines, max slope, max length*	Slope and length of longest line segment found in key frame (lines are found using Canny edge detection and Hough transforms)
Color stats	*RGB colors*	*max color, max color size*	Number of pixels in key frame for each of 16 RGB binned colors.
Blob stats	*face, max color*	*number, max size, avg. size, std. size*	Connected component statistics for face colored pixels and max (RGB) colored pixels.

Table 1. List of features used to abstract visual context.

2.1.2 Camera Motion

Whereas visual context features provide information about the global situation that is being observed, camera motion features afford more precise information about the actions occurring in the video. The intuition here is that the camera is a stand in for a viewer's focus of attention. As action in the video takes place, the camera moves to follow it, mirroring the action itself, and providing an informative feature for event representation.

Detecting camera motion (i.e., pan/tilt/zoom) is a well-studied problem in video analysis. We use the system of Bouthemy et al. [6] which computes the pan, tilt, and zoom motions using the parameters of a two-dimensional affine model fit to every pair of sequential frames in a video segment. The output of this system is then clustered into characteristic camera motions (e.g. zooming in fast while panning slightly left) using a 15 state 1st order Hidden Markov Model, implemented in the Graphical Modeling Toolkit.[2]

2.1.3 Audio Context

Abstracting audio context from raw audio requires both sound classification and segmentation. We employ a sound classification system based on supervised learning algorithms in which binary classifiers for *speech, excited_speech, cheering*, and *music* are built using boosted decision trees [19]. Classification operates on a sequence of overlapping 30 ms "frames" chunked from the audio stream. For each frame, a feature vector is computed using, MFCCs (often used in speaker identification and speech detection tasks), as well as energy, the number of zero crossings, spectral entropy, and relative power between different frequency bands. The classifier is applied to each frame, producing a sequence of class labels. These labels are then smoothed using a dynamic programming cost minimization algorithm (similar to those used in Hidden Markov Models). Performance of this system achieves between 78% and 94% accuracy.

2.2 Temporal Pattern Mining

In this step, temporal patterns are mined from the features abstracted from the raw video data. As described above, ideal semantic features (such as hitting and catching) cannot be extracted easily from video. We hypothesize that finding temporal patterns between audio, visual and camera motion features can produce representations that are highly correlated with sports events. Importantly, such temporal patterns are not strictly sequential, but rather, are composed of features that can occur in complex and varied temporal relations to each other. For example, Figure 1 shows the representation for a fly ball event that is composed of: a *camera panning up* followed by a *camera pan down*, occurring during a *field scene,* and before a *pitching scene*.

Following previous work in video content classification [9], we use techniques from temporal data mining to discover event patterns from feature streams. The algorithm we use is fully unsupervised. It processes feature streams by examining the relations that occur between individual features within a moving time window. Following Allen [1], any two features that occur within this window must be in one of seven temporal relations with each other (e.g. *before, during, etc.*). The algorithm keeps track of how often each of these relations is observed, and after the entire video corpus is analyzed, uses chi-square analyses to determine which relations are significant. The algorithm iterates through the data, and relations between individual features that are found significant in one iteration (e.g. [BEFORE, *camera panning up, camera panning down*]), are themselves treated as individual features in the next. This allows the system to build up higher-order nested relations in each iteration (e.g. [DURING, [BEFORE, *camera panning up, camera panning down*], *field scene*]]).

[2] http://ssli.ee.washington.edu/~bilmes/gmtk/

The temporal patterns found significant in this way make up a codebook which is used as a basis for representing events in video. Given an unseen video event, the raw video is abstracted into parallel feature streams (as described in Section 2.1). These feature streams are then scanned, looking for any temporal patterns that match those found in the discovered codebook (as well as their nested sub-patterns). The list of patterns that match form a feature vector representation of the event, in which each element in the vector corresponds to the duration of occurrence of a matched pattern from the codebook. Thus, each event is represented as a vector of real valued features, each value being the duration that a particular pattern (from the codebook) was observed during the event.

2.3 Linguistic Mapping

The last step in building the grounded language model is to map words onto the encoded event representations. We equate the learning of this mapping to the problem of estimating the conditional probability distribution of a word given a video event representation. As in previous work [7], we generate these estimates using techniques similar to those used in Machine Translation (MT).

Mappings between words and pattern features for an event are estimated based on a *paired* corpus of video event representations and the corresponding words uttered during that event. We generate this paired corpus from a corpus of raw video by first abstracting each video into the feature streams described in Section 2.1.1. For every shot classified as a *pitching scene*, a new instance is created in the paired corpus corresponding to an event that starts at the beginning of that shot and ends exactly four shots after. This definition of an event follows from the fact that most events in baseball must start with a pitch and usually do not last longer than four shots [11]. For each of the events in this paired corpus, a pattern feature representation is generated as described in Section 2.2. These video representations are then paired with all the words from the closed captioning that occur during that event (plus/minus 10 seconds). Because closed captioning is often not time synched with the audio, we use the technique described in [12] to align the closed captioning text with the announcers' speech.

While the MT framework shows promise, recent work on related tasks in automatic image annotation ([2], [4]) and natural language processing [17] have demonstrated the advantages of using hierarchical graphical models. In this work, we follow closely the Author Topic (AT) model [17], which is a generalization of Latent Dirichlet Allocation (LDA) [5].

LDA is a technique related to Latent Semantic Analysis ([15], [13]) that was developed to model the topics discussed in a large corpus of documents. The model assumes that each document is made up of a mixture of topics, and that each word in the document is generated from a probability distribution associated with one of those topics. The AT model generalizes LDA, saying that the mixture of topics is not dependent on the document itself, but rather on the authors who wrote it.

According to this model, for each word in a document, an author is chosen uniformly from the distribution of the document's authors. Then, a topic is chosen from a distribution of topics for that particular author. Finally, the word is generated from that chosen topic. Given this model, we can express the probability of the words in a document (W) given its authors (A) as:

$$p(W \mid A) = \prod_{m \in W} \frac{1}{A_d} \sum_{x \in A} \sum_{z \in T} p(m \mid z) p(z \mid x) \quad \textbf{(1)}$$

where T is the set of latent topics that are induced given a large set of training data.

We make use of the AT model to learn a grounded language model, by making an analogy between documents and video events. In our framework, the words in a document correspond to the closed captioning words spoken during an event, while the authors of a document correspond to the temporal patterns representing the activity that occurred during that event. We modify the model slightly, such that, unlike authors which are chosen from a uniform distribution, patterns are chosen from a multinomial distribution based upon the duration of the pattern. The intuition being that patterns which occur for a longer duration are more salient and must be given greater weight in the generative process. Thus, we rewrite (1) to give the probability of words during an event (W) given the vector of observed temporal patterns (P) as:

$$p(W \mid P) = \prod_{m \in W} \sum_{x \in P} \sum_{z \in T} p(m \mid z) p(z \mid x) p(x) \quad \textbf{(2)}$$

3 EXPERIMENTS

Work on video IR in the sports domain, e.g. [11], usually focuses on retrieving video data using a set of supervised classifiers that categorize events into pre-determined concepts (e.g. *homerun, infield out, outfield hit etc.).* Such supervised systems can be seen as discovering mappings from a closed set of query terms to a closed set of events (as in [23]). The goal of our approach, however, is to develop a system which discovers mappings from an open set of query terms to an open set of events[3].

A supervised system can only perform such an open task with the addition of a function that maps (automatically or manually) an open set of terms to its pre-defined set of event classes (as is done in the news domain, see [20]). As developing such a function is beyond the scope of this paper, we focus our evaluations on extending a traditional approach to open term IR, the language modeling approach of Ponte and Croft [16], by incorporating a grounded language model.

In Ponte and Croft [16], documents relevant to a query are ranked based on the probability that each document generated each query term. For video event retrieval, we can similarly rank video events based on the probability that a query term was generated from the closed captioning during that event:

$$p(query \mid event) = \prod_{word}^{query} p(word \mid caption) \quad \textbf{(3)}$$

In our experiments, we follow Ponte and Croft [16]in treating the probability of a word given a caption as an interpolation between the probability of the query term given the words in the caption and the probability of the word in the entire corpus:

$$p(word \mid caption) = \omega * P_{caption}(word \mid caption) + (1-\omega) P_{corpus}(word) \quad \textbf{(4)}$$

Here ω is a weighting coefficient (set to 0.5). We also use a simple add N smoothing technique (N=1e-6) to address issues of sparse data.

In extending the language modeling approach to incorporate contextual information in the video, we make the simplifying

[3] and to do so without hand labeling any events

	Query terms				
Result rank	walks (WALK)	strike out (STRIKEOUT)	left field (LEFT)	it's gone (HOMER)	towards the corner (DOUBLE)
1	WALK	NON_HIGH	LINE_LEFT_SINGLE	NON_HIGH	LINE_LEFT_DOUBLE
2	NON_HIGH	STRIKEOUT	LINE_LEFT_DOUBLE	FLY_LEFT_HOMER	NON_HIGH
3	NON_HIGH	STRIKEOUT	NON_HIGH	GROUND_1ST_OUT	LINE_LEFT_DOUBLE
4	WALK	NON_HIGH	FLY_LEFT_OUT	FLY_LEFT_HOMER	NON_HIGH
5	NON_HIGH	STRIKEOUT	FLY_LEFT_HOMER	NON_HIGH	NON_HIGH
Precision/ Ranked Prec.	0.4 / 0.3	0.6 / 0.353	0.8 / 0.71	0.4 / 0.2	0.4 / 0.333

Table 2. Example output of combined system [i.e., using an alpha=0.5 in equation (5)]. Query terms are displayed with relevant category in parentheses. Query results are presented in ranked order and described by all relevant categories to which they belong (e.g., GROUND_1ST_OUT represents an event where a batter hit a ground ball to 1st base and was called out. NON_HIGH represents a non-highlight event such as a foul ball). Both standard precision and ranked precision metrics are presented.

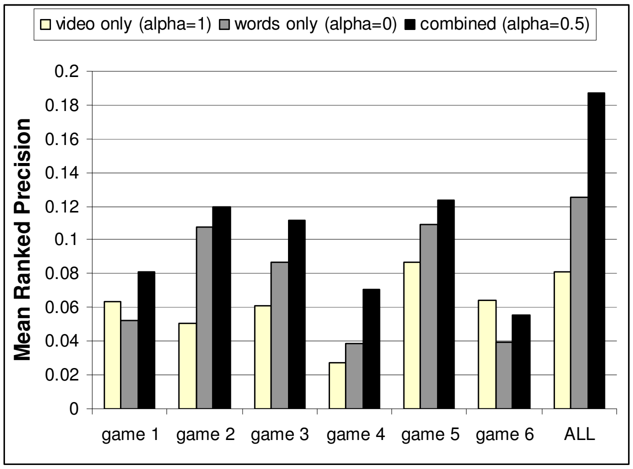


Figure 2. Information retrieval performance of grounded language model (equation 5) using only words (alpha=0), only video features (alpha=1), and using both together (alpha=0.5). Performance is shown on individual games as well as on full held out test set.

assumption that the relevance of an event to a query can be modeled as two independent probabilities: the probability of the query word given the speech of the announcer, and the probability of the word given the video event representation. We formalize this by extending (3):

$$p(query \mid event) = \prod_{word}^{query} p(word \mid caption)^{(1-\alpha)} * p(word \mid video)^{\alpha} \qquad \textbf{(5)}$$

The *p(word|caption)* is estimated using (4), while the *p(word|video)* is estimated as in (2). α is used to weight the models.

3.1 Data

As standardized corpora are unavailable in the sports domain, we recorded 99 Major League Baseball games from the 2006 season totaling approximately 275 hours and 20,000 distinct events. These games represent data from 25 teams in 23 stadiums, broadcast on five different television stations. From this set, six games were held out for testing (15 hours, 1200 events, nine teams, four stations). From this test set, highlights (i.e., events which terminate with the player either *out* or *safe*) were hand annotated for use in evaluation. Each highlight was categorized into one of 13 categories according to the type of the event (e.g., *strikeout vs. homerun)*, the location of the event (*e.g., right field vs. infield*), or the nature of the event (e.g., *fly ball vs. line drive*). (See Figure 3 for a complete listing of categories). Importantly, although only highlights are hand annotated, both highlights and non-highlights are used in the test set. Thus, retrieval operates over the complete set of events in a game (which is significantly more challenging than retrieval from just highlights alone).

Since a standard set of query terms was also unavailable for the sports domain, we automatically generate queries using a technique similar to that used in Berger & Lafferty [3]. For each of the highlight categories described above, a log likelihood ratio is used to generate a measure of how indicative each unigram, bigram, and trigram in the corpus is of a particular category [21]. Query terms are then selected by taking the top 10 ngrams that are most indicative of each category (e.g. "fly ball" for category *flyball*). This gives us a set of queries for each annotated category (130 in all; see Figure 3) for which relevant results can easily be determined (e.g., if a returned event for the query "strike out" is of the *strikeout* category, it is marked relevant).

3.2 Model

Following Steyver et al. [17] we train our AT model using Gibbs sampling , a Markov Chain Monte Carlo technique for obtaining parameter estimates (see Steyvers et al. [17] for more details). We run the sampler on a single chain for 1000 iterations. We set the number of topics to 50, and normalize the pattern durations first by individual pattern across all events, and then for all patterns within an event. The resulting parameter estimates are smoothed using a simple add N smoothing technique, where N=1 for the word by topic counts and N=.01 for the pattern by topic counts.

4 RESULTS

Table 2 shows example outputs of the system run on all events from the six test games (both highlights and non-highlights). The system uses an interpolation between a traditional language modeling retrieval system and a system using the grounded language model [i.e., using an alpha=0.5 in equation (5)]. The

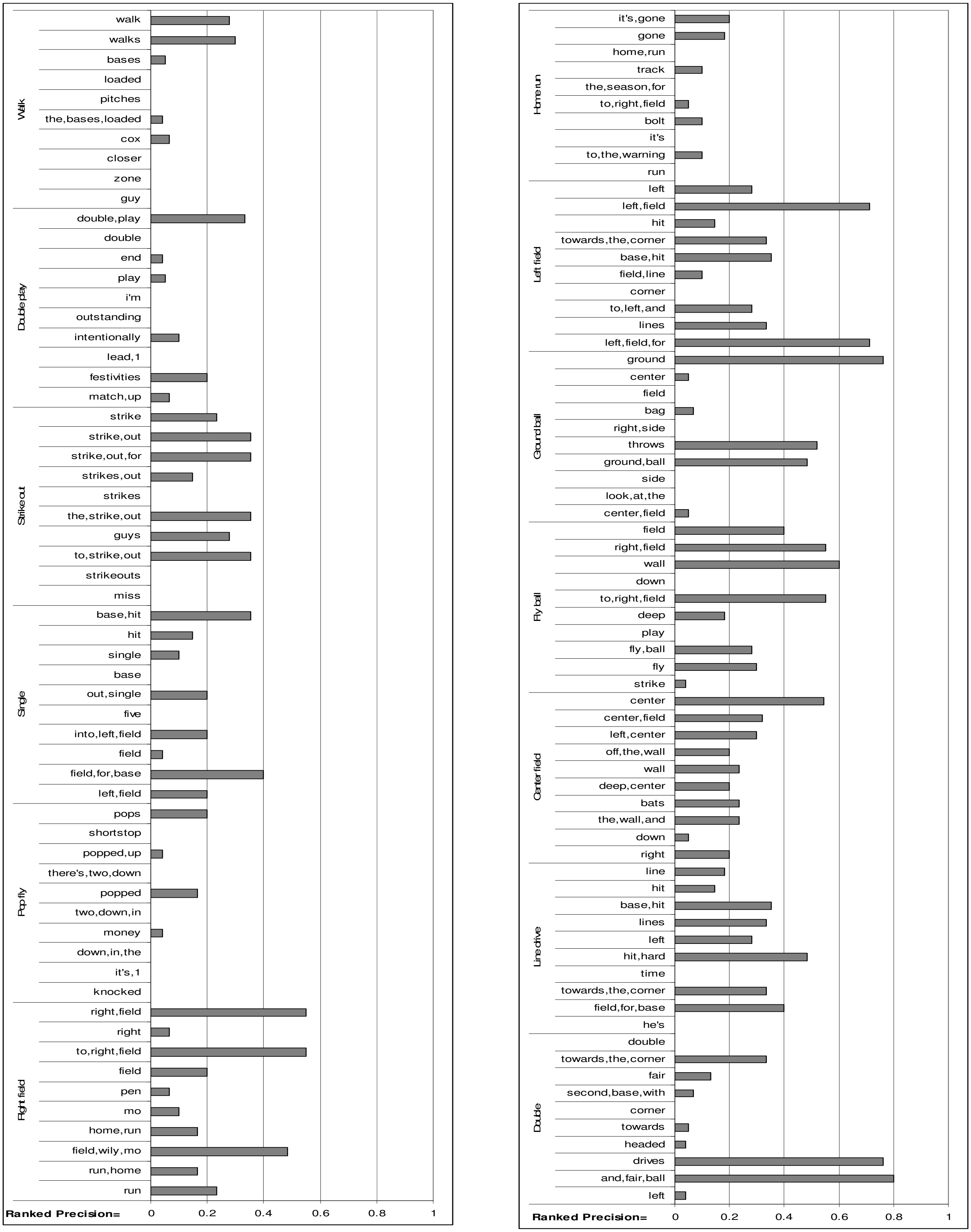


Figure 3. Performance of combined system (alpha=0.5) on individual query terms.

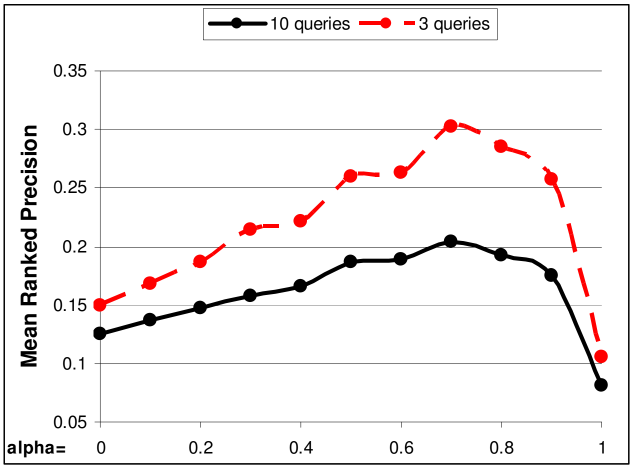


Figure 4. Effect of varying alpha parameter (equation 5) on information retrieval performance. Alpha=0 represents the system using only the traditional language model. Alpha=1 represents the system using only the grounded language model. Results are presented for automatically generated query sets using top 10, and top 3 most indicative query terms.

top five results are returned for each query, and the relevant classes are displayed. Results are reported using precision which indicates the number of relevant results divided by the number of returned results (set to five in these experiments). Results are also reported using the more stringent ranked precision defined as:

$$ranked\_precision = \frac{\sum_{r=1}^{N}(P(r) * rel(r))}{N} \qquad \textbf{(6)}$$

Where r is the rank of the result, P(r) is the precision at rank r, rel(r) is a binary indicator of the relevance of the result at rank r, and N is the number of returned results (set to five in these experiments). Like traditional precision, ranked precision measures the quality of retrieved results, but also punishes systems when relevant results are not returned first.[4] The remainder of the experiments are measured in ranked precision.

Figure 2 compares the performance of a traditional language modeling retrieval system versus a system using only the grounded language model and a system that interpolates between the two [using an alpha=0.5 in equation (5)]. Results are reported for the 130 automatically generated queries described in Section 3.1, run over all events in the test games (both highlights and non-highlights). Comparisons are made for searches within a single corpus made up of all six held out test games, as well as, for searches within each test game individually. For each query, the ranked precision is computed and the mean over all queries is reported.

Figure 3 shows a detailed view of the performance of the combined system (alpha=0.5) for each individual query. Queries are grouped according to their highlight category and ranked precision for each query is reported.

Figure 4, shows the effect on performance of varying the weighting parameter alpha from equation (5). We report results on two sets of queries generated using the automatic technique described in Section 3.1: one taking the top 10 ngrams per highlight category, and one taking only the top three. This second set of queries represents a smaller and cleaner test set to evaluate the performance of the system.

5 DISCUSSION

Figure 2 shows that for five out of six test games, using the grounded language model improves results over traditional IR techniques. This increase is even more evident when searching the complete set of test games. The more detailed results reported in Figure 3 show a large range in performance due in part to the quality of the query term used. Because query terms are generated automatically, as more terms are selected, their quality begins to deteriorate. Thus, by only examining terms with high log-likelihood ratios, we would expect better performance from the system. This is just what is shown in Figure 4.

Here we see that results on just the top three queries generated in the manner described in section 3.1 show markedly better performance than the larger set of test queries. Also in Figure 4, we see the benefit of varying the weight between the grounded and traditional language models. The increased performance is due to the complementary nature of the grounded language model and the traditional language model for IR. As described above, traditional IR approaches return many false positives because of the tendency of announcers to discuss things that are not currently occurring. A grounded language model faces its own challenges, due primarily to limitations in computer vision. By combining the two together, the grounded language model buttresses the traditional approach, leading to significant increases in performance when compared to either system on its own.

6 CONCLUSIONS

We have presented a system for unsupervised content-based indexing of sports video retrieval. The system relies on the learning of a grounded language model which maps query terms to non-linguistic contextual information from the video. By extending traditional language modeling approaches to IR with a grounded language model, the system is able to nearly double the performance over baseline methods.

Unlike most previous efforts to exploit non-linguistic information for video retrieval, our system does not require hand labeled examples of predefined event types. Instead, our system exploits automatically mined temporal patterns of low level features, which can be easily extracted with limited effort and high reliability.

Further, in our system no extra effort is required (by the system or the user) to match a natural language query to the system's predefined set of event types. Rather, events in video are indexed by natural language terms directly (as in traditional IR approaches) allowing retrieval to operate without additional effort.

Currently, we are examining how such grounded language models can improve search on noisier data, in particular, in games without closed captioning where speech must be transcribed

[4] Ranked precision is also similar to Mean Average Precision (MAP), but does not factor in recall. Unlike MAP, ranked precision does not give higher weight to queries for infrequent events. Rather, it strictly expresses the ability of the system to return relevant results in ranked order.

automatically. In future work, we will examine the ability of grounded language models to improve performance for other natural language tasks. Incorporating such contextual information may benefit tasks as diverse as Machine Translation, Summarization, and Automatic Speech Recognition. Finally, we are examining extending this approach to other sports domains such as basketball. In theory, however, our approach is applicable to any domain in which there is discussion of the here-and-now (e.g., home improvement shows, etc.). In future work, we plan to examine the strengths and limitations of grounded language modeling in these domains.

7 ACKNOWLEDGEMENTS

The authors would like to acknowledge Humberto Evans, Stephen Oney, Brandon Roy, and Yansui Wang for their assistance in making this project possible.

8 REFERENCES

[1] Allen, J.F. (1984). A General Model of Action and Time. Artificial Intelligence. 23(2).

[2] Barnard, K, Duygulu, P, de Freitas, N, Forsyth, D, Blei, D, and Jordan, M. (2003), Matching Words and Pictures, Journal of Machine Learning Research, Vol 3.

[3] Berger, A. and Lafferty, J. (1999). Information Retrieval as Statistical Translation. In Proceedings of SIGIR-99.

[4] Blei, D. and Jordan, M. (2003). Modeling annotated data. Proceedings of the 26th International Conference on Research and Development in Information Retrieval, ACM Press, 127–134.

[5] Blei, D. Ng, A., and Jordan, M (2003). "Latent Dirichlet allocation." Journal of Machine Learning Research 3:993–1022.

[6] Bouthemy, P., Gelgon, M., Ganansia, F. (1999). A unified approach to shot change detection and camera motion characterization. IEEE Trans. on Circuits and Systems for Video Technology, 9(7).

[7] Fleischman M, Roy, D. (2007). Situated Models of Meaning for Sports Video Retrieval. HLT/NAACL. Rochester, NY.

[8] Fleischman, M. B. and Roy, D. (2005) Why Verbs are Harder to Learn than Nouns: Initial Insights from a Computational Model of Intention Recognition in Situated Word Learning. 27th Annual Meeting of the Cognitive Science Society, Stresa, Italy.

[9] Fleischman, M., DeCamp, P. Roy, D. (2006). Mining Temporal Patterns of Movement for Video Content Classification. ACM Workshop on Multimedia Information Retrieval.

[10] Fleischman, M., Roy, B., and Roy, D. (2007). Temporal Feature Induction for Sports Highlight Classification. In Proceedings of ACM Multimedia. Augsburg, Germany.

[11] Gong, Y., Han, M., Hua, W., Xu, W. (2004). Maximum entropy model-based baseball highlight detection and classification. Computer Vision and Image Understanding. 96(2).

[12] Hauptmann, A. , Witbrock, M., (1998) Story Segmentation and Detection of Commercials in Broadcast News Video, Advances in Digital Libraries.

[13] Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA.

[14] Hongen, S., Nevatia, R. Bremond, F. (2004). Video-based event recognition: activity representation and probabilistic recognition methods. Computer Vision and Image Understanding. 96(2).

[15] Landauer, T. K. and Dumais, S. T. (1997) A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review, 104(2)* , 211-240.

[16] Ponte, J.M., and Croft, W.B. (1998). A Language Modeling Approach to Information Retrieval. In Proc. of SIGIR'98.

[17] Steyvers, M., Smyth, P., Rosen-Zvi, M., & Griffiths, T. (2004). Probabilistic Author-Topic Models for Information Discovery. The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, Washington.

[18] Tardini, G. Grana C., Marchi, R., Cucchiara, R., (2005). Shot Detection and Motion Analysis for Automatic MPEG-7 Annotation of Sports Videos. In 13th International Conference on Image Analysis and Processing.

[19] Witten, I. and Frank, E. (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005

[20] Worring, M., Snoek, C.. (2006). Semantic Indexing and Retrieval of Video. Tutorial at ACM Multimedia.

[21] Dunning, T. (1993) Accurate methods for the statistics of surprise and coincidence. *Computational Li'nguistics,* 19:61--7'4.

[22] Kokaram, A., Rea, N., Dahyot, R., Tekalp, A., Bouthemy, P., Gros, P., Sezan I. (2006). Browsing Sports Video. IEEE Signal Processing Magazine. 47.

[23] Babaguchi, N., Kawai, Y., and Kitahashi, T. (2002) Event Based Indexing of Broadcast Sports Video by Intermodal Collaboration. IEEE Transactions on Multimedia. (4;1) pgs.68-75.