EXTRACTING LATENT ATTRIBUTES FROM VIDEO SCENES USING TEXT AS BACKGROUND KNOWLEDGE

Anh Tran, Mihai Surdeanu, Paul Cohen
University of Arizona

*Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*

Motivation: Surveillance Application

- Two chase scenarios: Should the police be notified?

dark gun *angry / fear / distress*

light no-gun *joyful / happy*

- A surveillance system could automatically dispatch the police if it can identify that someone is in a **state of distress**.
  - Need **latent attributes** (e.g., mental state information) about the scene

Problem Definition

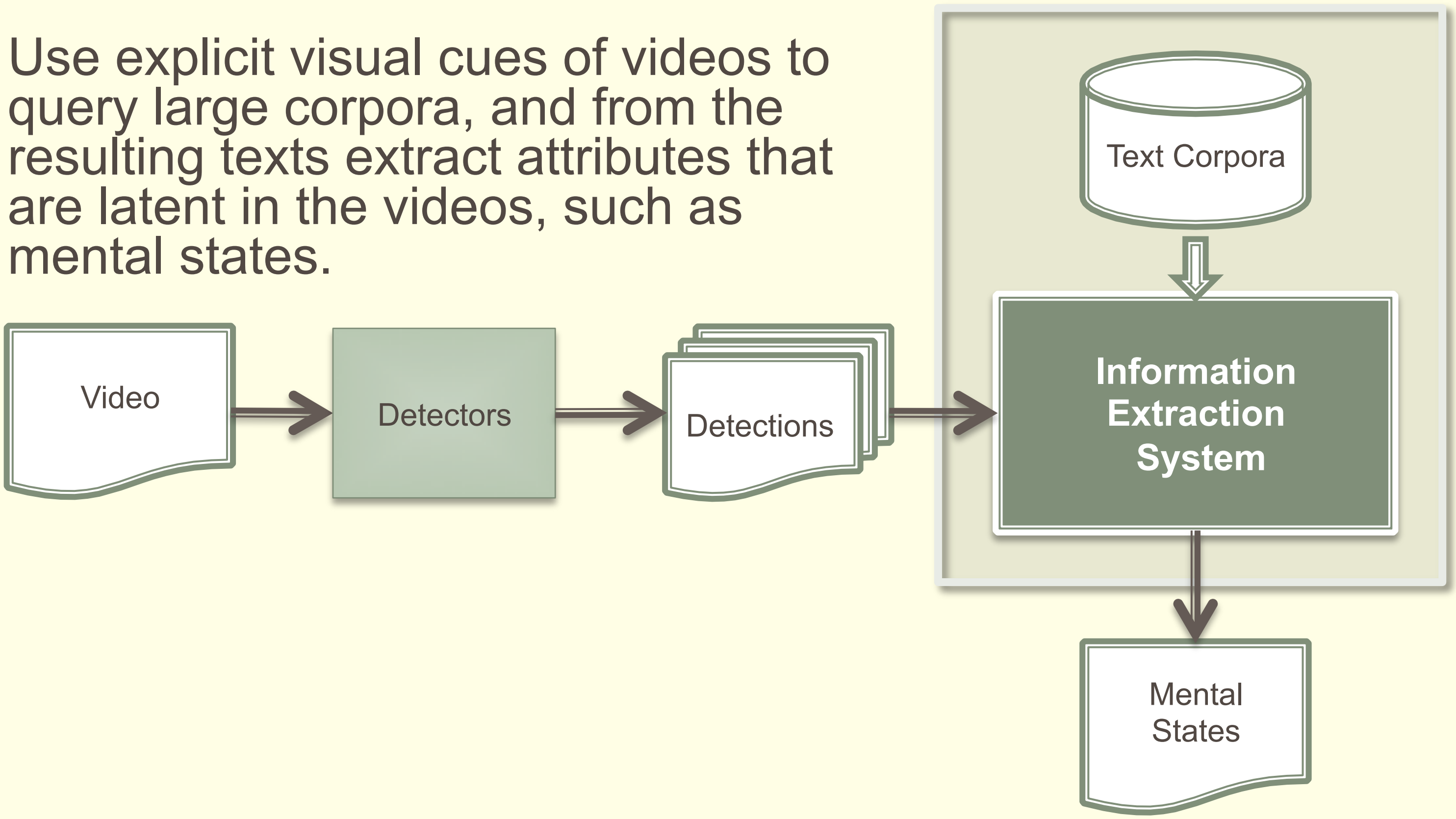
Identifying latent attributes from video scenes, with a focus on the mental states of activity participants, given some contextual information about the scenes.

- **Latent attributes:** unobservable elements about the scene.
  - Mental states, motives, intentions, etc.
- **Contextual knowledge:** any observable elements (or cues).
  - Activity, object, actor type (child vs. policeman)
- Automatic identification of latent attributes is a challenging task.
  - No access to the same background knowledge that humans possess.
  - Machines can only “detect” explicit contents (e.g., observable cues).
  - How to identify latent information using these cues as the the contextual knowledge?

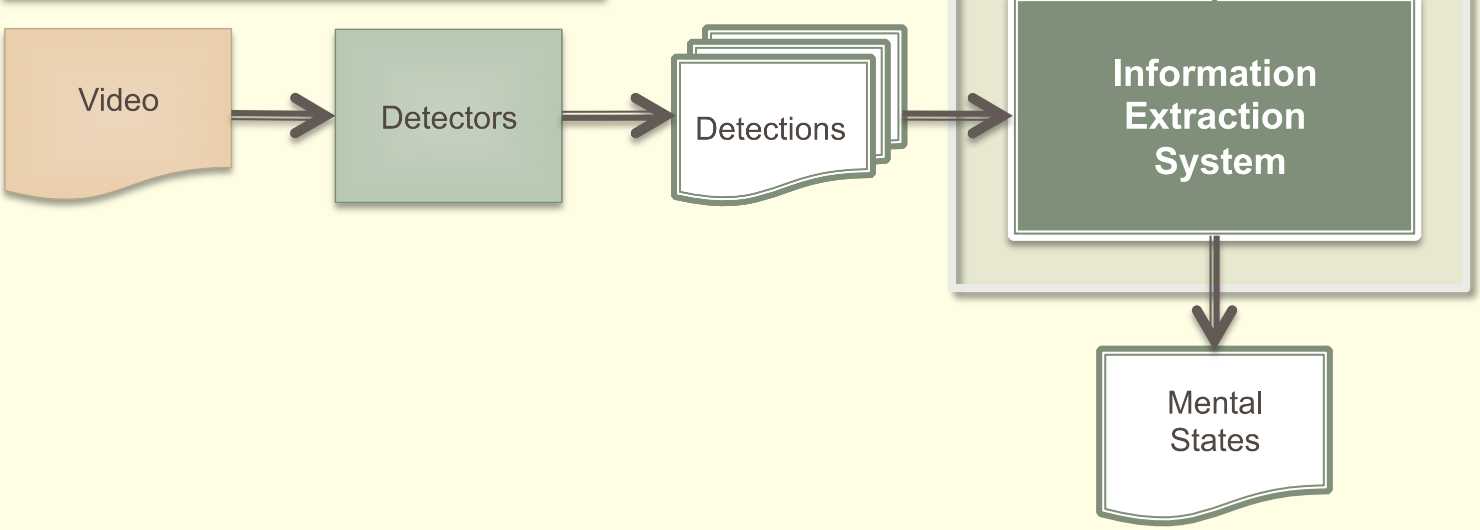
The Approach

Attributes that are latent in videos are often explicit in text.

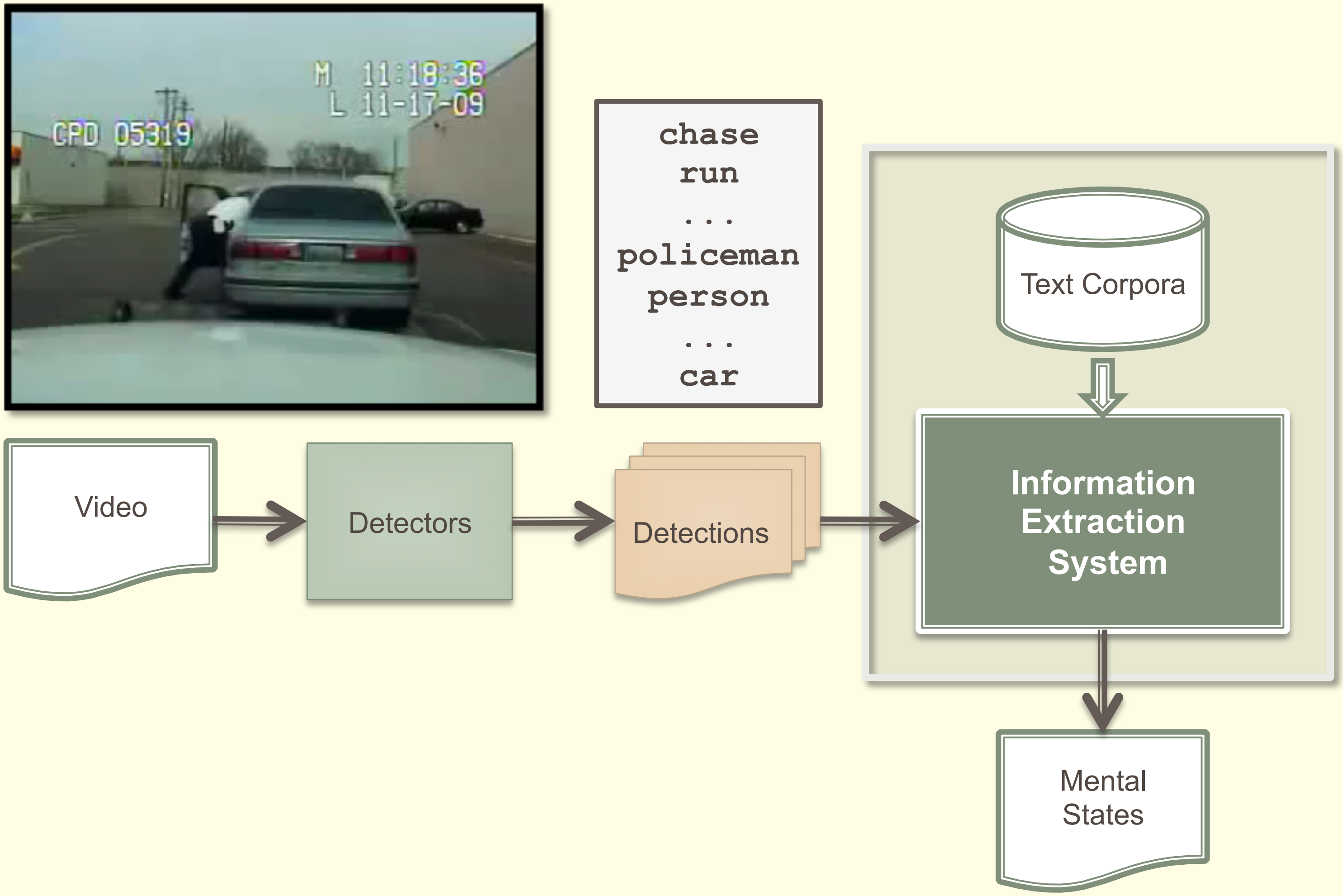
- Use explicit visual cues of videos to query large corpora, and from the resulting texts extract attributes that are latent in the videos, such as mental states.



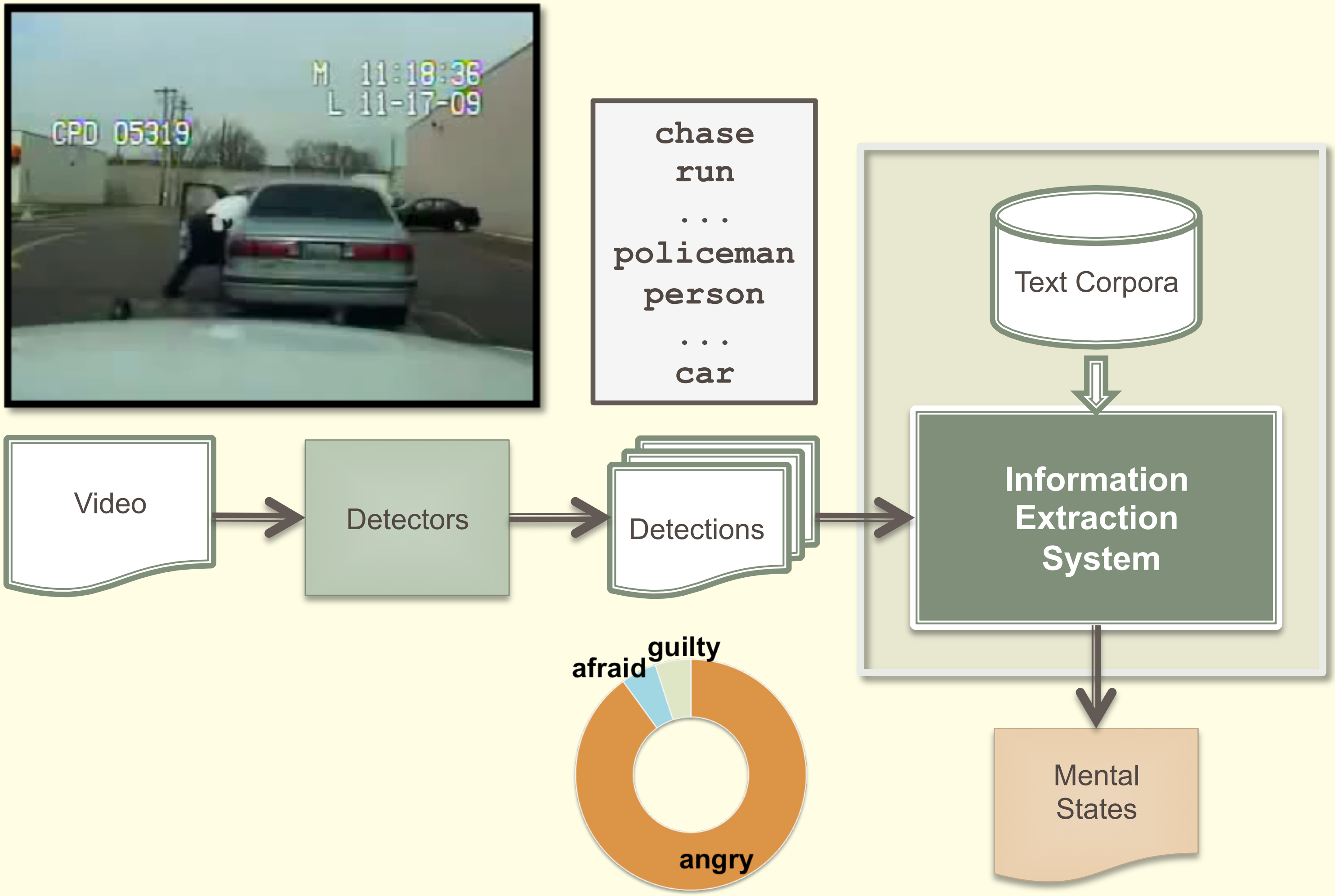
Run-through Example – Video



Run-through Example – Detection Labels

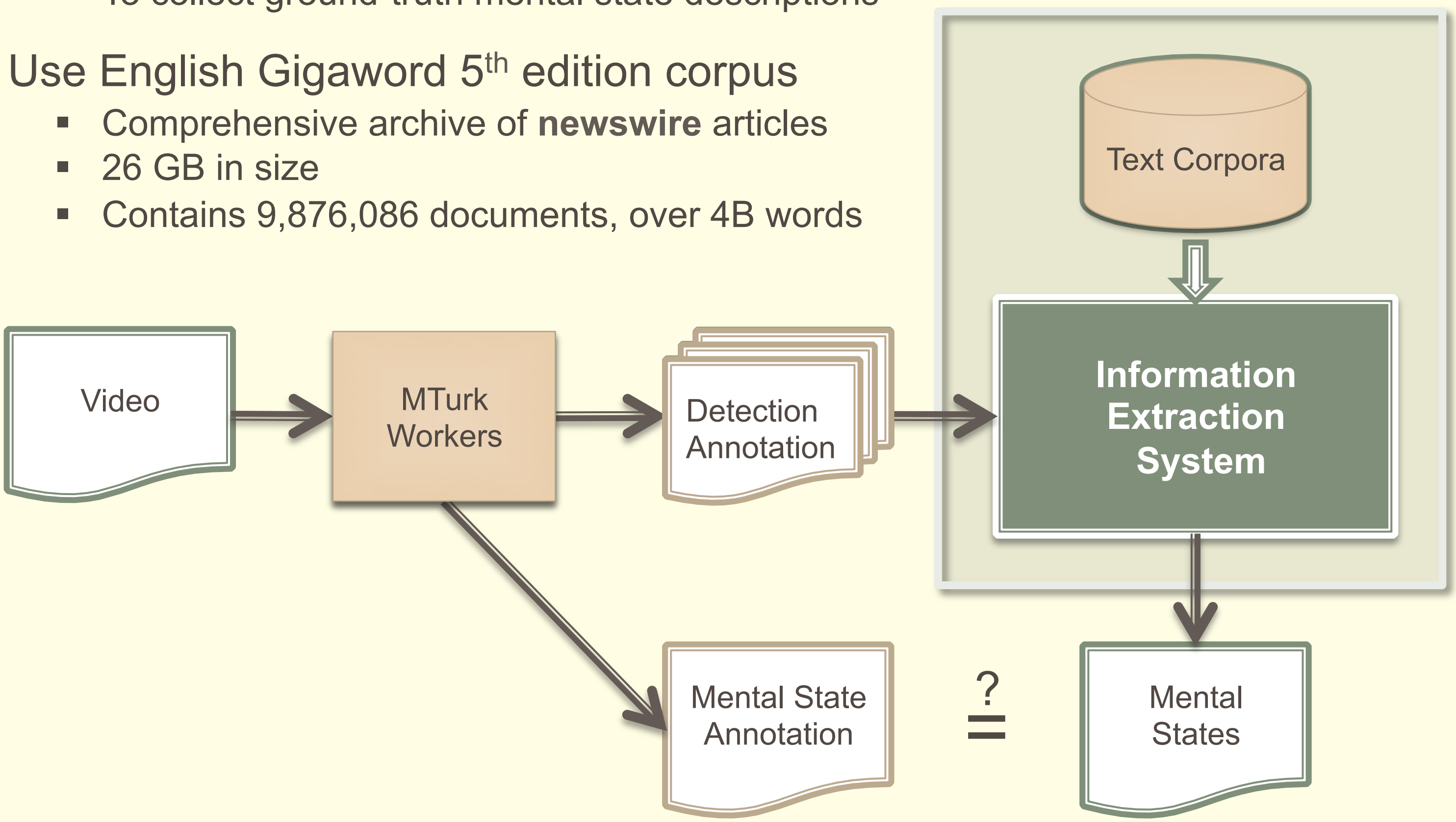


Run-through Example – Mental States



Data Sources

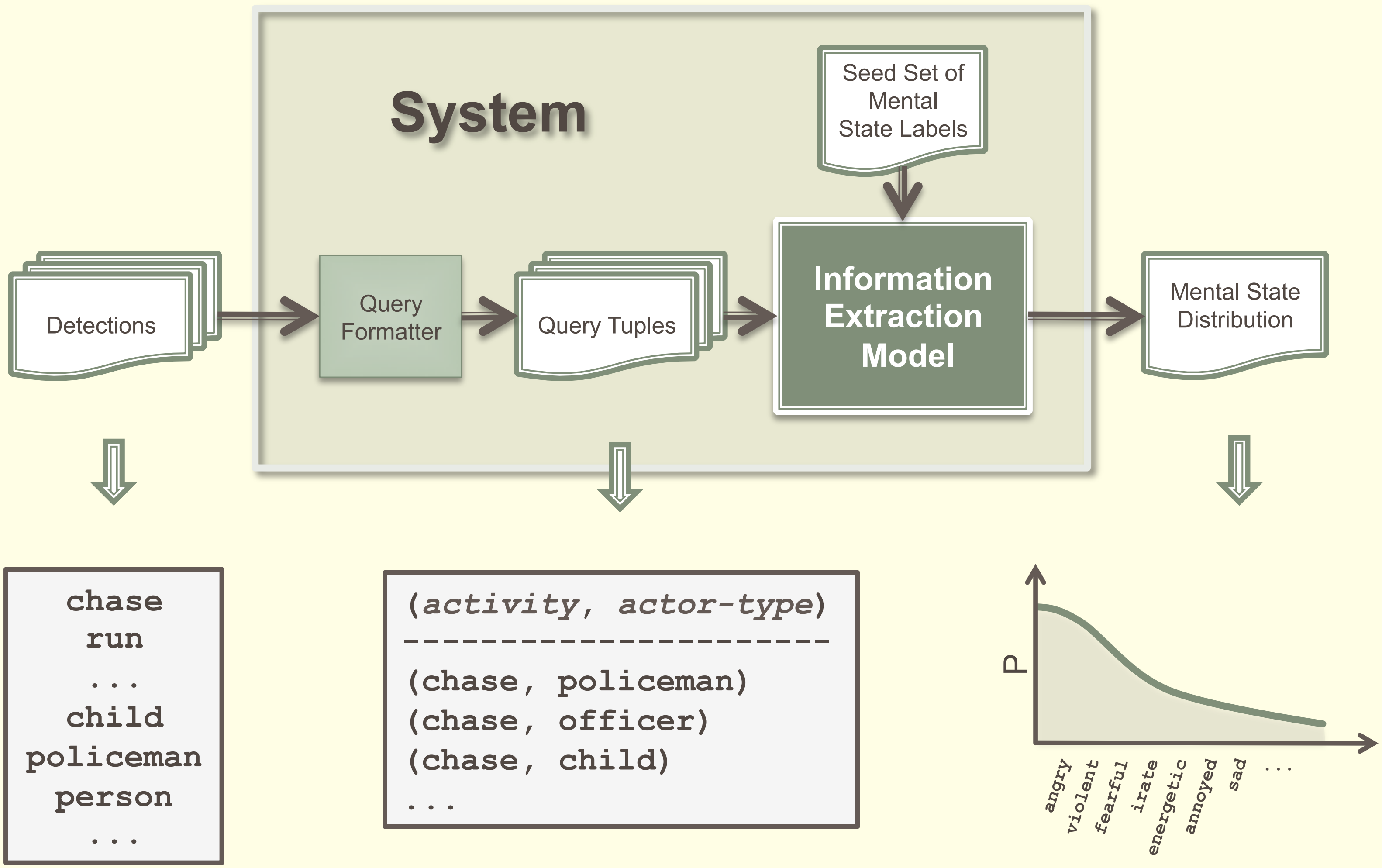
- Use Amazon Mechanical Turk workers
  - As proxy for automatic detection system
  - To collect ground-truth mental state descriptions
- Use English Gigaword 5th edition corpus
  - Comprehensive archive of **newswire** articles
  - 26 GB in size
  - Contains 9,876,086 documents, over 4B words



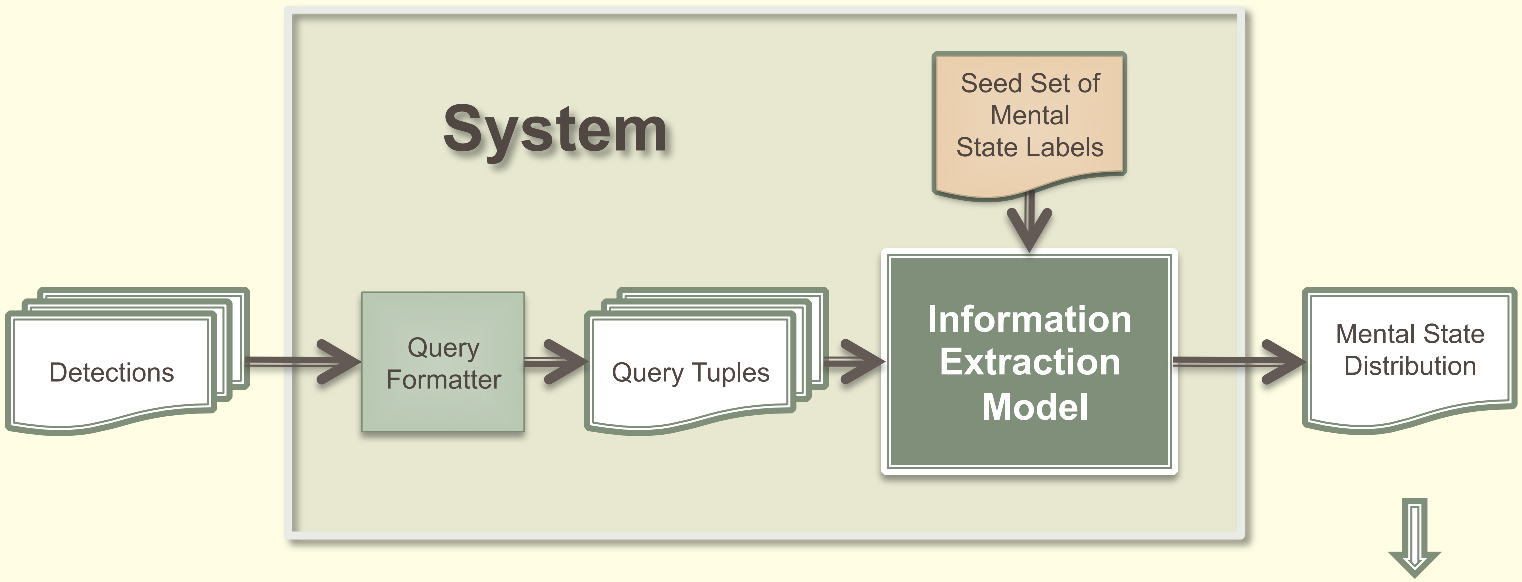
Video Dataset

- Generated a dataset of 26 chase videos:
  - Mixture of police (5), children (7), sport-related (4), and other (12) chases.

Detailed System Overview

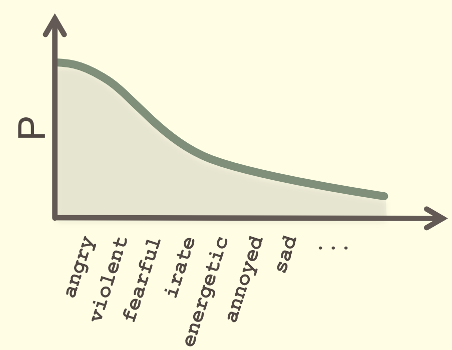


Seed Set of Mental State Labels

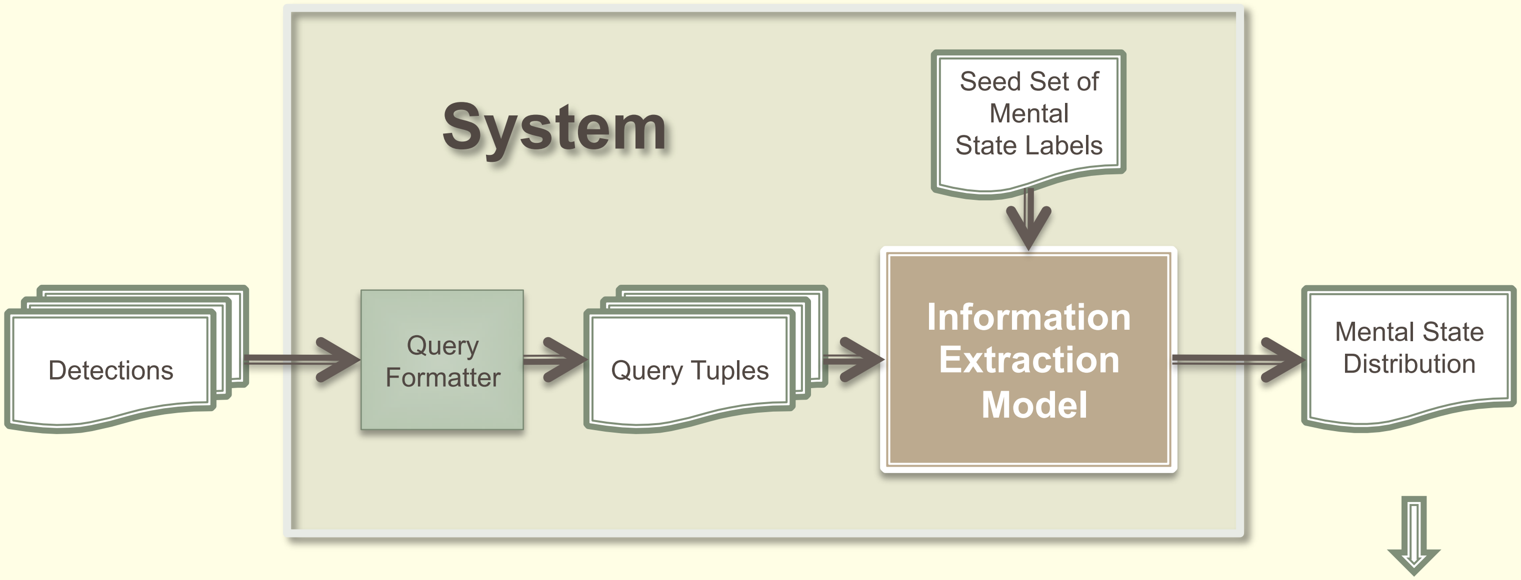


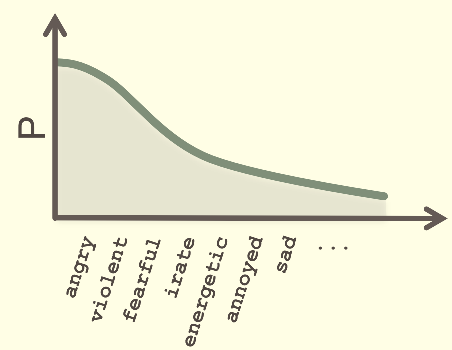
Source	Example Mental State Labels
POMS	alert, annoyed, energetic, exhausted, helpful, sad, terrified, unworthy, weary, etc.
Plutchik	angry, disgusted, fearful, joyful/joyous, sad, surprised, trusting, etc.
Others	agitated, competitive, cynical, disappointed, excited, giddy, happy, inebriated, violent,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

160 mental state labels total



Neighborhood Models





Three largely unsupervised information extraction models.

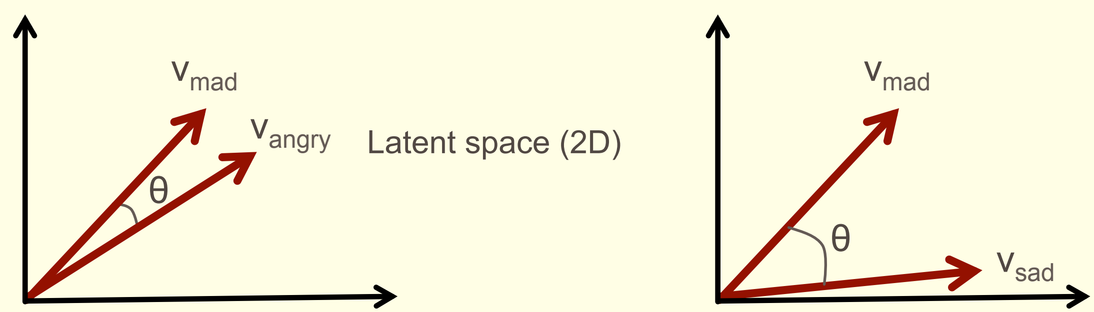
Vector Space with Back-off Linear Interpolation

- **Idea:** Project mental state labels and search context into latent conceptual space produced by a RNNLM (Mikolov et al., 2013a).

mad → Recurrent Neural Network Language Model → v_{mad} = [0.4, 0.9, …, 0.8]

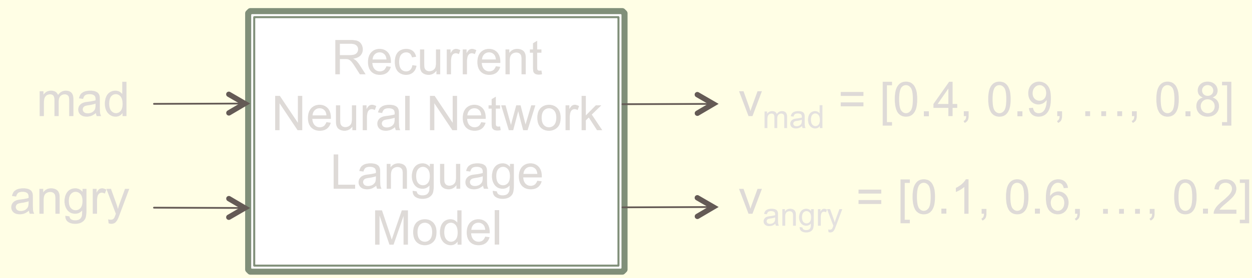
angry → Recurrent Neural Network Language Model → v_{angry} = [0.1, 0.6, …, 0.2]

- Compare in latent space using the angle between the vectors.

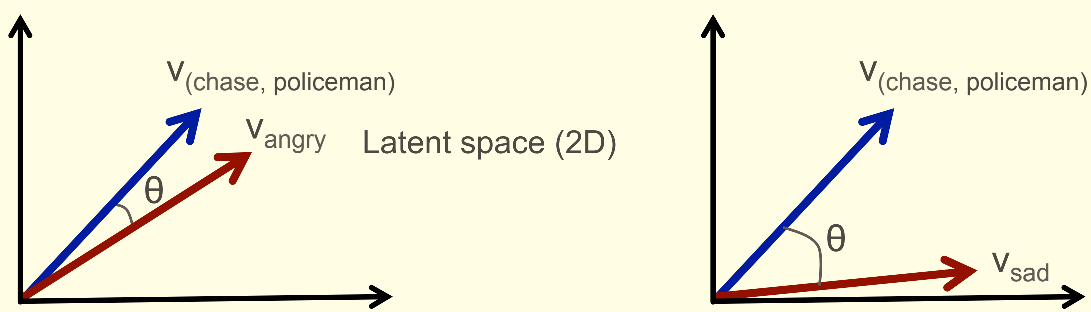


Vector Space with Back-off Linear Interpolation

- **Idea:** Project mental state labels and search context into latent conceptual space produced by a RNNLM (Mikolov et al., 2013a).



- Compare mental state labels to query tuple in latent space.



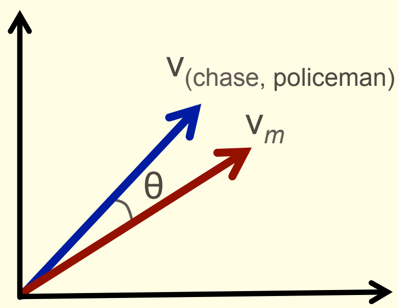
Vector Space with Back-off Linear Interpolation

- Compute context-vector for query tuple:

$$vec(chase, policeman) = vec(chase) + vec(policeman)$$

- Compute similarity to each mental state *m*:

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



- → 160 scores per context (or query) tuple.
- Normalize scores to generate a distribution per tuple, average across tuples to create one distribution, and prune to yield final response distribution.
- Improve robustness with back-off model (see paper for details)

Sentence Co-occurrence with Deleted Interpolation

- **Idea**: Words in the same sentence are likely to be related.
- Rank mental state labels based on the likelihood that they appear in sentences cued by query tuples.
- Interested in the conditional probability:

$$P(m|activity, actor\text{-}type) = \frac{f(m, activity, actor\text{-}type)}{f(activity, actor\text{-}type)}$$

- Normally, we could compute this probability based on relative frequencies.
- However, estimation is unreliable due to sparse data!

Sentence Co-occurrence with Deleted Interpolation

- Cannot estimate probability of trigrams reliably from the corpus, so we estimate probability as linear interpolation of unigrams, bigrams, trigrams.
- Define maximum likelihood probabilities $\hat{P}$ based on relative frequencies:

$$\text{Unigram:} \quad \hat{P}(m) = \frac{f(m)}{N}$$

$$\text{Bigram:} \quad \hat{P}(m|activity) = \frac{f(m, activity)}{f(activity)}$$

$$\text{Trigram:} \quad \hat{P}(m|activity, actor\text{-}type) = \frac{f(m, activity, actor\text{-}type)}{f(activity, actor\text{-}type)}$$

  - N = total number of tokens in the corpus
  - *f*(*m*, *activity*) = number of sentences containing both *m* as an adjective and *activity* as a verb

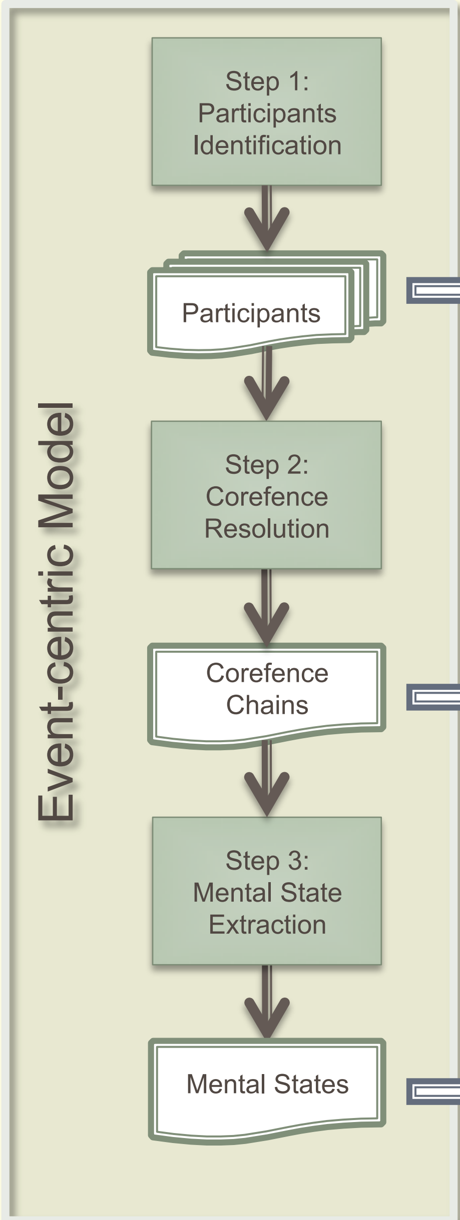
$$P(m|activity, actor\text{-}type) \ = \ \lambda_1\hat{P}(m) + \lambda_2\hat{P}(m|activity) + \lambda_3\hat{P}(m|activity, actor\text{-}type)$$

- Use deleted interpolation to estimate lambdas.
- 160 trigram probabilities for each query tuple, average across all query tuples and prune to yield final response distribution.

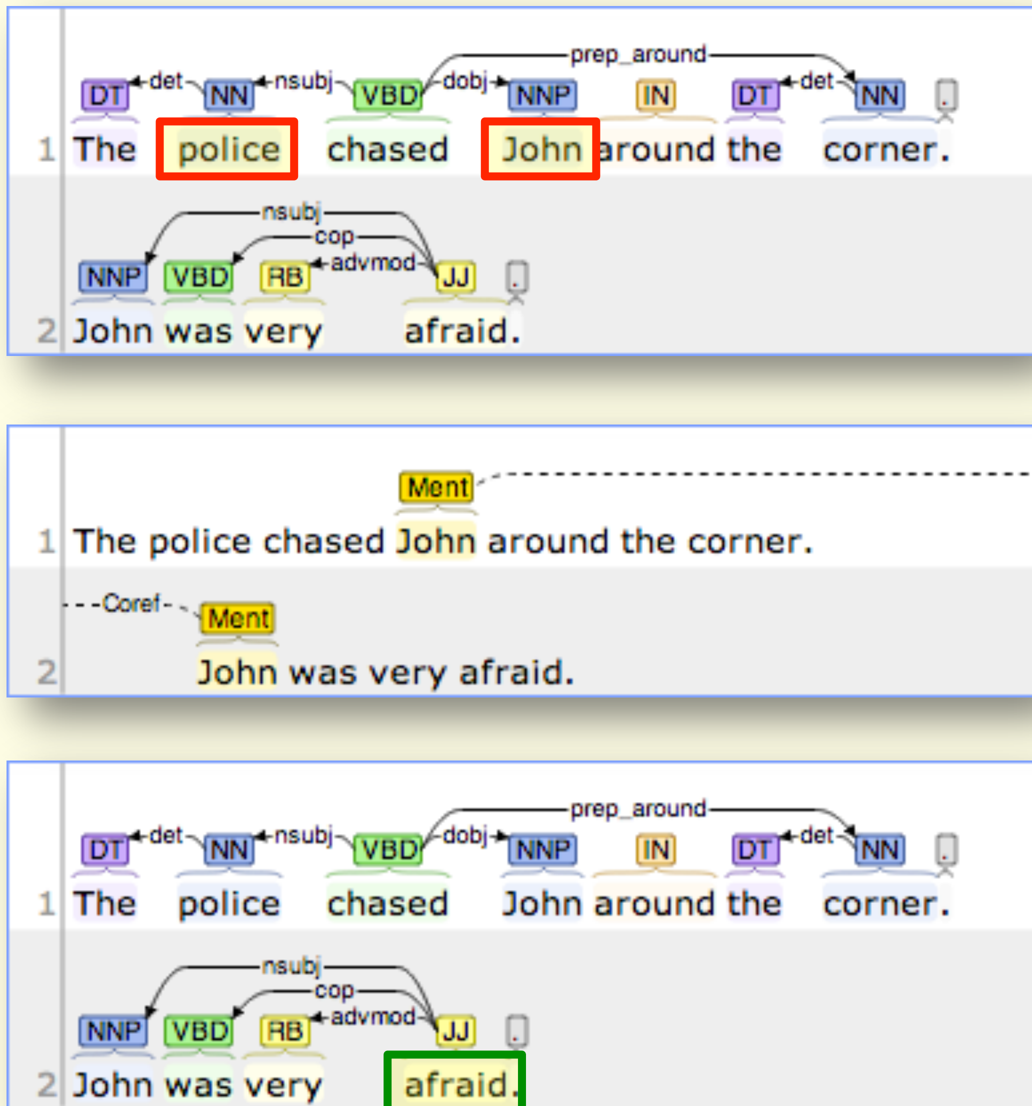
Event-centric with Deleted Interpolation

- **Idea:** Identify the event + its participants in the relevant sentences and focus only on the mental states of event participants.

- A smarter, more robust, way to find collocating mental states for joint frequency estimation.
  - Go beyond sentence boundary.
  - Focus on mental states of participants.



Example Output



Evaluation Measure

- New task → No standard performance measure
- Need to compare two normalized distributions over mental state labels.
- Similarity of **distribution shapes (weights)**
  - A good measure must account for the similarity between the shapes of the two distributions (i.e., ratios between weights)
- Semantic similarity of **distribution elements (synonyms)**
  - A good measure must allow for semantic comparisons at the level of distribution elements (i.e., recognize that irate and angry are similar)

Gold G	$(\text{angry}, 0.9), (\text{afraid}, 0.05), (\text{guilty}, 0.05)$
Response R_1	$(\text{angry}, 0.1), (\text{afraid}, 0.2), (\text{guilty}, 0.7)$
Response R_2	$(\text{irate}, 0.45), (\text{mad}, 0.45), (\text{scared}, 0.05), (\text{guilty}, 0.05)$

Evaluation Toy Example

Gold G	$(\text{angry}, 0.9), (\text{afraid}, 0.05), (\text{guilty}, 0.05)$
Response R_1	$(\text{angry}, 0.1), (\text{afraid}, 0.2), (\text{guilty}, 0.7)$
Response R_2	$(\text{irate}, 0.45), (\text{mad}, 0.45), (\text{scared}, 0.05), (\text{guilty}, 0.05)$

Constrained Weighted Similarity-Aligned F1

	F_1			CWSA-F_1		
	p	r	f_1	p	r	f_1
R_1	1	1	1	0.2	0.2	0.2
R_2	0.25	$0.3\bar{3}$	0.29	1	1	1

red = non-intuitive score
green = intuitive score

Mental State Identification in Chase Videos

	F_1			CWSA-F_1		
	p	r	f_1	p	r	f_1
baseline	.107	**.750**	.187	.284	.289	.286
sentence	.194	.293	.227	.366	.376	.368
vector	.226	.145	.175	.399	.392	.393
event	.231	.303	.256	.446	.488	.463
event+vector	**.259**	.296	**.274**	**.488**	**.517**	**.500**

The average evaluation performance across 26 different chase videos are shown against the baseline scores for our neighborhood information extraction models. Bold font indicates the best score in a given column.

* All average improvements over the baseline responses are significant ($p < 0.01$). All significance tests were one-tailed and were based on nonparametric bootstrap resampling with 10,000 iterations.

Mental State Identification in Chase Videos

	F_1			CWSA-F_1		
	p	r	f_1	p	r	f_1
baseline	.107	**.750**	.187	.284	.289	.286
sentence	.194	.293	.227	.366	.376	.368
vector	.226	.145	.175	.399	.392	.393
event	.231	.303	.256	.446	.488	.463
event+vector	**.259**	.296	**.274**	**.488**	**.517**	**.500**

- *event+vector* outperformed baseline by almost 75%.
- Ensemble outperformed individual components.
  - Operating in latent space (*vector*) and operating on text (*event*) yield complementary information.
- Incremental improvements due to NLP.
  - *sentence* is similar to current state-of-the-art (e.g., de Marneffe et al. 2010)
  - *sentence* < *vector* < *event*

Additional Results* – Actor-specific

	CWSA-F_1		
	p	r	f_1
baseline	.196	.195	.195
vector	.351	.338	.342
event	.353	.340	.340
event+vector	**.395**	**.400**	**.396**

The average evaluation performance for the mental state of the **subject** across 26 different chase videos.

	CWSA-F_1		
	p	r	f_1
baseline	.191	.181	.185
vector	.358	.374	.363
event	.383	.407	.391
event+vector	**.389**	**.415**	**.399**

The average evaluation performance for the mental state of the **object** across 26 different chase videos.

* Results not included in paper

Additional Results* – Hug Dataset

	CWSA-F_1		
	p	r	f_1
baseline	.226	.210	.217
vector	.347	.334	.339
sentence	.388	.378	.382
event	.406	.384	.394
event+vector	**.443**	**.437**	**.439**

- Performance average across 45 hug videos
- *event+vector* outperformed baseline by over 100%
- Consistent behavior as in chase videos:
  - Ensemble outperformed individual components.
  - Incremental improvement with each NLP module.

* Results not included in paper

Additional Results* – Noisy Detections

- **Introduce noise** into detections **based on published precision rates** of actual state-of-the-art detectors in computer vision
  - False-positive: randomly insert new detection label
  - False-negative: randomly withhold an annotated detection label

- Average errors introduced per movie

Stat Type	No. Occurrences per Movie
True Positives	3.17
False Negatives	2.83
True Negatives	9.75
False Positives	7.25

- Performance average across 20 different simulations (26 chase videos per simulation)

	F_1			CWSA-F_1		
	p	r	f_1	p	r	f_1
event+vector	.231	.255	.239	.443	.437	.439
baseline	.107	**.750**	.187	.284	.289	.286

* Results not included in paper

Conclusions

- Summary
  - **Problem**: Identifying latent attributes in videos, given some context.
  - **Data**: Videos from web, annotations via crowd sourcing
  - **Solution**: Largely unsupervised information extraction models
    - Lexical semantic in vector space (*vector*)
    - Sentence co-occurrence in text (*sentence*)
    - Event-centric in text (*event*)
  - **Evaluation**: CWSA-F_1 score to compare mental state distributions
- Findings
  - First to show how to identify latent information from videos using text collections as the sole background knowledge
  - More NLP → better performance
  - Robust models: work on different datasets, tolerate noisy detections
- All code (Scala) + data (annotations & videos) at
  - **https://trananh.github.io/vlsa/**

THE END

BACKUP SLIDES

admiring	cranky	fearful	instinctive	pleased	sympathetic
afraid	crazy	focused	interested	protective	tense
aggressive	curious	forgetful	irate	raging	terrified
agitated	cynical	frantic	irritated	rebellious	terrifying
alarmed	demented	friendly	jealous	refreshed	thankful
alert	depressed	frightened	joyful	relaxed	threatening
ambitious	desperate	frustrated	joyous	relieved	tired
amazed	determined	fun	lively	reluctant	trustful
amused	devious	furious	loathsome	remorseful	trusting
angry	disappointed	giddy	lonely	resentful	uncomfortab
annoyed	discontent	glamorous	loved	restless	uneasy
anxious	discouraged	gleeful	mad	revengeful	unhappy
apprehensive	disgusted	grateful	mellow	romantic	unworthy
ashamed	distracted	grumpy	merciless	sad	upset
assertive	drunken	guilty	mischievous	satisfied	urgent
bitter	eager	happy	miserable	scared	vengeful
bored	ecstatic	helpful	motivated	selfish	vigilant
calm	encouraged	helpless	naughty	selfless	vigorous
carefree	energetic	homicidal	nervous	serious	violent
cautious	energized	hopeful	numb	shaky	wary
cheerful	enraged	hopeless	optimistic	shocked	weary
competitive	enthusiastic	hostile	panicked	sickened	weird
complacent	envious	hurried	panicky	spiteful	welcoming
concerned	excited	impressed	peaceful	stressed	worried
confused	exhausted	indifferent	peeved	submissive	worthless
considerate	exhilarating	inebriated	pessimistic	surprised	
content	fatigued	infuriated	playful	suspenseful	

participants-nlp-3

```
==== DOCUMENT ====
The police chased John around the corner . John was very afraid .


==> Step 1: PARTICIPANTS IDENTIFICATION

Target Sentence(s):
"The police chased John around the corner ."

Identified subject (CoreNLP):   "police"
Identified object (CoreNLP):    "John"


==> Step 2: COREFERENCE RESOLUTION

Coreference chain(s) for SUBJECTS:
One chain found containing the following mentions:
    sentence (0):  [The police] chased John around the corner .

Coreference chain(s) for OBJECTS:
One chain found containing the following mentions:
    sentence (1):  [John] was very afraid .
    sentence (0): The police chased [John] around the corner .


==> Step 3: COMPLEMENTS EXTRACTION

Complement(s) for SUBJECTS:
    sentence (0): NONE

Complement(s) for OBJECT:
    sentence (0): NONE
    sentence (1): afraid
```

Line: 1 Column: 1 Plain Text Tab Size: 4

Constrained Weighted Similarity-Aligned F1

- We start with the standard F_1 measure.

$$precision = \frac{|R \cap G|}{|R|}\ ,\ recall = \frac{|R \cap G|}{|G|}\ ,\ F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$ F_1

- Generalize the formulas to address our criteria.

$$precision = \frac{1}{|R|} \sum_{r \in R} \max_{g \in G} \sigma(r,g) \qquad \sigma(r,g) = \begin{cases} 1\ , & \text{if } r = g \\ 0\ , & \text{otherwise} \end{cases}$$ SA-F_1

$$= \sum_{r \in R} R(r) \cdot \max_{g \in G} \sigma(r,g) \qquad R(r) = \frac{1}{|R|}$$

$$= \sum_{r \in R} R(r) \cdot \sigma^*_G(r)$$ WSA-F_1

- Address greedy problem of WSA-F_1

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

$$precision = \sum_{r \in R} \min\left(R(r), \sum_{e \in M_G(r)} G(e)\right) \cdot \sigma^*_G(r)$$ CWSA-F_1

Constrained Weighted Similarity-Aligned F1

- We start with the standard F_1 measure.

$$precision = \frac{|R \cap G|}{|R|}, \; recall = \frac{|R \cap G|}{|G|}, \; F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

F_1

- Genera… to address ou…

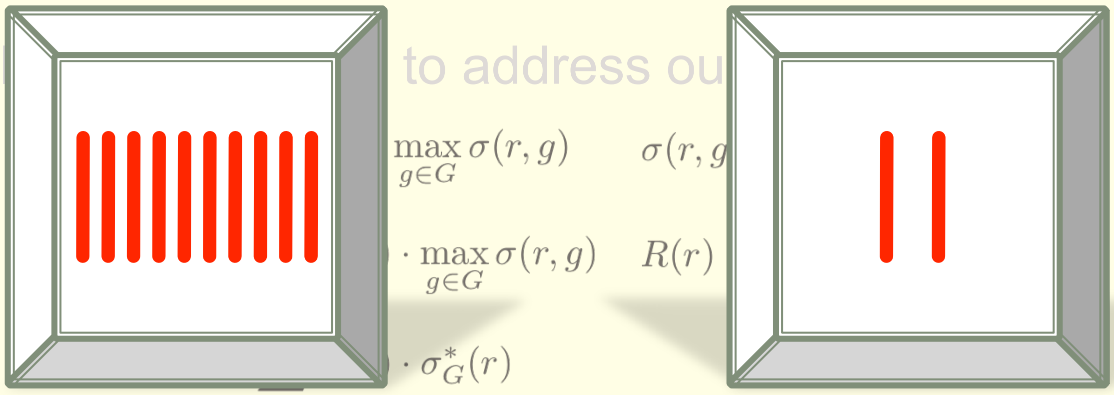
$\max_{g \in G} \sigma(r, g)$ $\quad \sigma(r, g$…

SA-F_1

$\cdot \max_{g \in G} \sigma(r, g)$ $\quad R(r)$

$\cdot \sigma^*_G(r)$ $\quad r \in R$

WSA-F_1



- Address greedy problem of WSA-F_1

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

$$precision = \sum_{r \in R} \min\left(R(r), \sum_{e \in M_G(r)} G(e)\right) \cdot \sigma^*_G(r)$$

CWSA-F_1

Limitation: Biases in Underlying Data

Categories	Baseline	*event+vector*	Change
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

The average CWSA-F_1 scores for the ensemble model *event+vector* are shown in comparison to the baseline performance, categorized by video scenarios.

children = video contains a child participant
police = video contains a policeman participant
sports = video is sports-related
other = video does not fit in the first categories (e.g., civilian adults)

Limitations: Biases in Underlying Data

Categories	Baseline	*event+vector*	Change
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

- Baseline did worse on children and sports-related videos than police related videos.
- Baseline uses all 160 mental states with uniform probability.
- **Initial seed set is more fit to describe police chases**.
- See biggest improvement over baseline on police videos, least improvement on children videos.
- Gigaword corpus = newswire articles
- **Underlying corpus is biased towards police chases (i.e., news-worthy events).**

Effectiveness of Coreference Resolution

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

Comparing the average CWSA-F1 scores of a naïve windowing model, under different window sizes, to the performance of the *coref* model. The *p*-values, based on the average differences, were obtained using one-tailed nonparametric bootstrap resampling with 10,000 iterations.

win-n extends the single sentence boundary of *sentence* to also include the *n* preceeding and *n* following sentences, while also piecing all relevant sentences of a document together to generate 1 neighborhood per document.

Effectiveness of Coreference Resolution

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

- *coref* outperformed all tested windowing configurations.
- Improvement over *win-1* is not significant.
  - *coref* and *win-1* generate very similar neighborhoods (extracted roughly the same number of sentences relevant to *chase*).
- ***coref* does not do worse + provides references to participants for downstream process.**

Models	Total Sentences
win-0	90,399
win-1	260,423
coref	281,666
win-2	418,827
win-3	567,706

Ensemble Models

	F_1			CWSA-F_1		
	p	r	f_1	p	r	f_1
vector	.226	.145	.175	.399	.392	.393
sentence	.194	.293	.227	.366	.376	.368
sentence+vector	.192	.377	.250	.434	.444	.438
coref	.264	.251	.253	.382	.461	.416
coref+vector	.231	.337	.271	.448	.481	.462
event	.231	.303	.256	.446	.488	.463
event+vector	.259	.296	.274	.488	.517	.500

- Combine a deleted interpolation model (text space) with vector model (latent space) creates an ensemble model.
- Every ensemble model outperformed its respective individual components.
- **Information gained from operating on text and operating in the latent vector space are highly complementary.**
  - Improvement to each will improve the resulting ensemble model.