Extracting Latent Attributes from Video Scenes Using Text as Background Knowledge

Anh Tran, Mihai Surdeanu, Paul Cohen
University of Arizona
{trananh, msurdeanu, prcohen}@email.arizona.edu

Abstract

We explore the novel task of identifying latent attributes in video scenes, such as the mental states of actors, using only large text collections as background knowledge and minimal information about the videos, such as activity and actor types. We formalize the task and a measure of merit that accounts for the semantic relatedness of mental state terms. We develop and test several largely unsupervised information extraction models that identify the mental states of human participants in video scenes. We show that these models produce complementary information and their combination significantly outperforms the individual models as well as other baseline methods.

1 Introduction

"Labeling a narrowly avoided vehicular manslaughter as *approach(car, person)* is missing something."[1] The recognition of activities, participants, and objects in videos has advanced considerably in recent years (Li et al., 2010; Poppe, 2010; Weinland et al., 2011; Yang and Ramanan, 2011; Ng et al., 2012). However, identifying latent attributes of scenes, such as the mental states of human participants, has not been addressed. Latent attributes matter: If a video surveillance system detects one person chasing another, the response from law enforcement should be radically different if the people are happy (e.g., children playing) or afraid and angry (e.g., a person running from an assailant).

[1] James Donlon, former manager of DARPA's Mind's Eye program, personal communication.

Attributes that are latent in visual representations are often explicit in textual representations. This suggests a novel method for inferring latent attributes: Use explicit features of videos to query text corpora, and from the resulting texts extract attributes that are latent in the videos, such as mental states. The contributions of this work are:

1: We formalize the novel task of latent attribute identification from video scenes, focusing on the identification of actors' mental states. The input for the task is contextual information about the scene, such as detections about the activity (e.g., chase) and actor types (e.g., policeman or child), and the output is a distribution over mental state labels. We show that gold standard annotations for this task can be reliably generated using crowd sourcing. We define a novel evaluation measure, called *constrained weighted similarity-aligned* F_1 score, that accounts for both the differences between mental state distributions and the semantic relatedness of mental state terms (e.g., partial credit is given for *irate* when the target is *angry*).

2: We propose several robust and largely unsupervised information extraction (IE) models for identifying the mental state labels of human participants in a scene, given solely the activity and actor types: a lexical semantic (LS) model that extracts mental state labels that are highly similar to the context of the scene in a latent, conceptual vector space; and an information retrieval (IR) model that identifies labels commonly appearing in sentences related to the explicit scene context. We show that these models are complementary and their combination performs better than either model, alone.

3: Furthermore, we show that an event-centric model that focuses on the mental state labels of the participants in the relevant event (identified using syntactic patterns and coreference resolution) outperforms the above shallower models.

2 Related Work

As far as we know, the task proposed here is novel. We can, however, review work relevant to each part of the problem and our solution. Mental state inference is often formulated as a classification problem, where the goal is to predict target mental state labels based on low-level sensory input data. Most solutions try to learn classification models based on large amounts of training data, while some require human engineering of domain knowledge. Hidden Markov Models (HMMs) and Dynamic Bayesian Networks (DBNs) are popular representations because they can model the temporal evolution of mental states. For instance, the mental states of students can be inferred from unintentional body gestures using a DBN (Abbasi et al., 2009). Likewise, an HMM can also be used to model the emotional states of humans (Liu and Wang, 2011). Some solutions combine HMMs and DBNs in a Bayesian inference framework to yield a multi-layer representation that can do real-time inference of complex mental and emotional states (El Kaliouby and Robinson, 2004; Baltrusaitis et al., 2011). Our work differs from these approaches in several ways: It is mostly unsupervised, multi-modal, and requires little training.

Relevant video processing technology includes object detection (e.g., (Felzenszwalb et al., 2008)), person detection, and pose detection (e.g., (Yang and Ramanan, 2011)). Many tracking algorithms have been developed, such as group tracking (McKenna et al., 2000), tracking by learning appearances (Ramanan et al., 2007), and tracking in 3D space (Giebel et al., 2004; Brau et al., 2013). For human action recognition, current state-of-the-art techniques are capable of achieving near perfect performance on the commonly used KTH Actions dataset (Schuldt et al., 2004) and high performance rates on other more challenging datasets (O'Hara and Draper, 2012; Sadanand and Corso, 2012).

To extract mental state information from texts, one might use any or all of the technologies of natural language processing, so a complete review of relevant technologies is impossible, here. Of immediate relevance is the work of de Marneffe et al. (2010), which identified the latent meaning behind scalar adjectives (e.g., which ages people have in mind when talking about "little kids"). The authors learned these meanings by extracting scalars, such as children's ages, that were commonly collocated with phrases, such as "little kids," in web documents. Mohtarami et al. (2011) tried to infer yes/no answers from indirect yes/no question-answer pairs (IQAPs) by predicting the uncertainty of sentiment adjectives in indirect answers. Their method employs antonyms, synonyms, word sense disambiguation as well as the semantic association between the sentiment adjectives that appear in the IQAP to assign a degree of certainty to each answer. Sokolova and Lapalme (2011) further showed how to learn a model for predicting the opinions of users based on their written contents, such as reviews and product descriptions, on the Web. Gabbard et al. (2011) found that coreference resolution can significantly improve the recall rate of relations extraction without much expense to the precision rate.

Our work builds on these efforts by combining information retrieval, lexical semantics, and event extraction to extract latent scene attributes.

3 Data

For the experiments in this paper, we focus solely on videos containing chase scenes. Chases often invoke clear mental state inferences, and depending on context can suggest very different mental state distributions for the actors involved.

3.1 Video Corpus

We compiled a video dataset of 26 chase videos found on the Web. Of these, five involve police officers, seven involve children, four show sports-related scenes, and twelve describe different chase scenarios involving civilian adults (two videos involve children playing sports). The average duration of the dataset is 8.8 seconds with a range of $[4, 18]$. Most videos involve a single chaser and a single chasee (a person being chased) while a few have several chasers and/or chasees.

For each video, we used Amazon Mechanical Turk (MTurk) to identify both the actors and their mental states. Each worker was asked to view a video in its entirety before answering some questions about the scene. We give no prior training to the workers. The questions were carefully phrased to apply to all participants of a particular role, for example all chasers (if there are more than one). We also ask obvious validation questions about the participants in each role (e.g., are the chasers running towards the camera?) and use the answers to these questions to filter out poor responses. In gen-

eral, we found that most responses were good and only a few incomplete submissions were rejected.

In the first experiment, we asked MTurk workers to select the actor types and various other detections from a predefined list of tags. This labeling task is a proxy for a computer vision detection system that functions at a human level of performance. Indeed, we restricted the actor type labels to a set that can be reasonably expected from automatic detection algorithms: *person*, *police officer*, *child*, and (non-human) *object*. For instance, police officers often wear distinctive color uniforms that can be learned using the Felzenszwalb detector (Felzenszwalb et al., 2008), whereas children can be reliably differentiated by their heights under a 3D-tracking model (Brau et al., 2013). Each video was annotated by three different workers and the union of their annotations is produced. The overall accuracy of the annotation was excellent. The MTurk workers correctly identified the important actors in every video.

Next, we collected a gold standard list of mental state labels for each video by asking MTurk workers to identify *all* applicable mental state adjectives for the actors involved. We used a text-box to allow for free-form input. Studies have shown that people of different cultures can perceive emotions very differently, and having forced choice options cannot always capture their true perception (Gendron et al., 2014). Therefore, we did not restrict the response of the workers in any way. Workers could abstain from answering if they felt the video was too ambiguous. Each video was evaluated by ten different workers. We converted each term provided to the closest adjective form if possible. Terms with no equivalent adjective forms were left in place. On rare occasions, workers provided sentence descriptions despite being asked for single-word adjectives. These sentences were either removed, or collapsed into a single word if appropriate. The overall quality of the annotations was good and generally followed common intuition. Asides from the frequently used terms, we also received some colorful (yet informative) descriptions, like *incredulous* and *vindictive*. In general, chases involving police scenarios often contained violent and angry states while chases involving children received more cheerful labels. There were unexpected descriptions, such as *annoy* for a playful chase between two children. Upon review of the video, we agreed that one child did indeed look annoyed. Thus, the resulting descriptions were subjective, but very few were hard to rationalize. By aggregating the answers from the workers, we generated a gold standard distribution of mental state terms for each video.[2]

3.2 Text Corpus

The text corpus used for our models is the English Gigaword 5th Edition corpus[3], made available by the Linguistics Data Consortium and indexed by Lucene[4]. It is a comprehensive archive of newswire text data (approximately 26 GB), acquired over several years. It is in this corpus that we expect to find mental state terms cued by contextual information from videos.

4 Neighborhood Models

We developed several individual models based on the *neighborhood paradigm*, that is, the hypothesis that relevant mental state labels will appear "near" text cued by the visual features of a scene.

The models take as input the *context* extracted from a video scene, defined simply as a list of "activity and actor-type" tuples (e.g., (*chase*, *police*)). Multiple actor types will result in multiple tuples for a video. The actors can be either a person, a policeman, a child, or a (non-human) object. If the detections describe the actor as both a person and a child, or a person and a policeman, we automatically remove the *person* label as it is a WordNet (Miller, 1995) hypernym of both *child* and *policeman*. For each human actor type, we further increase our coverage by retrieving the synonym set (synset) of its most frequent sense (i.e., sense #1) from WordNet. For example, a chase involving a policeman would generate the following tuples: (*chase*, *policeman*) and (*chase*, *officer*).

We call these *query tuples* because they are used to query text for sentences that – if all goes well – will contain relevant mental state labels.

Given query tuples, our models use an initial seed set of 160 mental state adjectives to produce a single distribution over mental state labels, referred to as the *response distribution*, for each video. The seed set is compiled from popular mental and emotional state dictionaries, including the Profile of Mood States (POMS) (McNair et al., 1971) and Plutchik's wheel of emotion. We

[2]All videos and annotations are available at: `http://trananh.github.io/vlsa`

[3]Linguistics Data Consortium catalog no. LDC2011T07

[4]Apache Lucene: `http://lucene.apache.org`

Source	Example Mental State Labels
POMS	alert, annoyed, energetic, exhausted, helpful, sad, terrified, unworthy, weary, etc.
Plutchik	angry, disgusted, fearful, joyful/joyous, sad, surprised, trusting, etc.
Others	agitated, competitive, cynical, disappointed, excited, giddy, happy, inebriated, violent, etc.

Table 1: The initial seed set contains 160 mental state labels, compiled from different sources like the popular Profile of Mood States dictionary and Plutchik's wheel of emotion.

also included frequently used labels gathered from synsets found in WordNet (see Table 1 for examples). Note that the gold standard annotations produced by MTurk workers (Sec. 3) was not a source for this set, nor was it restricted to these terms.

4.1 Back-off Interpolation in Vector Space

Our first model uses the recurrent neural network language model (RNNLM) of Mikolov et al. (2013) to project both mental state labels and query tuples into a latent conceptual space. Similarity is then trivially computed as the cosine similarity between these vectors. In all of our experiments, we used a RNNLM computed over the Gigaword corpus with 600-dimensional vectors.

For this vector space (*vec*) model, we separate the query tuples into different levels of back-off context. The first level includes the set of activity types as singleton context tuples, e.g., (*chase*), while the second level includes all (*activity*, *actor*) context tuples. Hence, each query tuple will yield two different context tuples, one for each back-off level. For each context tuple with multiple terms, such as (*chase*, *policeman*), we find the vector representation for the context by aggregating the vectors representing the search terms:

$$vec(\textit{chase}, \textit{policeman}) = vec(\textit{chase}) + vec(\textit{policeman}) \; .$$

The vector representation for a singleton context tuple is just the vector of the single search term. We then calculate the distance of each mental state label m to the normalized vector representation of the context tuple by computing the cosine similarity score between the two vectors:

$$cos(\Theta_m) = \frac{vec(m) \cdot vec(\textit{context tuple})}{||vec(m)|| \; ||vec(\textit{context tuple})||} \; .$$

The hypothesis here is that mental state labels that are related to the search context will have a RNNLM vector that is closer to the context tuple vector, resulting in a high cosine similarity score. Because the number of latent dimensions is relatively small (when compared to vocabulary size), cosine similarity scores in this latent space tend to be close. To further separate these scores, we raise them to an exponential power:

$$score(m) = e^{cos(\Theta_m)+1} - 1 \; .$$

The processing of each context tuple yields 160 different scores, one for each mental state label. We normalize these scores to form a single distribution of scores for each context tuple. The distributions are then integrated into a single distribution representative of the complete activity as follows: (a) the distributions at each context back-off level are averaged to generate a single distribution per level – for the second level (which includes activity and actor types), it means distributions for all (*activity*, *actor*) tuples are averaged, whereas the first level only has a single distribution from the singleton activity tuple (*chase*); and (b) distributions for the different levels are linearly interpolated, similar to the back-off strategy of (Collins, 1997). Let e_1 and e_2 represent the weights of some mental state label m from the average distribution at the first and second level, respectively. Then the interpolated distribution score e for m is:

$$e = \lambda e_1 + (1 - \lambda) e_2 \; .$$

Compiling the distribution scores for each m produces the final distribution representing the activity modeled. We prune this final distribution by taking the top ranked items that make up some γ proportion of the distribution. We delay the discussion of how γ is tuned to Section 6. The final pruned distribution is normalized to produce the response distribution.

4.2 Sentence Co-occurrence with Deleted Interpolation

Our second model, the *sent* model, extracts mental state labels based on the likelihood that they appear in sentences cued by query tuples. For each tuple, we estimate the conditional probability that we will see a mental state label m in a sentence, where m is from the seed set, given that we already observed the desired activity and actor type in the same sentence: $P(m|\textit{activity}, \textit{actor})$. In this case, we refer to the sentence length as the neighborhood window. Furthermore, all terms must appear as the correct part-of-speech (POS): m must

appear as an adjective or verb, the activity as a verb, and the actor as a noun. (Mental state adjectives are allowed to appear as verbs because some are often mis-tagged as verbs; e.g., agitated, determined, welcoming.) We used Stanford's CoreNLP toolkit for tokenization and POS tagging.[5]

Note that this probability is similar to a trigram probability in POS tagging, except the triples need not form an ordered sequence but must appear in the same sentence and under the correct POS tag. Unfortunately, we cannot always compute this trigram probability directly from the corpus because there might be too few instances of each trigram to compute a probability reliably. As is common, we instead estimate it as a linear interpolation of unigrams, bigrams, and trigrams. We define the maximum likelihood probabilities $\hat{P}$, derived from relative frequencies f, for the unigrams, bigrams, and trigrams as follows:

$$\hat{P}(m) = \frac{f(m)}{N}$$

$$\hat{P}(m|\textit{activity}) = \frac{f(m, \textit{activity})}{f(\textit{activity})}$$

$$\hat{P}(m|\textit{activity}, \textit{actor}) = \frac{f(m, \textit{activity}, \textit{actor})}{f(\textit{activity}, \textit{actor})}$$

for all mental state labels m, activities, and actor types in our queries. N is the total number of tokens in the corpus. The aforementioned POS requirement is enforced: $f(m)$ is the number of occurrences of m as an adjective or verb. We define $\hat{P} = 0$ if the corresponding numerator and denominator are zero. The desired trigram probability is then estimated as:

$$P(m|\textit{activity}, \textit{actor}) = \lambda_1 \hat{P}(m) + \lambda_2 \hat{P}(m|\textit{activity}) + \lambda_3 \hat{P}(m|\textit{activity}, \textit{actor}) .$$

As $\lambda_1 + \lambda_2 + \lambda_3 = 1$, P represents a probability distribution. We use the deleted interpolation algorithm (Brants, 2000) to estimate one set of lambda values for the model, based on all trigrams.

For each query tuple generated in a video, 160 different trigrams are computed, one for each mental state label in the seed set, resulting in 160 conditional probability scores. We normalize these scores into a single distribution – the mental state distribution for that query tuple. We then combine all resulting distributions, one from each query tuple, and take the average to produce a single distribution over mental state labels for the video. As before, we prune this distribution by taking the top-ranked items that cover a large fraction γ of total probability. The pruned distribution is renormalized to yield the final response distribution.

[5] http://nlp.stanford.edu/software/corenlp.shtml.

4.3 Event-centric with Deleted Interpolation

The *sent* model has two limitations. On one hand, it is too sparse: the single sentence neighborhood window is too small to reliably estimate the frequencies of trigrams for the probabilities of mental state terms. On the other hand, it may be too lenient, as it extracts all mental state mentions appearing in the same sentence with the activity, or event, under consideration, regardless if they apply to this event or not. We address these limitations next with an event-centric model (*event*).

Intuitively, the *event* model focuses on the mental state labels of event participants. Formally, these mental state terms are extracted as follows:

1: We identify event participants (or actors). We do this by analyzing the syntactic dependencies of sentences containing the target verb (e.g., *chase*) to find the subject and object. In most cases, the nominal subject of the verb *chase* is the chaser and the direct object is the person being chased. We implemented additional patterns to model passive voice and other exceptions. We used Stanford's CoreNLP toolkit for syntactic dependency parsing and the downstream coreference resolution.

2: Once the phrases that point to actors are identified, we identify all mentions of these actors in the entire document by traversing the coreference chains containing the phrases extracted in the previous step. The sentences traversed in the chains define the neighborhood area for this model.

3: Lastly, we identify the mental state terms of event participants using a second set of syntactic patterns. First, we inspect several copulative verbs, such as to *be* and *feel*, and extract mental state labels from these structures if the corresponding subject is one of the mentions detected above. Second, we search for mental states along adjectival modifier relations, where the head is an actor mention. For all patterns, we make sure to filter for only mental state complements belonging to the initial seed list. The same POS restriction as in the other models also applies. We increment the joint frequency f for the n-gram once for

each neighborhood that properly contain all search terms from the n-gram in the correct POS.

The *event* model addresses both limitations of the *sent* model: it avoids the lenient extraction of mental state labels by focusing on labels associated with event participants; it addresses sparsity by considering all mentions of event participants in a document.

To understand the impact of this model, we compare it against two additional baselines. The first baseline investigates the importance of focusing on mental state terms associated with event participants. This model, called *coref*, implements the first two steps of the above algorithm, but instead of extracting only mental state terms associated with event actors (last step), it considers all mentions appearing anywhere in the coreference neighborhood. That is, all unique sentences traversed by the relevant coreference chains are first pieced together to define a single neighborhood for a given document; then the relative joint frequencies of n-grams are computed by incrementing f once for each neighborhood that contains all terms with correct POS tags.

The second baseline analyzes the importance of coreference resolution to our problem. This model is similar to *sent*, with the modification that it increases the size of the neighborhood window to include the immediate neighbors of target sentences that contain activity labels. We call this the *win-n* model: The window around a target verb contains $2n + 1$ sentences. We build the context neighborhood by concatenating all target sentences and their windows together for a given document. This defines a single neighborhood for each document. This contrasts with the *sent* model, in which the neighborhood is defined for each sentence containing the activity label in the document, resulting in several possible neighborhoods in a document. The joint frequency f for each n-gram – where $n > 1$ – is computed similarly with the *coref* model: it is incremented once for each neighborhood that contains all the terms from the n-gram in the correct POS. Frequencies for unigrams are computed similar to *sent*.

As before, 160 different trigrams are generated for each query tuple, one for each mental state label in the seed set, resulting in 160 conditional probability scores. We similarly combine these scores and generate a single pruned distribution as the response for each of the model above.

G	$(\text{irate}, 0.8), (\text{afraid}, 0.2)$
R_1	$(\text{angry}, 0.6), (\text{mad}, 0.4)$
R_2	$(\text{irate}, 0.2), (\text{afraid}, 0.8)$
R_3	$(\text{mad}, 0.4), (\text{irate}, 0.4), (\text{scared}, 0.2)$

Table 2: We show an example gold standard distribution G and several candidate response distributions to be matched against G. Here, R_3 best matches the shape and meaning of G, because (*irate*, *mad*) and (*afraid*, *scared*) are close synonyms. R_2 appears to match G semantically, but matches its shape poorly. R_1 misses one of the mental state labels, *afraid*, but contains labels that are semantically close to the weightiest term in G.

4.4 Ensemble Model

We combined the results from the *event* and *vec* models to produce an ensemble model (*ens*) which, for a mental state label m, returns the average of m's scores according to the response distributions of the two individual models.

5 Evaluation Measures

Let R denote the response distribution over mental state labels produced for a single video by one of the models described in the previous section, and let G denote the gold standard distribution produced for the same video by MTurk workers. If R is similar to G then our models produce similar mental state terms as the workers. There are many ways to compare distributions (e.g., KL distance, chi-square statistics) but these give bad results when distributions are sparse. More importantly, for our purposes, the measures that compare the shapes of distributions do not allow semantic comparisons at the level of distribution elements. Suppose R assigns high scores to *angry* and *mad*, only, while G assigns a high score to *happy*, only. Clearly, R is wrong. But if instead G had assigned a high score to *irate*, only, then R would be more right than wrong because, at the level of the individual elements, *angry* and *mad* are similar to *irate* but not similar to *happy*.

We describe a series of measures, starting with the familiar F_1 score, and discuss their applicability. To illustrate the effectiveness of each measure, we will use the examples shown in Table 2.

5.1 F_1 Score

The F_1 *score* measures the similarity between two sets of elements, R and G. $F_1 = 1$ when $R = G$

and $F_1 = 0$ when R and G share no elements. F_1 is the harmonic mean of *precision* and *recall*:

$$precision = \frac{|R \cap G|}{|R|}, \quad recall = \frac{|R \cap G|}{|G|}, \tag{1}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{2}$$

The F_1 score penalizes the responses in Table 3 that include semantically similar labels to those in G, and fails to reflect the weights of the labels in G and R.

5.2 Similarity-Aligned F_1 Score

Although the standard F_1 does not immediately fit our needs, it is a good starting point. We can incorporate the semantic similarity of distribution elements by generalizing the formulas for precision and recall as follows:

$$\begin{aligned} precision &= \frac{1}{|R|} \sum_{r \in R} \max_{g \in G} \sigma(r, g), \\ recall &= \frac{1}{|G|} \sum_{g \in G} \max_{r \in R} \sigma(r, g), \end{aligned} \tag{3}$$

where $\sigma \in [0, 1]$ is a function that yields the similarity between two elements. The standard F_1 has:

$$\sigma(r, g) = \begin{cases} 1, & \text{if } r = g \\ 0, & \text{otherwise} \end{cases},$$

but clearly σ can be defined to take values proportional to the similarity of r and g. We can choose from a wide range of semantic similarity and relatedness measures that are based on WordNet (Pedersen et al., 2004). The recent RNNLM of Mikolov opens the door to even more similarity measures based on vector space representations of words (Mikolov et al., 2013). After experimentations, we settled on one proposed by Hirst and St-Onge (1998). It represents two lexicalized concepts as semantically close if their WordNet synsets are connected by a path that is not too long and that "does not change direction too often" (Hirst and St-Onge, 1998). We chose this metric because it has a finite range, accommodates numerous POS pairs, and works well in practice.

Given the generalized precision and recall formulas in Eq 3, our *similarity-aligned* (SA) F_1 score can be computed in the usual way, as the harmonic mean of precision and recall (Eq 2).

SA-F_1 is inspired by the Constrained Entity-Aligned F-Measure (CEAF) metric proposed by (Luo, 2005) for coreference resolution. CEAF computes an optimal one-to-one mapping between subsets of reference and system entities before it computes recall, precision and F. Similarly, SA-F_1 finds optimal mappings between the labels of the two sets based on σ (this is what the max terms in Eq 3 do). Table 3 shows that SA-F_1 correctly rewards the use of synonyms. The high scores given to R_2, however, indicate that it does not measure the similarity between distribution shapes.

	F_1			SA-F_1			CWSA-F_1		
	p	r	f_1	p	r	f_1	p	r	f_1
$\mathcal{R}_1$	0	0	0	1	.5	$\frac{2}{3}$	1	.8	.89
$\mathcal{R}_2$	1	1	1	1	1	1	.4	.4	.4
$\mathcal{R}_3$	$\frac{1}{3}$	.5	.4	1	1	1	1	1	1

Table 3: The precision (p), recall (r), and F_1 (f_1) scores under various evaluation models are presented for the examples from Table 2. Suppose that $\sigma(\textit{irate}, \textit{angry}) = \sigma(\textit{irate}, \textit{mad}) = \sigma(\textit{afraid}, \textit{scared}) = 1$, with σ of any two identical strings being 1, and σ of all other pairs are 0.

5.3 Constrained Weighted Similarity-Aligned F_1 Score

Let $R(r)$ and $G(r)$ be the probabilities of label r in the R and G distributions, respectively. Let $\sigma^*_S(\ell)$ denote the best similarity score achievable when comparing elements from set S to ℓ using the similarity function σ. That is, $\sigma^*_S(\ell) = \max_{e \in S} \sigma(\ell, e)$. We can easily weight $\sigma^*_S(\ell)$ by the probability of ℓ. For example, we might redefine precision as $\sum_{r \in R} R(r) \cdot \sigma^*_G(r)$. However, this would not account for the probability of r in the gold standard distribution, G.

An analogy might help here: Suppose we have an unknown "mystery bag" of 100 colored pencils that we will try to match with a "response bag" of pencils. If we fill our response bag with 100 crimson pencils, while the mystery bag contains only 25 crimson pencils, then our precision score should get points only for the first 25 pencils, while the remaining 75 in the response bag should not be rewarded. For recall, the reward given for each color in the mystery bag is capped by the number of pencils of that color in the response bag. The analogy is complete when we consider that crimson pencils should perhaps be partially rewarded when matched by cardinal, rose or cerise pencils. In other words, a similarity mea-

sure should account for an accumulated mass of synonyms. Let $M_S(\ell)$ denote the subset of terms from S that have the *best* similarity score to ℓ:

$$M_S(\ell) = \{e \mid \sigma(\ell, e) = \sigma^*_S(\ell), \forall e \in S\} .$$

We define new forms of precision and recall as:

$$p = \sum_{r \in R} \min \left(R(r), \sum_{e \in M_G(r)} G(e) \right) \sigma^*_G(r) ,$$
$$r = \sum_{g \in G} \min \left(G(g), \sum_{e \in M_R(g)} R(e) \right) \sigma^*_R(g) . \quad (4)$$

The resulting *constrained weighted similarity-aligned* (CWSA) F_1 score is the harmonic mean of these new precision and recall scores. Table 3 shows that CWSA-F_1 yields the most intuitive evaluation of the response distributions, down-weighting R_2 in favor of R_3 and R_1.

6 Experimental Procedure

As described in Section 3, MTurk workers annotated 26 videos by identifying the actor types and mental state labels for each video. The actor types become query tuples of the form (*activity*, *actor*) and the mental state labels are compiled into one probability distribution over labels for each video, designated G. The query tuples were provided to our neighborhood models (Sec. 4), which returned a response distribution over mental state labels for each video, designated R.

We selected four videos of the 26 to calibrate the prune parameters γ and the interpolation parameters λ (Sec. 4). One of these videos contains children, one has police involvement, and two contain adults. We asked additional MTurk workers to annotate these videos, yielding an independent set of annotations to be used solely for calibration.

The experimental question is, how well does G match R for each video?

7 Results & Discussions

We report the average performance of our models along with two additional baseline methods in Table 4. The naïve baseline method *unif* simply binds R to the initial seed set of 160 mental state labels with uniform probability, while the stronger *freq* baseline uses the occurrence frequency distribution of the labels from the Gigaword corpus (note that only occurrences tagged as adjectives or verbs were counted). All average improvements of the ensemble model over the baseline models are significant ($p < 0.01$). Significance tests were one-tailed and were based on nonparametric bootstrap resampling with $10,000$ iterations.

	F_1			CWSA-F_1		
	p	r	f_1	p	r	f_1
unif	.107	**.750**	.187	.284	.289	.286
freq	.107	**.750**	.187	.362	.352	.355
sent	.194	.293	.227	.366	.376	.368
vec	.226	.145	.175	.399	.392	.393
coref	**.264**	.251	.253	.382	.461	.416
event	.231	.303	.256	.446	.488	.463
ens	.259	.296	**.274**	**.488**	**.517**	**.500**

Table 4: The average evaluation performance across 26 different chase videos are shown against 2 different baselines for all proposed models. Bold font indicates the best score in a given column.

Using the classical F_1 measure, the *coref* model scored highest on precision, while the ensemble method did best on F_1. Not surprisingly, no model can top the baseline methods on recall as both baselines use the entire seed set of 160 terms. Even so, the average recall for the baselines were only .750, which means that the initial seed set did not include words that were used by the MTurk annotators. As we've mentioned, the classical F_1 is misleading because it does not credit synonyms. For example, in one movie, one of our models was rewarded once for matching the label *angry* and penalized six times for also reporting *irate*, *enraged*, *raging*, *upset*, *furious*, and *mad*. Frequently, our models were penalized for using the terms *scared* and *afraid* instead of *fearful*.

Under the CWSA-F_1 evaluation measure, which correctly accounts for both synonyms and label probabilities, our ensemble model performed best. The average CWSA-F_1 score of the ensemble model improves upon the simple uniform baseline *unif* by almost 75%, and over the stronger *freq* baseline by over 40%. The ensemble method also outperforms each individual method in all measured scores. These improvements were also found to be significant. This strongly suggests that the *vec* and *event* models are complementary, and not entirely redundant. Furthermore, Table 4 shows that the *event* model performs considerably better than *coref*. This result emphasizes the importance of focusing on the mental state labels of event participants rather than considering all mental state terms collocated in the same sentence with an actor or action verb.

Models	CWSA-F1	Versus *coref*	p-value
win-0	0.388682	−0.027512	0.0067
win-1	0.415328	−0.000866	0.4629
win-2	0.399777	−0.016417	0.0311
win-3	0.392832	−0.023362	0.0029

Table 5: The average CWSA-F_1 scores for the *win-n* model with different window parameters are shown in comparison to the *coref* model. The *coref* model outperformed all tested configurations, though the difference is not significant for $n = 1$. The p-value based on the average differences were obtained using one-tailed nonparametric bootstrap resampling with $10,000$ iterations.

Table 5 explores the effectiveness of coreference resolution in expanding the neighborhood area. The *coref* model outperformed the simple windowing method under every tested configuration. However, the improvement over windowing with $n = 1$ is not significant. This can be explained by fact that immediately neighboring sentences are more likely to be related. Moreover, since newswire articles tend to be short, the neighborhoods generated by *win-1* tend to be similar to those generated by *coref*. In general, *coref* does not do worse than a simple windowing method and has the bonus advantage of providing references to the actors of interest for downstream processes.

In Table 6, we show the performance results based on the types of chase scenarios happening in the videos. The average scores under the uniform baseline *unif* for chase videos involving children and sporting events are lower than for police and other chases. This suggests that our seed set of 160 mental state labels is biased towards the latter types of events, and is not as fit to describe chases involving children.

On average, videos involving police officers show the biggest improvement in the CWSA-F_1 scores over the *unif* baseline ($+0.2693$), whereas videos involving children received the lowest gain ($+0.1517$). We believe this is the effect of the Gigaword text corpus, which is a comprehensive archive of newswire text, and thus is heavily biased towards high-speed and violent chases involving the police. The Gigaword corpus is not the place to find children happily chasing each other. Similarly, sports-related chases, which are also news-worthy, have a higher gain than children's videos on average.

Categories	Unif	Ensemble	Gain
children	0.2082	0.3599	+0.1517
police	**0.3313**	**0.6006**	**+0.2693**
sports	0.2318	0.4126	+0.1808
others	0.3157	0.5457	+0.2300

Table 6: The average CWSA-F_1 scores for the ensemble model are shown in comparison to the uniform baseline method, categorize by video types.

8 Conclusion and Future Work

We introduced the novel task of identifying latent attributes in video scenes, specifically the mental states of actors in chase scenes. We showed that these attributes can be identified by using explicit features of videos to query text corpora, and from the resulting texts extract attributes that are latent in the videos. We presented several largely unsupervised methods for identifying distributions of actors' mental states in video scenes. We defined a similarity measure, CWSA-F_1, for comparing distributions of mental state labels that accounts for both semantic relatedness of the labels and their probabilities in the corresponding distributions. We showed that very little information from videos is needed to produce good results that significantly outperform baseline methods.

In the future, we plan to add more detection types. Additional contextual information from videos (e.g., scene locations) should help improve performance, especially on tougher videos (e.g., videos involving children chases). Moreover, we believe that the initial seed set of mental state labels can be learned simultaneously with the extraction patterns of the *event* model using a mutual bootstrapping method, similar to that of (Riloff and Jones, 1999).

Currently, our experiments assume one distribution of mental state labels for each video. They do not distinguish between the mental states of the chaser and chasee, while in reality these participants may be in very different states of mind. Our *event* model is capable of making this distinction and we will test its performance on this task in the future. We also plan to test the effectiveness of our models with actual computer vision detectors. As a first approximation, we will simulate the noisy nature of detectors by degrading the quality of annotated data. Using artificial noise on ground-truth data, we can simulate the performance of real detectors and test the robustness of our models.

References

Abdul Rehman Abbasi, Matthew N. Dailey, Nitin V. Afzulpurkar, and Takeaki Uno. 2009. Student mental state inference from unintentional body gestures using dynamic Bayesian networks. *Journal on Multimodal User Interfaces*, 3(1-2):21–31, December.

Tadas Baltrusaitis, Daniel McDuff, Ntombikayise Banda, Marwa Mahmoud, Rana el Kaliouby, Peter Robinson, and Rosalind Picard. 2011. Real-time inference of mental states from facial expressions and upper body gestures. In *Face and Gesture 2011*, pages 909–914. IEEE, March.

Thorsten Brants. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231, Morristown, NJ, USA. Association for Computational Linguistics.

Ernesto Brau, Jinyan Guan, Kyle Simek, Luca Del Pero, Colin Reimer Dawson, and Kobus Barnard. 2013. Bayesian 3D Tracking from monocular video. In *The IEEE International Conference on Computer Vision (ICCV)*, December.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics* -, pages 16–23, Morristown, NJ, USA. Association for Computational Linguistics.

R. El Kaliouby and P. Robinson. 2004. Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 154–154. IEEE.

Pedro Felzenszwalb, David McAllester, and Deva Ramanan. 2008. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June.

Ryan Gabbard, Marjorie Freedman, and RM Weischedel. 2011. Coreference for learning to extract relations: yes, Virginia, coreference matters. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, pages 288–293.

Maria Gendron, Debi Roberson, Jacoba Marieta van der Vyver, and Lisa Feldman Barrett. 2014. Cultural relativity in perceiving emotion from vocalizations. *Psychological science*, 25(4):911–20, April.

J Giebel, DM Gavrila, and C Schnörr. 2004. A bayesian framework for multi-cue 3d object tracking. In *Computer Vision-ECCV 2004*, pages 241–252.

Graeme Hirst and D St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, pages 305–332. The MIT Press.

LJ Li, Hao Su, L Fei-Fei, and EP Xing. 2010. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in Neural Information Processing Systems*.

Zhilei Liu and Shangfei Wang. 2011. Emotion recognition using hidden Markov models from facial temperature sequence. In *ACII'11 Proceedings of the 4th international conference on Affective computing and intelligent interaction - Volume Part II*, pages 240–247.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

MC De Marneffe, CD Manning, and Christopher Potts. 2010. "Was it good? It was provocative." Learning the meaning of scalar adjectives. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 167–176.

Stephen J. McKenna, Sumer Jabri, Zoran Duric, Azriel Rosenfeld, and Harry Wechsler. 2000. Tracking Groups of People. *Computer Vision and Image Understanding*, 80(1):42–56, October.

D M McNair, M Lorr, and L F Droppleman. 1971. Profile of Mood States (POMS).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, pages 1–12.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, November.

Mitra Mohtarami, Hadi Amiri, Man Lan, and Chew Lim Tan. 2011. Predicting the uncertainty of sentiment adjectives in indirect answers. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, page 2485, New York, New York, USA. ACM Press.

CB Ng, YH Tay, and BM Goi. 2012. Recognizing human gender in computer vision: a survey. *PRICAI 2012: Trends in Artificial Intelligence*, 7458:335–346.

S O'Hara and B. A. Draper. 2012. Scalable action recognition with a subspace forest. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1210–1217. IEEE, June.

Ted Pedersen, S Patwardhan, and J Michelizzi. 2004. WordNet::Similarity: measuring the relatedness of concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 1024–1025, San Jose, CA.

Ronald Poppe. 2010. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June.

Deva Ramanan, David a Forsyth, and Andrew Zisserman. 2007. Tracking people by learning their appearance. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):65–81, January.

E Riloff and R Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the sixteenth national conference on Artificial intelligence (AAAI-1999)*, pages 474–479.

S. Sadanand and J. J. Corso. 2012. Action bank: A high-level representation of activity in video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1234–1241. IEEE, June.

C Schuldt, I Laptev, and B Caputo. 2004. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pages 32–36 Vol.3. IEEE.

M. Sokolova and G. Lapalme. 2011. Learning opinions in user-generated web content. *Natural Language Engineering*, 17(04):541–567, March.

Daniel Weinland, Remi Ronfard, and Edmond Boyer. 2011. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, February.

Yi Yang and Deva Ramanan. 2011. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392. IEEE, June.