Grounded Language Modeling for Automatic Speech Recognition of Sports Video

Michael Fleischman
Massachusetts Institute of Technology
Media Laboratory
mbf@mit.edu

Deb Roy
Massachusetts Institute of Technology
Media Laboratory
dkroy@media.mit.edu

Abstract

Grounded language models represent the relationship between words and the non-linguistic context in which they are said. This paper describes how they are learned from large corpora of unlabeled video, and are applied to the task of automatic speech recognition of sports video. Results show that grounded language models improve perplexity and word error rate over text based language models, and further, support video information retrieval better than human generated speech transcriptions.

1 Introduction

Recognizing speech in broadcast video is a necessary precursor to many multimodal applications such as video search and summarization (Snoek and Worring, 2005;). Although performance is often reasonable in controlled environments (such as studio news rooms), automatic speech recognition (ASR) systems have significant difficulty in noisier settings (such as those found in live sports broadcasts) (Wactlar et al., 1996). While many researches have examined how to compensate for such noise using acoustic techniques, few have attempted to leverage information in the visual stream to improve speech recognition performance (for an exception see Murkherjee and Roy, 2003).

In many types of video, however, visual context can provide valuable clues as to what has been said. For example, in video of Major League Baseball games, the likelihood of the phrase "home run" increases dramatically when a home run has actually been hit. This paper describes a method for incorporating such visual information in an ASR system for sports video. The method is based on the use of *grounded language models* to represent the relationship between words and the non-linguistic context to which they refer (Fleischman and Roy, 2007).

Grounded language models are based on research from cognitive science on grounded models of meaning. (for a review see Roy, 2005, and Roy and Reiter, 2005). In such models, the meaning of a word is defined by its relationship to representations of the language users' environment. Thus, for a robot operating in a laboratory setting, words for colors and shapes may be grounded in the outputs of its computer vision system (Roy & Pentland, 2002); while for a simulated agent operating in a virtual world, words for actions and events may be mapped to representations of the agent's plans or goals (Fleischman & Roy, 2005).

This paper extends previous work on grounded models of meaning by learning a grounded language model from naturalistic data collected from broadcast video of Major League Baseball games. A large corpus of unlabeled sports videos is collected and paired with closed captioning transcriptions of the announcers' speech. [1] This corpus is used to train the grounded language model, which like traditional language models encode the prior probability of words for an ASR system. Unlike traditional language models, however, grounded language models represent the probability of a word conditioned not only on the previous word(s), but also on features of the non-linguistic context in which the word was uttered.

Our approach to learning grounded language models operates in two phases. In the first phase, events that occur in the video are represented using hierarchical temporal pattern automatically mined

[1] Closed captioning refers to human transcriptions of speech embedded in the video stream primarily for the hearing impaired. Closed captioning is reasonably accurate (although not perfect) and available on some, but not all, video broadcasts.

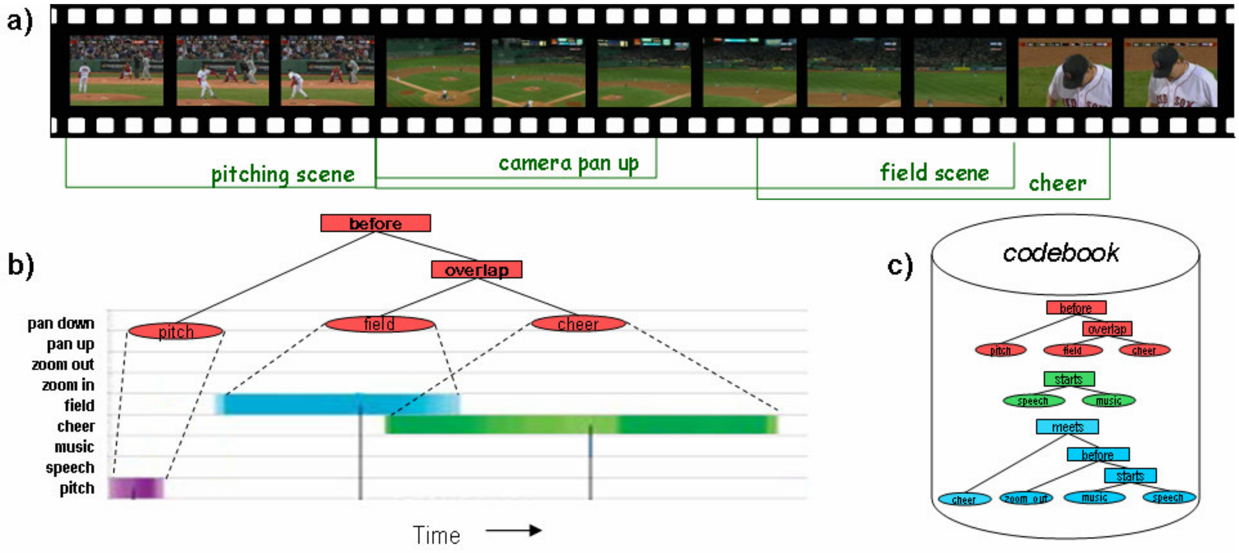

Figure 1. Representing events in video. a) Events are represented by first abstracting the raw video into visual context, camera motion, and audio context features. b) Temporal data mining is then used to discover hierarchical temporal patterns in the parallel streams of features. c) Temporal patterns found significant in each iteration are stored in a codebook that is used to represent high level events in video.

from low level features. In the second phase, a conditional probability distribution is estimated that describes the probability that a word was uttered given such event representations. In the following sections we describe these two aspects of our approach and evaluate the performance of our grounded language model on a speech recognition task using video highlights from Major League Baseball games. Results indicate improved performance using three metrics: perplexity, word error rate, and precision on an information retrieval task.

2 Representing Events in Sports Video

Recent work in video surveillance has demonstrated the benefit of representing complex events as temporal relations between lower level sub-events (Hongen et al., 2004). Thus, to represent events in the sports domain, we would ideally first represent the basic sub events that occur in sports video (e.g., hitting, throwing, catching, running, etc.) and then build up complex events (such as *home run*) as a set of temporal relations between these basic events. Unfortunately, due to the limitations of computer vision techniques, reliably identifying such basic events in video is not feasible. However, sports video does have characteristics that can be exploited to effectively represent complex events.

Like much broadcast video, sports video is highly produced, exploiting many different camera angles and a human director who selects which camera is most appropriate given what is happening on the field. The styles that different directors employ are extremely consistent within a sport and make up a “language of film” which the machine can take advantage of in order to represent the events taking place in the video.

Thus, even though it is not easy to automatically identify a player hitting a ball in video, it is easy to 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). Although these correlations are not perfect, experiments have shown that baseball events can be classified using such features (Fleischman et al., 2007).

We exploit the language of film to represent events in sports video in two phases. First, low level features that correlate with basic events in sports are extracted from the video stream. Then, temporal data mining is used to find patterns within this low level event stream.

2.1 Feature Extraction

We extract three types of features: visual context features, camera motion features, and audio context features.

Visual Context Features

Visual context features encode general properties of the visual scene in a video segment. Supervised classifiers are trained to identify these features, which are relatively simple to classify in comparison to high level events (like home runs) that require more training data and achieve lower accuracy. The first step in classifying visual context features is to segment the video into shots (or scenes) based on changes in the visual scene due to editing (e.g. jumping from a close up to a wide shot of the field). Shot detection and segmentation is a well studied problem; in this work we use the method of Tardini et al. (2005).

After the video is segmented into shots, individual frames (called key frames) are selected and represented as a vector of low level features that describe the key frame's color distribution, entropy, etc. (see Fleischman and Roy, 2007 for the full list of low level features used). The WEKA machine learning package is used to train a boosted decision tree to classify these frames into one of three categories: *pitching-scene, field-scene, other* (Witten and Frank, 2005). Those shots whose key frames are classified as *field-scenes* are then subcategorized (using boosted decision trees) into one of the following categories: *infield, outfield, wall, base, running, and misc.* Performance of these classification tasks is approximately 96% and 90% accuracy respectively.

Camera Motion Features

In addition to visual context features, we also examine the camera motion that occurs within a video. Unlike visual context features, which provide information about the global situation that is being observed, camera motion features represent more precise information about the actions occurring in a video. The intuition here is that the camera is a stand in for a viewer's focus of attention. As actions occur in a video, the camera moves to follow it; this camera motion thus mirrors the actions themselves, providing informative features for event representation.

Like shot boundary detection, detecting the motion of the camera in a video (i.e., the amount it pans left to right, tilts up and down, and zooms in and out) is a well-studied problem. We use the system of Bouthemy et al. (1999) which computes the camera motion using the parameters of a two-dimensional affine model to fit every pair of sequential frames in a video. A 15 state 1st order Hidden Markov Model, implemented with the Graphical Modeling Toolkit,[2] then converts the output of the Bouthemy system into a stream of clustered characteristic camera motions (e.g. state 12 clusters together motions of zooming in fast while panning slightly left).

Audio Context

The audio stream of a video can also provide useful information for representing non-linguistic context. We use boosted decision trees to classify audio into segments of *speech, excited_speech, cheering*, and *music*. Classification operates on a sequence of overlapping 30 ms frames extracted 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

Given a set of low level features that correlate with the basic events in sports, we can now focus on building up representations of complex events. Unlike previous work (Hongen et al., 2005) in which representations of the temporal relations between low level events are built up by hand, we employ temporal data mining techniques to automatically discover such relations from a large corpus of unannotated video.

As described above, ideal basic events (such as hitting and catching) cannot be identified easily in sports video. By finding temporal patterns between audio, visual and camera motion features, however, we 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

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

in complex and varied temporal relations to each other.

To find such patterns automatically, we follow previous work in video content classification in which temporal data mining techniques are used to discover event patterns within streams of lower level features. The algorithm we use is fully unsupervised and proceeds by examining the relations that occur between features in multiple streams within a moving time window. Any two features that occur within this window must be in one of seven temporal relations with each other (e.g. *before, during, etc.*) (Allen, 1984). 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. [OVERLAP, *field-scene, cheer*]), 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. [BEFORE, [OVERLAP, *field-scene, cheer*], *field scene*]]).

The temporal patterns found significant in this way make up a codebook which can then be used as a basis for representing a video. The term codebook is often used in image analysis to describe a set of features (stored in the codebook) that are used to encode raw data (images or video). Such codebooks are used to represent raw video using features that are more easily processed by the computer.

Our framework follows a similar approach in which raw video is encoded (using a codebook of temporal patterns) as follows. First, the raw video is abstracted into the visual context, camera motion, and audio context feature streams (as described in Section 2.1). These feature streams are then scanned, looking for any temporal patterns (and nested sub-patterns) that match those found in the codebook. For each pattern, the duration for which it occurs in the feature streams is treated as the value of an element in the vector representation for that video.

Thus, a video is represented as an n length vector, where n is the total number of temporal patterns in the codebook. The value of each element of this vector is the duration for which the pattern associated with that element was observed in the video. So, if a pattern was not observed in a video at all, it would have a value of 0, while if it was observed for the entire length of the video, it would have a value equal to the number of frames present in that video.

Given this method for representing the non-linguistic context of a video, we can now examine how to model the relationship between such context and the words used to describe it.

3 Linguistic Mapping

Modeling the relationship between words and non-linguistic context assumes that the speech uttered in a video refers consistently (although not exclusively) to the events being represented by the temporal pattern features. We model this relationship, much like traditional language models, using conditional probability distributions. Unlike traditional language models, however, our grounded language models condition the probability of a word not only on the word(s) uttered before it, but also on the temporal pattern features that describe the non-linguistic context in which it was uttered. We estimate these conditional distributions using a framework similar that used for training acoustic models in ASR and translation models in Machine Translation (MT).

We generate a training corpus of utterances paired with representations of the non-linguistic context in which they were uttered. The first step in generating this corpus is to generate the low level features described in Section 2.1 for each video in our training set. We then segment each video into a set of independent events based on the visual context features we have extracted. We follow previous work in sports video processing (Gong et al., 2004) and define an event in a baseball video as any sequence of shots starting with a *pitching-scene* and continuing for four subsequent shots. This definition follows from the fact that the vast majority of events in baseball start with a pitch and do not last longer than four shots. For each of these events in our corpus, a temporal pattern feature vector is generated as described in section 2.2. These events are then paired with all the words from the closed captioning transcription that occur during each event (plus or minus 10 seconds). Because these transcriptions are not necessarily time synched with the audio, we use the method described in Hauptmann and Witbrock

(1998) to align the closed captioning to the announcers' speech.

Previous work has examined applying models often used in MT to the paired corpus described above (Fleischman and Roy, 2006). Recent work in automatic image annotation (Barnard et al., 2003; Blei and Jordan, 2003) and natural language processing (Steyvers et al., 2004), however, have demonstrated the advantages of using hierarchical Bayesian models for related tasks. In this work we follow closely the Author-Topic (AT) model (Steyvers et al., 2004) which is a generalization of Latent Dirichlet Allocation (LDA) (Blei et al., 2005).[3]

LDA is a technique that was developed to model the distribution of topics discussed in a large corpus of documents. The model assumes that every document is made up of a mixture of topics, and that each word in a 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 (or phrase) in a document, an author is chosen uniformly from the set of the authors of the document. Then, a topic is chosen from a distribution of topics associated with that particular author. Finally, the word is generated from the distribution associated with that chosen topic. We can express the probability of the words in a document (W) given its authors (A) as:

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

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

We use the AT model to estimate our grounded language model by making an analogy between documents and events in video. In our framework, the words in a document correspond to the words in the closed captioning transcript associated with an event. The authors of a document correspond to the temporal patterns representing the non-linguistic context of that event. We modify the AT model slightly, such that, instead of selecting from a uniform distribution (as is done with authors of documents), we select patterns from a multinomial distribution based upon the duration of the pattern. The intuition here is that patterns that occur for a longer duration are more salient and thus, should be given greater weight in the generative process. We can now rewrite (1) to give the probability of words during an event (W) given the vector of observed temporal patterns (P) as:

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

In the experiments described below we follow Steyver et al., (2004) and train our AT model using Gibbs sampling, a Markov Chain Monte Carlo technique for obtaining parameter estimates. We run the sampler on a single chain for 200 iterations. We set the number of topics to 15, 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 Evaluation

In order to evaluate our grounded language modeling approach, a parallel data set of 99 Major League Baseball games with corresponding closed captioning transcripts was recorded from live television. These games represent data totaling approximately 275 hours and 20,000 distinct events 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, baseball highlights (i.e., events which terminate with the player either *out* or *safe*) were hand annotated for use in evaluation, and manually transcribed in order to get clean text transcriptions for gold standard comparisons. Of the 1200 events in the test set, 237 were highlights with a total word count of 12,626 (vocabulary of 1800 words).

The remaining 93 unlabeled games are used to train unigram, bigram, and trigram grounded language models. Only unigrams, bigrams, and trigrams that are not proper names, appear greater than three times, and are not composed only of stop words were used. These grounded language models are then combined in a backoff strategy

[3] In the discussion that follows, we describe a method for estimating unigram grounded language models. Estimating bigram and trigram models can be done by processing on word pairs or triples, and performing normalization on the resulting conditional distributions.

with traditional unigram, bigram, and trigram language models generated from a combination of the closed captioning transcripts of all training games and data from the switchboard corpus (see below). This backoff is necessary to account for the words not included in the grounded language model itself (i.e. stop words, proper names, low frequency words). The traditional text-only language models (which are also used below as baseline comparisons) are generated with the SRI language modeling toolkit (Stolcke, 2002) using Chen and Goodman's modified Kneser-Ney discounting and interpolation (Chen and Goodman, 1998). The backoff strategy we employ here is very simple: if the ngram appears in the GLM then it is used, otherwise the traditional LM is used. In future work we will examine more complex backoff strategies (Hsu, in review).

We evaluate our grounded language modeling approach using 3 metrics: perplexity, word error rate, and precision on an information retrieval task.

4.1 Perplexity

Perplexity is an information theoretic measure of how well a model predicts a held out test set. We use perplexity to compare our grounded language model to two baseline language models: a language model generated from the switchboard corpus, a commonly used corpus of spontaneous speech in the telephony domain (3.65M words; 27k vocab); and a language model that interpolates (with equal weight given to both) between the switchboard model and a language model trained only on the baseball-domain closed captioning (1.65M words; 17k vocab). The results of calculating perplexity on the test set highlights for these three models is presented in Table 1 (lower is better).

Not surprisingly, the switchboard language model performs far worse than both the interpolated text baseline and the grounded language model. This is due to the large discrepancy between both the style and vocabulary of language about sports compared to the domain of telephony sampled by the switchboard corpus. Of more interest is the decrease in perplexity seen when using the grounded language model compared to the interpolated model. Note that these two language models are generated using the same speech transcriptions, i.e. the closed captioning from the training games and the switchboard corpus. However, whereas the baseline model remains the same for each of the 237 test highlights, the grounded language model generates different word distributions for each highlight depending on the event features extracted from the highlight video.

	Switchboard	Interpolated (Switch+CC)	Grounded
ppl	1404	145.27	83.88

Table 1. Perplexity measures for three different language models on a held out test set of baseball highlights (12,626 words). We compare the grounded language model to two text based language models: one trained on the switchboard corpus alone; and interpolated with one trained on closed captioning transcriptions of baseball video.

4.2 Word Accuracy and Error Rate

Word error rate (WER) is a normalized measure of the number of word insertions, substitutions, and deletions required to transform the output transcription of an ASR system to a human generated gold standard transcription of the same utterance. Word accuracy is simply the number of words in the gold standard that they system correctly recognized. Unlike perplexity which only evaluates the performance of language models, examining word accuracy and error rate requires running an entire ASR system, i.e. both the language and acoustic models.

We use the Sphinx system to train baseball specific acoustic models using parallel acoustic/text data automatically mined from our training set. Following Jang and Hauptman (1999), we use an off the shelf acoustic model (the hub4 model) to generate an extremely noisy speech transcript of each game in our training set, and use dynamic programming to align these noisy outputs to the closed captioning stream for those same games. Given these two transcriptions, we then generate a paired acoustic/text corpus by sampling the audio at the time codes where the ASR transcription matches the closed captioning transcription.

For example, if the ASR output contains the term sequence "… and farther *home run for David* forty says…" and the closed captioning contains the sequence "…another *home run for David* Ortiz…," the matched phrase *"home run for David"* is assumed a correct transcription for the audio at the time codes given by the ASR system. Only looking at sequences of three words or more,

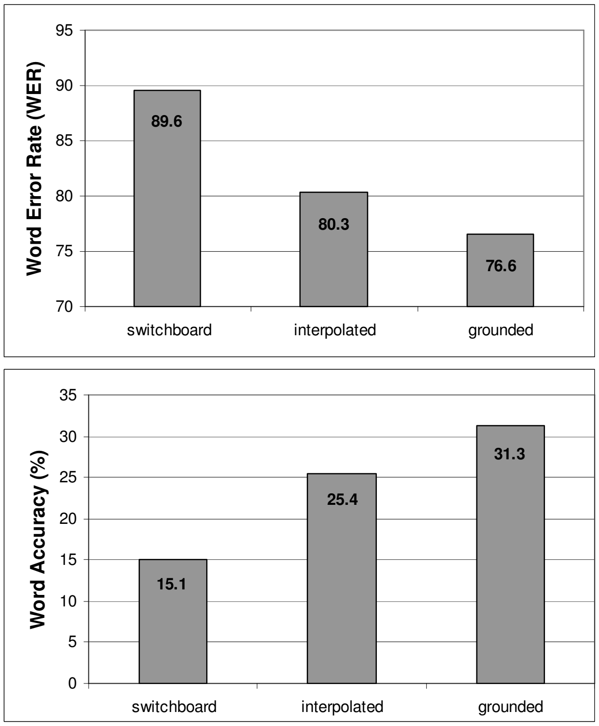

Figure 3. Word accuracy and error rates for ASR systems using a grounded language model, a text based language model trained on the switchboard corpus, and the switchboard model interpolated with a text based model trained on baseball closed captions.

we extract approximately 18 hours of clean paired data from our 275 hour training corpus. A continuous acoustic model with 8 gaussians and 6000 ties states is trained on this data using the Sphinx speech recognizer.[4]

Figure 3 shows the WERs and accuracy for three ASR systems run using the Sphinx decoder with the acoustic model described above and either the grounded language model or the two baseline models described in section 4.1. Note that performance for all of these systems is very poor due to limited acoustic data and the large amount of background crowd noise present in sports video (and particularly in sports highlights). Even with this noise, however, results indicate that the word accuracy and error rates when using the grounded language model is significantly better than both the switchboard model (absolute WER reduction of 13%; absolute accuracy increase of 15.2%) and the switchboard interpolated with the baseball specific text based language model (absolute WER reduction of 3.7%; absolute accuracy increase of 5.9%).

Drawing conclusions about the usefulness of grounded language models using word accuracy or error rate alone is difficult. As it is defined, these measures penalizes a system that mistakes “a” for “uh” as much as one that mistakes “run” for “rum.” When using ASR to support multimedia applications (such as search), though, such substitutions are not of equal importance. Further, while visual information may be useful for distinguishing the latter error, it is unlikely to assist with the former. Thus, in the next section we examine an extrinsic evaluation in which grounded language models are judged not directly on their effect on word accuracy or error rate, but based on their ability to support video information retrieval.

4.3 Precision of Information Retrieval

One of the most commonly used applications of ASR for video is to support information retrieval (IR). Such video IR systems often use speech transcriptions to index segments of video in much the same way that words are used to index text documents (Wactlar et al., 1996). For example, in the domain of baseball, if a video IR system were issued the query “home run,” it would typically return a set of video clips by searching its database for events in which someone uttered the phrase “home run.” Because such systems rely on ASR output to search video, the performance of a video IR system gives an indirect evaluation of the ASR’s quality. Further, unlike the case with word accuracy or error rate, such evaluations highlight a systems ability to recognize the more relevant content words without being distracted by the more common stop words.

Our metric for evaluation is the precision with which baseball highlights are returned in a video IR system. We examine three systems: one that uses ASR with the grounded language model, a baseline system that uses ASR with the text only interpolated language model, and finally a system that uses human produced closed caption transcriptions to index events.

For each system, all 1200 events from the test set (not just the highlights) are indexed. Queries are generated artificially using a method similar to Berger and Lafferty (1999) and used in Fleischman and Roy (2007). First, each highlight is labeled with the event’s type (e.g. *fly ball*), the event’s location (e.g. *left field*) and the event’s result (e.g. *double play*): 13 labels total. Log likelihood ratios

[4] http://cmusphinx.sourceforge.net/html/cmusphinx.php

are then used to find the phrases (unigram, trigram, and bigram) most indicative of each label (e.g. "fly ball" for category *fly ball*). For each label, the three most indicative phrases are issued as queries to the system, which ranks its results using the language modeling approach of Ponte and Croft (1998). Precision is measured on how many of the top five returned events are of the correct category.

Figure 4 shows the precision of the video IR systems based on ASR with the grounded language model, ASR with the text-only interpolated language model, and closed captioning transcriptions. As with our previous evaluations, the IR results show that the system using ASR with the grounded language model performed better than the one using ASR with the text-only language model (5.1% absolute improvement). More notably, though, Figure 4 shows that the system using the grounded language model performed better than the system using the hand generated closed captioning transcriptions (4.6% absolute improvement). Although this is somewhat counterintuitive given that hand transcriptions are typically considered gold standards, these results follow from a limitation of using text-based methods to index video.

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.

By taking into account non-linguistic context during speech recognition, the grounded language model system indirectly circumvents some of these false positive results. This follows from the fact that an effect of using the grounded language model is that when an announcer utters a phrase (e.g., "fly ball"), the system is more likely to recognize that phrase correctly if the event it refers to is actually occurring (e.g. if someone actually hit a fly ball). Because the grounded language model system is biased to recognize phrases that describe what is currently happening, it returns fewer false positives and gets higher precision.

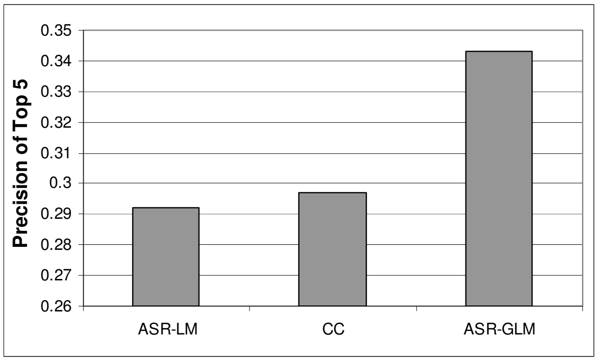

Figure 4. Precision of top five results of a video IR system based on speech transcriptions. Three different transcriptions are compared: ASR-LM uses ASR with a text-only interpolated language model (trained on baseball closed captioning and the switchboard corpus); ASR-GLM uses ASR with a grounded language model; CC uses human generated closed captioning transcriptions (i.e., no ASR).

5 Conclusions

We have described a method for improving speech recognition in video. The method uses grounded language modeling, an extension of tradition language modeling in which the probability of a word is conditioned not only on the previous word(s) but also on the non-linguistic context in which the word is uttered. Context is represented using hierarchical temporal patterns of low level features which are mined automatically from a large unlabeled video corpus. Hierarchical Bayesian models are then used to map these representations to words. Initial results show grounded language models improve performance on measures of perplexity, word accuracy and error rate, and precision on an information retrieval task.

In future work, we will examine the ability of grounded language models to improve performance for other natural language tasks that exploit text based language models, such as Machine Translation. Also, we are examining extending this approach to other sports domains such as American football. In theory, however, our approach is applicable to any domain in which there is discussion of the here-and-now (e.g., cooking shows, etc.). In future work, we will examine the strengths and limitations of grounded language modeling in these domains.

References

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

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

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.

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

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).

Chen, S. F. and Goodman, J., (1998). An Empirical Study of Smoothing Techniques for Language Modeling, Tech. Report TR-10-98, Computer Science Group, Harvard U., Cambridge, MA.

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

Fleischman, M. and Roy, D. (2007). Unsupervised Content-Based Indexing of Sports Video Retrieval. *9th ACM Workshop on Multimedia Information Retrieval (MIR).* Augsburg, Germany.

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.

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

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

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).

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

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

Hsu , Bo-June (Paul). (in review). Generalized Linear Interpolation of Language Models.

Jang, P., Hauptmann, A. (1999). Learning to Recognize Speech by Watching Television. *IEEE Intelligent Systems Magazine*, 14(5), pp. 51-58.

Mukherjee, N. and Roy, D.. (2003). A Visual Context-Aware Multimodal System for Spoken Language Processing. *Proc. Eurospeech*, 4 pages.

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

Roy, D. (2005). . Grounding Words in Perception and Action: Insights from Computational Models. TICS.

Roy, D. and Pentland, A. (2002). Learning Words from Sights and Sounds: A Computational Model. *Cognitive Science*, 26(1).

Roy. D. and Reiter, E. (2005). . Connecting Language to the World. Artificial Intelligence, 167(1-2), 1-12.

Snoek, C.G.M. and Worring, M.. (2005). Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5-35.

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.

Stolcke, A., (2002). SRILM - An Extensible Language Modeling Toolkit, in Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado.

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.

Wactlar, H., Witbrock, M., Hauptmann, A., (1996). Informedia: News-on-Demand Experiments in Speech Recognition. ARPA Speech Recognition Workshop, Arden House, Harriman, NY.

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