

Whodunnit? Crime Drama as a Case for Natural Language Understanding

Lea Frermann Shay B. Cohen Mirella Lapata

Institute for Language, Cognition and Computation

School of Informatics, University of Edinburgh

10 Crichton Street, Edinburgh EH8 9AB

`l.frermann@ed.ac.uk scohen@inf.ed.ac.uk mlap@inf.ed.ac.uk`

Abstract

In this paper we argue that crime drama exemplified in television programs such as *CSI: Crime Scene Investigation* is an ideal testbed for approximating real-world natural language understanding and the complex inferences associated with it. We propose to treat crime drama as a new inference task, capitalizing on the fact that each episode poses the same basic question (i.e., who committed the crime) and naturally provides the answer when the perpetrator is revealed. We develop a new dataset¹ based on CSI episodes, formalize perpetrator identification as a sequence labeling problem, and develop an LSTM-based model which learns from multi-modal data. Experimental results show that an incremental inference strategy is key to making accurate guesses as well as learning from representations fusing textual, visual, and acoustic input.

1 Introduction

The success of neural networks in a variety of applications (Sutskever et al., 2014; Vinyals et al., 2015) and the creation of large-scale datasets have played a critical role in advancing machine understanding of natural language on its own or together with other modalities. The problem has assumed several guises in the literature such as reading comprehension (Richardson et al., 2013; Rajpurkar et al., 2016), recognizing textual entailment (Bowman et al., 2015; Rocktäschel et al., 2016), and notably

question answering based on text (Hermann et al., 2015; Weston et al., 2015), images (Antol et al., 2015), or video (Tapaswi et al., 2016).

In order to make the problem tractable and amenable to computational modeling, existing approaches study isolated aspects of natural language understanding. For example, it is assumed that understanding is an offline process, models are expected to digest large amounts of data before being able to answer a question, or make inferences. They are typically exposed to non-conversational texts or still images when focusing on the visual modality, ignoring the fact that understanding is situated in time and space and involves interactions between speakers. In this work we relax some of these simplifications by advocating a new task for natural language understanding which is multi-modal, exhibits spoken conversation, and is incremental, i.e., unfolds sequentially in time.

Specifically, we argue that crime drama exemplified in television programs such as *CSI: Crime Scene Investigation* can be used to approximate real-world natural language understanding and the complex inferences associated with it. CSI revolves around a team of forensic investigators trained to solve criminal cases by scouring the crime scene, collecting irrefutable evidence, and finding the missing pieces that solve the mystery. Each episode poses the same “whodunnit” question and naturally provides the answer when the perpetrator is revealed. Speculation about the identity of the perpetrator is an integral part of watching CSI and an incremental process: viewers revise their hypotheses based on new evidence gathered around the suspect/s or on new in-

¹Our dataset is available at <https://github.com/EdinburghNLP/csi-corpus>.

ferences which they make as the episode evolves.

We formalize the task of identifying the perpetrator in a crime series as a sequence labeling problem. Like humans watching an episode, we assume the model is presented with a sequence of inputs comprising information from different modalities such as text, video, or audio (see Section 4 for details). The model predicts for each input whether the perpetrator is mentioned or not. Our formulation generalizes over episodes and crime series. It is not specific to the identity and number of persons committing the crime as well as the type of police drama under consideration. Advantageously, it is *incremental*, we can track model predictions from the beginning of the episode and examine its behavior, e.g., how often it changes its mind, whether it is consistent in its predictions, and when the perpetrator is identified.

We develop a new dataset based on 39 CSI episodes which contains goldstandard perpetrator mentions as well as viewers’ guesses about the perpetrator while each episode unfolds. The sequential nature of the inference task lends itself naturally to recurrent network modeling. We adopt a generic architecture which combines a one-directional long-short term memory network (Hochreiter and Schmidhuber, 1997) with a softmax output layer over binary labels indicating whether the perpetrator is mentioned. Based on this architecture, we investigate the following questions:

1. What type of knowledge is necessary for performing the perpetrator inference task? Is the textual modality sufficient or do other modalities (i.e., visual and auditory input) also play a role?
2. What type of inference strategy is appropriate? In other words, does access to past information matter for making accurate inferences?
3. To what extent does model behavior simulate humans? Does performance improve over time and how much of an episode does the model need to process in order to make accurate guesses?

Experimental results on our new dataset reveal that multi-modal representations are essential for the

task at hand boding well with real-world natural language understanding. We also show that an incremental inference strategy is key to guessing the perpetrator accurately although the model tends to be less consistent compared to humans. In the remainder, we first discuss related work (Section 2), then present our dataset (Section 3) and formalize the modeling problem (Section 4). We describe our experiments in Section 5.

2 Related Work

Our research has connections to several lines of work in natural language processing, computer vision, and more generally multi-modal learning. We review related literature in these areas below.

Language Grounding Recent years have seen increased interest in the problem of grounding language in the physical world. Various semantic space models have been proposed which learn the meaning of words based on linguistic and visual or acoustic input (Bruni et al., 2014; Silberer et al., 2016; Lazaridou et al., 2015; Kiela and Bottou, 2014). A variety of cross-modal methods which fuse techniques from image and text processing have also been applied to the tasks of generating image descriptions and retrieving images given a natural language query (Vinyals et al., 2015; Xu et al., 2015; Karpathy and Fei-Fei, 2015). Another strand of research focuses on how to explicitly encode the underlying semantics of images making use of structural representations (Ortiz et al., 2015; Elliott and Keller, 2013; Yatskar et al., 2016; Johnson et al., 2015). Our work shares the common goal of grounding language in additional modalities. Our model is, however, not static, it learns representations which evolve over time.

Video Understanding Work on video understanding has assumed several guises such as generating descriptions for video clips (Venugopalan et al., 2015a; Venugopalan et al., 2015b), retrieving video clips with natural language queries (Lin et al., 2014), learning actions in video (Bojanowski et al., 2013), and tracking characters (Sivic et al., 2009). Movies have also been aligned to screenplays (Cour et al., 2008), plot synopses (Tapaswi et al., 2015), and books (Zhu et al., 2015) with the aim of improv-

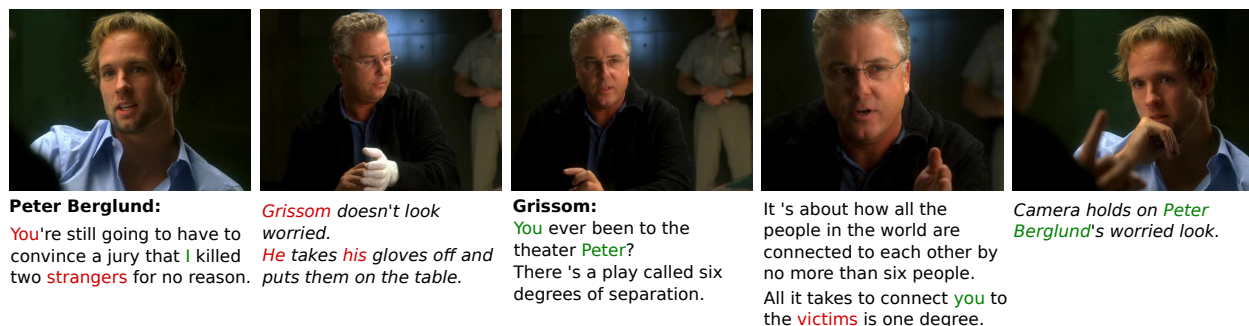


Figure 1: Excerpt from a CSI script (Episode 03, Season 03: “Let the Seller Beware”). Speakers are shown in bold, spoken dialog in normal font, and scene descriptions in italics. Gold-standard entity mention annotations are in color. Perpetrator mentions (e.g., Peter Berglund) are in green, while words referring to other entities are in red.

ing scene prediction and semantic browsing. Other work uses low-level features (e.g., based on face detection) to establish social networks of main characters in order to summarize movies or perform genre classification (Rasheed et al., 2005; Sang and Xu, 2010; Dimitrova et al., 2000). Although visual features are used mostly in isolation, in some cases they are combined with audio in order to perform video segmentation (Boreczky and Wilcox, 1998) or semantic movie indexing (Naphide and Huang, 2001).

A few datasets have been released recently which include movies and textual data. MovieQA (Tapaswi et al., 2016) is a large-scale dataset which contains 408 movies and 14,944 questions, each accompanied with five candidate answers, one of which is correct. For some movies, the dataset also contains subtitles, video clips, scripts, plots, and text from the Described Video Service (DVS), a narration service for the visually impaired. MovieDescription (Rohrbach et al., 2017) is a related dataset which contains sentences aligned to video clips from 200 movies. Scriptbase (Gorinski and Lapata, 2015) is another movie database which consists of movie screenplays (without video) and has been used to generate script summaries.

In contrast to the story comprehension tasks envisaged in MovieQA and MovieDescription, we focus on a single cinematic genre (i.e., crime series), and have access to *entire* episodes (and their corresponding screenplays) as opposed to video-clips or DVSs for some of the data. Rather than answering multiple factoid questions, we aim to solve a *single* problem, albeit one that is inherently challenging to both humans and machines.

Question Answering A variety of question answering tasks (and datasets) have risen in popularity in recent years. Examples include reading comprehension, i.e., reading text and answering questions about it (Richardson et al., 2013; Rajpurkar et al., 2016), open-domain question answering, i.e., finding the answer to a question from a large collection of documents (Voorhees and Tice, 2000; Yang et al., 2015), and cloze question completion, i.e., predicting a blanked-out word of a sentence (Hill et al., 2015; Hermann et al., 2015). Visual question answering (VQA; Antol et al. (2015)) is another related task where the aim is to provide a natural language answer to a question about an image.

Our inference task can be viewed as a form of question answering over multi-modal data, focusing on one type of question. Compared to previous work on machine reading or visual question answering, we are interested in the temporal characteristics of the inference process, and study how understanding evolves incrementally with the contribution of various modalities (text, audio, video). Importantly, our formulation of the inference task as a sequence labeling problem departs from conventional question answering allowing us to study how humans and models alike make decisions over time.

3 The CSI Dataset

In this work, we make use of episodes of the U.S. TV show “Crime Scene Investigation Las Vegas” (henceforth CSI), one of the most successful crime series ever made. Fifteen seasons with a total of 337 episodes were produced over the course of fifteen

years. CSI is a procedural crime series, it follows a team of investigators employed by the Las Vegas Police Department as they collect and evaluate evidence to solve murders, combining forensic police work with the investigation of suspects.

We paired official CSI videos (from seasons 1–5) with screenplays which we downloaded from a website hosting TV show transcripts.² Our dataset comprises 39 CSI episodes, each approximately 43 minutes long. Episodes follow a regular plot, they begin with the display of a crime (typically without revealing the perpetrator) or a crime scene. A team of five recurring police investigators attempt to reconstruct the crime and find the perpetrator. During the investigation, multiple (innocent) suspects emerge, while the crime is often committed by a single person, who is eventually identified and convicted. Some CSI episodes may feature two or more unrelated cases. At the beginning of the episode the CSI team is split and each investigator is assigned a single case. The episode then alternates between scenes covering each case, and the stories typically do not overlap. Figure 1 displays a small excerpt from a CSI screenplay. Readers unfamiliar with script writing conventions should note that scripts typically consist of scenes, which have headings indicating where the scene is shot (e.g., inside someone’s house). Character cues preface the lines the actors speak (see boldface in Figure 1), and scene descriptions explain what the camera sees (see second and fifth panel in Figure 1).

Screenplays were further synchronized with the video using closed captions which are time-stamped and provided in the form of subtitles as part of the video data. The alignment between screenplay and closed captions is non-trivial, since the latter only contain dialogue, omitting speaker information or scene descriptions. We first used dynamic time warping (DTW; Myers and Rabiner (1981)) to approximately align closed captions with the dialogue in the scripts. And then heuristically time-stamped remaining elements of the screenplay (e.g., scene descriptions), allocating them to time spans between spoken utterances. Table 1 shows some descriptive statistics on our dataset, featuring the number of cases per episode, its length (in terms of number

episodes with one case		19		
episodes with two cases		20		
total number of cases		59		
per case		min	max	avg
	sentences	228	1209	689
	sentences with perpetrator	0	267	89
	scene descriptions	64	538	245
	spoken utterances	144	778	444
	characters	8	38	20
type of crime	murder	51		
	accident	4		
	suicide	2		
	other	2		

Table 1: Statistics on the CSI data set. The type of crime was identified by our annotators via a multiple-choice questionnaire (which included the option “other”). Note that accidents may also involve perpetrators.

of sentences), the type of crime, among other information.

The data was further annotated, with two goals in mind. Firstly, in order to capture the characteristics of the *human inference* process, we recorded how participants incrementally update their beliefs about the perpetrator. Secondly, we collected gold-standard labels indicating whether the perpetrator is mentioned. Specifically, while a participant watches an episode, we record their guesses about who the perpetrator is (Section 3.1). Once the episode is finished and the perpetrator is revealed, the same participant annotates entities in the screenplay referring to the *true* perpetrator (Section 3.2).

3.1 Eliciting Behavioral Data


All annotations were collected through a web-interface. We recruited three annotators, all post-graduate students and proficient in English, none of them regular CSI viewers. We obtained annotations for 39 episodes (comprising 59 cases).

A snapshot of the annotation interface is presented in Figure 2. The top of the interface provides a short description of the episode, i.e., in the form of a one-sentence summary (carefully designed to not give away any clues about the perpetrator). Summaries were adapted from the CSI season summaries available in Wikipedia.³ The annotator watches the

²<http://transcripts.foreverdreaming.org/>

³See e.g., [https://en.wikipedia.org/wiki/CSI:_Crime_Scene_Investigation_\(season_1\)](https://en.wikipedia.org/wiki/CSI:_Crime_Scene_Investigation_(season_1)).

Number of cases: 2
Case 1: Grissom, Catherine, Nick and Warrick investigate when a wealthy couple is murdered at their house.
Case 2: Meanwhile Sara is sent to a local high school where a cheerleader was found eviscerated on the football field.



Screenplay	Perpetrator mentioned?	Relates to case 1/2/none?
(Nick cuts the canopy around MONICA NEWMAN.)	<input type="checkbox"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>
Nick okay, Warrick, hit it	<input type="checkbox"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>
(WARRICK starts the crane support under the awning to remove the body and the canopy area that NICK cut.)	<input type="checkbox"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>
Nick white female, multiple bruising ... bullet hole to the temple doesn't help	<input type="checkbox"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>
Nick .380 auto on the side	<input type="checkbox"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>
Warrick yeah, somebody man-handled her pretty good before they killed her	<input type="checkbox"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>

Figure 2: Annotation interface (first pass): after watching three minutes of the episode, the annotator indicates whether she believes the perpetrator has been mentioned.

episode (i.e., the video without closed captions) as a sequence of three minute intervals. Every three minutes, the video halts, and the annotator is presented with the screenplay corresponding to the part of the episode they have just watched. While reading through the screenplay, they must indicate for every sentence whether they believe the perpetrator is mentioned. This way, we are able to monitor how humans create and discard hypotheses about perpetrators *incrementally*. As mentioned earlier, some episodes may feature more than one case. Annotators signal for each sentence, which case it belongs to or whether it is irrelevant (see the radio buttons in Figure 2). In order to obtain a more fine-grained picture of the human guesses, annotators are addi-

	(It	's	a	shell casing	.)
Perpetrator	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suspect	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	GRISSOM	moves	his	light	to	the	canopy	below
Perpetrator	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suspect	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 3: Annotation interface (second pass): after watching the episode, the annotator indicates for each word whether it refers to the perpetrator.

tionally asked to press a large red button (below the video screen) as soon as they “think they know who the perpetrator is”, i.e., at any time while they are watching the video. They are allowed to press the button multiple times throughout the episode in case they change their mind.

Even though the annotation task just described reflects individual rather than gold-standard behavior, we report inter-annotator agreement (IAA) as a means of estimating variance amongst participants. We computed IAA using Cohen’s (1960) Kappa based on three episodes annotated by two participants. Overall agreement on this task (second column in Figure 2) is 0.74. We also measured percent agreement on the minority class (i.e., sentences tagged as “perpetrator mentioned”) and found it to be reasonably good at 0.62, indicating that despite individual differences, the process of guessing the perpetrator is broadly comparable across participants. Finally, annotators had no trouble distinguishing which utterances refer to which case (when the episode revolves around several), achieving an IAA of $\kappa = 0.96$.

3.2 Gold Standard Mention Annotation

After watching the entire episode, the annotator reads through the screenplay for a second time, and tags entity mentions, now knowing the perpetrator. Each word in the script has three radio buttons attached to it, and the annotator selects one only if a word refers to a perpetrator, a suspect, or a character who falls into neither of these classes (e.g., a police investigator or a victim). For the majority of words, no button will be selected. A snapshot of our interface for this second layer of annotations is shown in Figure 3. To ensure consistency, annota-

tors were given detailed guidelines about what constitutes an entity. Examples include proper names and their titles (e.g., *Mr Collins*, *Sgt. O’ Reilly*), pronouns (e.g., *he*, *we*), and other referring expressions including nominal mentions (e.g., *let’s arrest the guy with the black hat*).

Inter-annotator agreement based on three episodes and two annotators was $\kappa = 0.90$ on the perpetrator class and $\kappa = 0.89$ on other entity annotations (grouping together suspects with other entities). Percent agreement was 0.824 for perpetrators and 0.823 for other entities. The high agreement indicates that the task is well-defined and the elicited annotations reliable. After the second pass, various entities in the script are disambiguated in terms of whether they refer to the perpetrator or other individuals.

Note that in this work we do not use the token-level gold standard annotations directly. Our model is trained on sentence-level annotations which we obtain from token-level ones, under the assumption that a sentence mentions the perpetrator if it contains a token that does.

4 Model Description

We formalize the problem of identifying the perpetrator in a crime series episode as a sequence labeling task. Like humans watching an episode, our model is presented with a sequence of (possibly multi-modal) inputs, each corresponding to a sentence in the script, and assigns a label l indicating whether the perpetrator is mentioned in the sentence ($l = 1$) or not ($l = 0$). The model is fully incremental, each labeling decision is based solely on information derived from *previously* seen inputs.

We could have formalized our inference task as a multi-label classification problem where labels correspond to characters in the script. Although perhaps more intuitive, the multi-class framework results in an output label space different for each episode which renders comparison of model performance across episodes problematic. In contrast, our formulation has the advantage of being directly applicable to any episode or indeed any crime series.

A sketch of our inference task is shown in Figure 4. The core of our model (see Figure 5) is a one-directional long-short term memory net-

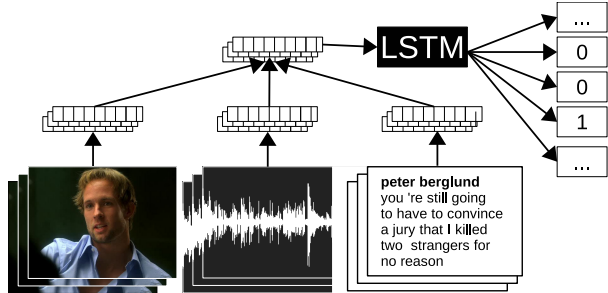


Figure 4: Overview of the perpetrator prediction task. The model receives input in the form of text, images, and audio. Each modality is mapped to a feature representation. Feature representations are fused and passed to an LSTM which predicts whether a perpetrator is mentioned (label $l = 1$) or not ($l = 0$).

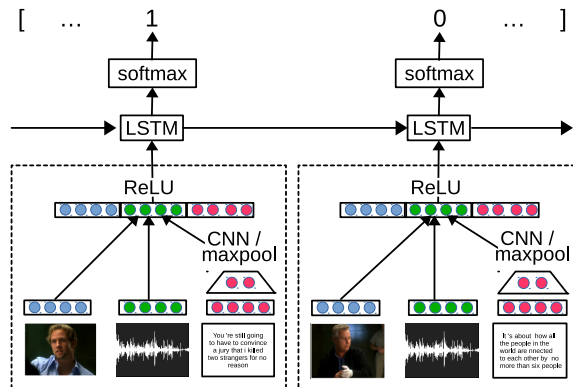


Figure 5: Illustration of input/output structure of our LSTM model for two time steps.

work (LSTM; Hochreiter and Schmidhuber (1997; Zaremba et al. (2014))). LSTM cells are a variant of recurrent neural networks with a more complex computational unit which have emerged as a popular architecture due to their representational power and effectiveness at capturing long-term dependencies. LSTMs provide ways to selectively store and forget aspects of previously seen inputs, and as a consequence can memorize information over longer time periods. Through input, output and forget gates, they can flexibly regulate the extent to which inputs are stored, used, and forgotten.

The LSTM processes a sequence of (possibly multi-modal) inputs $s = \{x_1^h, x_2^h, \dots, x_N^h\}$. It utilizes a memory slot c_t and a hidden state h_t which are incrementally updated at each time step t . Given input x_t , the previous latent state h_{t-1} and previous memory state c_{t-1} , the latent state h_t for time t and the

updated memory state c_t , are computed as follows:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \hat{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t$$

$$h_t = o_t \odot \tanh(c_t).$$

The weight matrix W is estimated during inference, and i , o , and f are memory gates.

As mentioned earlier, the input to our model consists of a sequence of sentences, either spoken utterances or scene descriptions (we do not use speaker information). We further augment textual input with multi-modal information obtained from the alignment of screenplays to video (see Section 3).

Textual modality Words in each sentence are mapped to 50-dimensional GloVe embeddings, pre-trained on Wikipedia and Gigaword (Pennington et al., 2014). Word embeddings are subsequently concatenated and padded to the maximum sentence length observed in our data set in order to obtain fixed-length input vectors. The resulting vector is passed through a convolutional layer with max-pooling to obtain a sentence-level representation x^s . Word embeddings are fine-tuned during training.

Visual modality We obtain the video corresponding to the time span covered by each sentence and sample one frame per sentence from the center of the associated period.⁴ We then map each frame to a 1,536-dimensional visual feature vector x^v using the final hidden layer of a pre-trained convolutional network which was optimized for object classification (inception-v4; Szegedy et al. (2016)).

Acoustic modality For each sentence, we extract the audio track from the video which includes all sounds and background music but no spoken dialog. We then obtain Mel-frequency cepstral coefficient (MFCC) features from the continuous signal. MFCC features were originally developed in the context of speech recognition (Davis and Mer-melstein, 1990; Sahidullah and Saha, 2012), but

⁴We also experimented with multiple frames per sentence but did not observe any improvement in performance.

have also been shown to work well for more general sound classification (Chachada and Kuo, 2014). We extract a 13-dimensional MFCC feature vector for every five milliseconds in the video. For each input sentence, we sample five MFCC feature vectors from its associated time interval, and concatenate them in chronological order into the acoustic input x^a .⁵

Modality Fusion Our model learns to fuse multi-modal input as part of its overall architecture. We use a general method to obtain any combination of input modalities (i.e., not necessarily all three). Single modality inputs are concatenated into an m -dimensional vector (where m is the sum of dimensionalities of all the input modalities). We then multiply this vector with a weight matrix W^h of dimension $m \times n$, add an m -dimensional bias b^h , and pass the result through a rectified linear unit (ReLU):

$$x^h = \text{ReLU}([x^s; x^v; x^a]W^h + b^h)$$

The resulting multi-modal representation x^h is of dimension n and passed to the LSTM (see Figure 5).

5 Evaluation

In our experiments we investigate what type of *knowledge* and *strategy* are necessary for identifying the perpetrator in a CSI episode. In order to shed light on the former question we compare variants of our model with access to information from different modalities. We examine different inference strategies by comparing the LSTM to three baselines. The first one lacks the ability to flexibly fuse multi-modal information (a CRF), while the second one does not have a notion of history, classifying inputs independently (a multilayer perceptron). Our third baseline is a rule-base system that neither uses multi-modal inputs nor has a notion of history. We also compare the LSTM to humans watching CSI. Before we report our results, we describe our setup and comparison models in more detail.

5.1 Experimental Settings

Our CSI data consists of 39 episodes giving rise to 59 cases (see Table 1). The model was trained on

⁵Preliminary experiments showed that concatenation outperforms averaging or relying on a single feature vector.

53 cases using cross-validation (five splits with 47/6 training/test cases). The remaining 6 cases were used as truly held-out test data for final evaluation.

We trained our model using ADAM with stochastic gradient-descent and mini-batches of six episodes. Weights were initialized randomly, except for word embeddings which were initialized with pre-trained 50-dimensional GloVe vectors (Pennington et al., 2014), and fine-tuned during training. We trained our networks for 100 epochs and report the best result obtained during training. All results are averages of five runs of the network. Parameters were optimized using two cross-validation splits.

The sentence convolution layer has three filters of sizes 3, 4, 5 each of which after convolution returns 75-dimensional output. The final sentence representation x^s is obtained by concatenating the output of the three filters and is of dimension 225. We set the size of the hidden representation of merged cross-modal inputs x^h to 300. The LSTM has one layer with 128 nodes. We set the learning rate to 0.001 and apply dropout with probability of 0.5.

We compared model output against the gold standard of perpetrator mentions which we collected as part of our annotation effort (second pass).

5.2 Model Comparison

CRF Conditional Random Fields (Lafferty et al., 2001) are probabilistic graphical models for sequence labeling. The comparison allows us to examine whether the LSTM’s use of long-term memory and (non-linear) feature integration is beneficial for sequence prediction. We experimented with a variety of features for the CRF, and obtained best results when the input sentence is represented by concatenated word embeddings.

MLP We also compared the LSTM against a multi-layer perceptron with two hidden layers, and a softmax output layer. We replaced the LSTM in our overall network structure with the MLP, keeping the methodology for sentence convolution and modality fusion and all associated parameters fixed to the values described in Section 5.1. The hidden layers of the MLP have ReLU activations and layer-size of 128, as in the LSTM. We set the learning rate to 0.0001. The MLP makes independent predictions for each element in the sequence. This comparison

Model	Modality			Cross-val			Held-out		
	T	V	A	pr	re	f1	pr	re	f1
PRO	+	-	-	19.3	76.3	31.6	19.5	77.2	31.1
CRF	+	-	-	33.1	15.4	20.5	30.2	16.1	21.0
MLP	+	-	-	36.7	32.5	33.7	35.9	36.8	36.3
	+	+	-	37.4	35.1	35.1	38.0	41.0	39.3
	+	-	+	39.6	34.2	35.7	38.7	36.5	37.5
	+	+	+	38.4	34.6	35.7	38.5	42.3	40.2
LSTM	+	-	-	39.2	45.7	41.3	36.9	50.4	42.3
	+	+	-	39.9	48.3	43.1	40.9	54.9	46.8
	+	-	+	39.2	52.0	44.0	36.8	56.3	44.5
	+	+	+	40.6	49.7	44.1	42.8	51.2	46.6
Humans				74.1	49.4	58.2	76.3	60.2	67.3

Table 2: Precision (pr) recall (re) and f1 for detecting the minority class (perpetrator mentioned) for humans (bottom) and various systems. We report results with cross-validation (center) and on a held-out data set (right) using the textual (T) visual (V), and auditory (A) modalities.

sheds light on the importance of *sequential* information for perpetrator identification task. All results are best checkpoints over 100 training epochs, averaged over five runs.

PRO Aside from the supervised models described so far, we developed a simple rule-based system which does not require access to labeled data. The system defaults to the perpetrator class for any sentence containing a personal (e.g., *you*), possessive (e.g., *mine*) or reflexive pronoun (e.g., *ourselves*). In other words, it assumes that every pronoun refers to the perpetrator. Pronoun mentions were identified using string-matching and a precompiled list of 31 pronouns. This system cannot incorporate any acoustic or visual data.

Human Upper Bound Finally, we compared model performance against humans. In our annotation task (Section 3.1), participants annotate sentences incrementally, while watching an episode for the first time. The annotations express their *belief* as to whether the perpetrator is mentioned. We evaluate these first-pass guesses against the gold standard (obtained in the second-pass annotation).

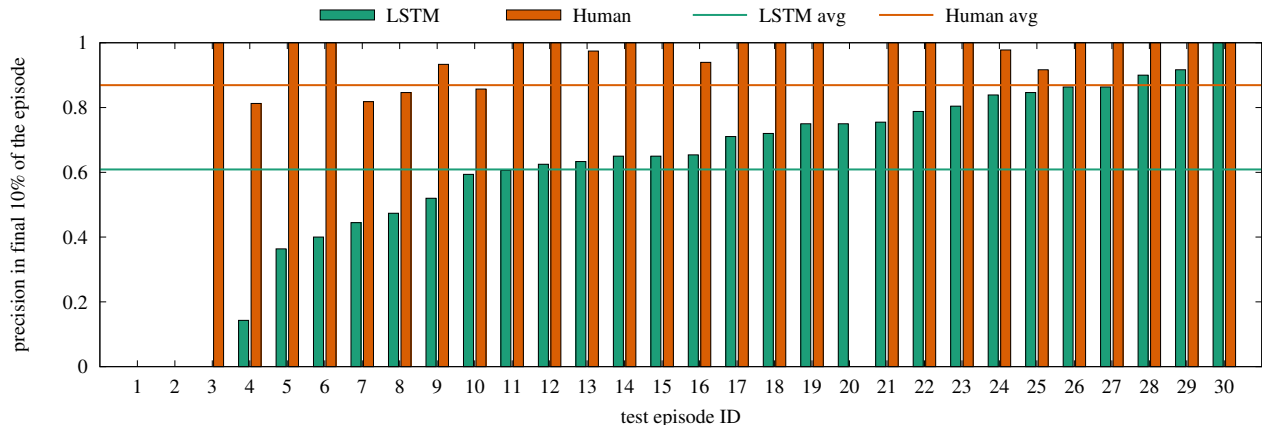


Figure 6: Precision in the final 10% of an episode, for 30 test episodes from five cross-validation splits. We show scores per episode and global averages (horizontal bars). Episodes are ordered by increasing model precision.

5.3 Which Model Is the Best Detective?

We report precision, recall and f1 on the minority class, focusing on how accurately the models identify perpetrator mentions. Table 2 summarizes our results, averaged across five cross-validation splits (left) and on the truly held-out test episodes (right).

Overall, we observe that humans outperform all comparison models. In particular, human precision is superior, whereas recall is comparable, with the exception of PRO which has high recall (at the expense of precision) since it assumes that all pronouns refer to perpetrators. We analyze the differences between model and human behavior in more detail in Section 5.5. With regard to the LSTM, both visual and acoustic modalities bring improvements over the textual modality, however, their contribution appears to be complementary. We also experimented with acoustic and visual features on their own, but without high-level textual information, the LSTM converges towards predicting the majority class only. Results on the held-out test set reveal that our model generalizes well to unseen episodes, despite being trained on a relatively small data sample compared to standards in deep learning.

The LSTM consistently outperforms the non-incremental MLP. This shows that the ability to utilize information from previous inputs is essential for this task. This is intuitively plausible; in order to identify the perpetrator, viewers must be aware of the plot’s development and make inferences while the episode evolves. The CRF is outperformed by

all other systems, including rule-based PRO. In contrast to the MLP and PRO, the CRF utilizes sequential information, but cannot flexibly fuse information from different modalities or exploit non-linear mappings like neural models. The only type of input which enabled the CRF to predict perpetrator mentions were concatenated word embeddings (see Table 2). We trained CRFs on audio or visual features, together with word embeddings, however these models converged to only predicting the majority class. This suggests that CRFs do not have the capacity to model complex long sequences and draw meaningful inferences based on them. PRO achieves a reasonable f1 score but does so because it achieves high recall at the expense of very low precision. The precision-recall tradeoff is much more balanced for the neural systems.

5.4 Can the Model Identify the Perpetrator?

In this section we assess more directly how the LSTM compares against humans when asked to identify the perpetrator by the end of a CSI episode. Specifically, we measure precision in the final 10% of an episode, and compare human performance (first-pass guesses) and an LSTM model which uses all three modalities. Figure 6 shows precision results for 30 test episodes (across five cross-validation splits) and average precision as horizontal bars.

Perhaps unsurprisingly, human performance is superior, however, the model achieves an average precision of 60% which is encouraging (compared to

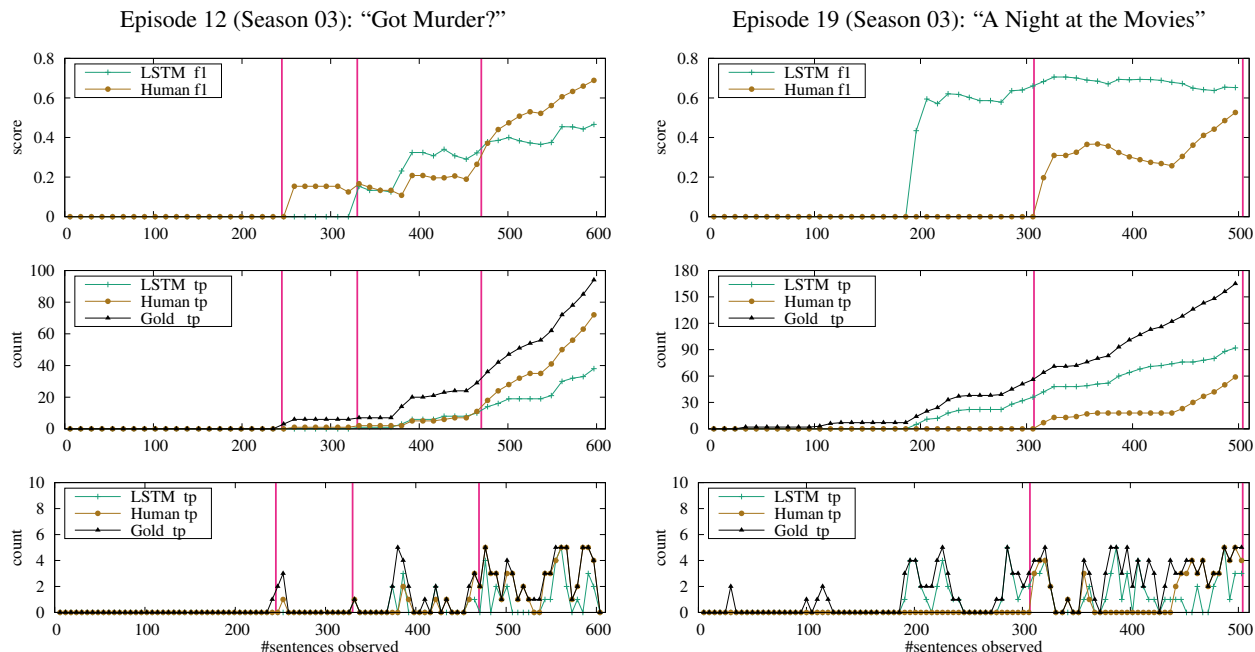


Figure 7: Human and LSTM behavior over the course of two episodes (left and right). Top plots show cumulative f1; true positives (tp) are shown cumulatively (center) and as individual counts for each interval (bottom). Statistics relating to *gold* perpetrator mentions are shown in black. Red vertical bars show when humans press the red button to indicate that they (think they) have identified the perpetrator.

85% achieved by humans). Our results also show a moderate correlation between model and humans: episodes which are difficult for the LSTM (see left side of the plot in Figure 6) also result in lower human precision. Two episodes on the very left of the plot have 0% precision and are special cases. The first one revolves around a suicide, which is not strictly speaking a crime, while the second one does not mention the perpetrator in the final 10%.

5.5 How Is the Model Guessing?

We next analyze how the model’s guessing ability compares to humans. Figure 7 tracks model behavior over the course of two episodes, across 100 equally sized intervals. We show the cumulative development of f1 (top plot), cumulative true positive counts (center plot), and true positive counts within each interval (bottom plot). Red bars indicate times at which annotators pressed the red button.

Figure 7 (right) shows that humans may outperform the LSTM in precision (but not necessarily in recall). Humans are more cautious at guessing the perpetrator: the first human guess appears around sentence 300 (see the leftmost red vertical bars in

Figure 7 right), the first model guess around sentence 190, and the first true mention around sentence 30. Once humans guess the perpetrator, however, they are very precise and consistent. Interestingly, model guesses at the start of the episode closely follow the pattern of gold-perpetrator mentions (bottom plots in Figure 7). This indicates that early model guesses are not noise, but meaningful predictions.

Further analysis of human responses is illustrated in Figure 8. For each of our three annotators we plot the points in each episode where they press the red button to indicate that they know the perpetrator (bottom). We also show the number of times (all three) annotators pressed the red button individually for each interval and cumulatively over the course of the episode. Our analysis reveals that viewers tend to press the red button more towards the end, which is not unexpected since episodes are inherently designed to obfuscate the identification of the perpetrator. Moreover, Figure 8 suggests that there are two types of viewers: *eager* viewers who like our model guess early on, change their mind often, and therefore press the red button frequently (annotator 1 pressed the red button 6.1 times on average per

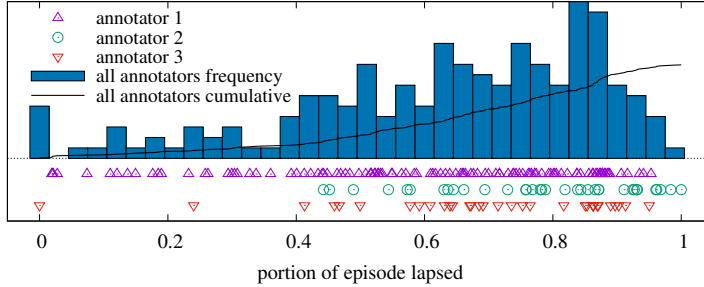


Figure 8: Number of times the red button is pressed by each annotator individually (bottom) and by all three within each time interval and cumulatively (top). Times are normalized with respect to length. Statistics are averaged across 18/12/9 cases per annotator 1/2/3.

First correct perpetrator prediction			
	min	max	avg
LSTM	2	554	141
Human	12	1014	423

Table 3: Sentence ID in the script where the LSTM and Humans predict the true perpetrator for the first time. We show the earliest (min) latest (max) and average (avg) prediction time over 30 test episodes (five cross-validation splits).

Episode 03 (Season 03): “Let the Seller Beware”								
s1	s2	s3	s4	s5	s6	s7	s8	s9
Grissom pulls out a small evidence bag with the filling	He puts it on the table	Tooth filling 0857	10-7-02	Brass We also found your fingerprints and your hair	Peter B. Look I’m sure you’ll find me all over the house	Peter B. I wanted to buy it	Peter B. I was everywhere	Brass well you made sure you were everywhere too didn’t you?

Episode 21 (Season 05): “Committed”							
s1	s2	s3	s4	s5	s6	s7	s8
Grissom What’s so amusing?	Adam Trent So let’s say you find out who did it and maybe it’s me.	Adam Trent What are you going to do?	Adam Trent Are you going to convict me of murder and put me in a bad place?	Adam smirks and starts biting his nails.	Grissom Is it you?	Adam Trent Check the files sir.	Adam Trent I’m a rapist not a murderer.

Table 4: Excerpts of CSI episodes together with model predictions. Model confidence ($p(l = 1)$) is illustrated in red, with darker shades corresponding to higher confidence. True perpetrator mentions are highlighted in blue. Top: a conversation involving the true perpetrator. Bottom: a conversation with a suspect who is not the perpetrator.

episode) and *conservative* viewers who guess only late and press the red button less frequently (on average annotator 2 pressed the red button 2.9 times per episode, and annotator 3 and 3.7 times). Notice that statistics in Figure 8 are averages across several episodes each annotator watched and thus viewer behavior is unlikely to be an artifact of individual episodes (e.g., featuring more or less suspects). Table 3 provides further evidence that the LSTM behaves more like an eager viewer. It presents the time in the episode (by sentence count) where the model correctly identifies the perpetrator for the first time. As can be seen, the minimum and average identification times are lower for the LSTM compared to human viewers.

Table 4 shows model predictions on two CSI screenplay excerpts. We illustrate the degree of the model’s belief in a perpetrator being mentioned by color intensity. True perpetrator mentions are highlighted in blue. In the first example, the model mostly identifies perpetrator mentions correctly. In the second example, it identifies seemingly plausible sentences which, however, refer to a suspect and not the true perpetrator.

5.6 What if There Is No Perpetrator?

In our experiments, we trained our model on CSI episodes which typically involve a crime, committed by a perpetrator, who is ultimately identified. How does the LSTM generalize to episodes *without*

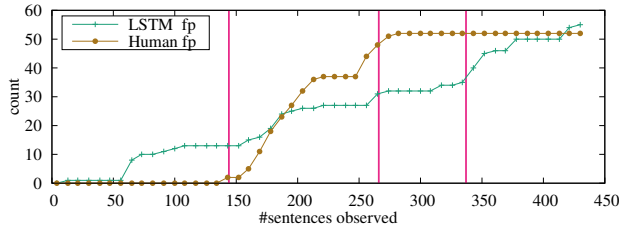


Figure 8: Cumulative counts of false positives (fp) for the LSTM and a human viewer for an episode with no perpetrator (the victim committed suicide). Red vertical bars show the times at which the viewer pressed the red button indicating that they (think they) have identified the perpetrator.

a crime, e.g., because the “victim” turns out to have committed suicide? To investigate how model and humans alike respond to atypical input we present both with an episode featuring a suicide, i.e., an episode which did not have any true positive perpetrator mentions.

Figure 8 tracks the incremental behavior of a human viewer and the model while watching the suicide episode. Both are primed by their experience with CSI episodes to identify characters in the plot as potential perpetrators, and predict false positive perpetrator mentions. The human realizes after roughly two thirds of the episode that there is no perpetrator involved (he does not annotate any subsequent sentences as “perpetrator mentioned”), whereas the LSTM continues to make perpetrator predictions until the end of the episode. The LSTM’s behavior is presumably an artifact of the recurring pattern of discussing the perpetrator in the very end of an episode.

6 Conclusions

In this paper we argued that crime drama is an ideal testbed for models of natural language understanding and their ability to draw inferences from complex, multi-modal data. The inference task is well-defined and relatively constrained: every episode poses and answers the same “whodunnit” question. We have formalized perpetrator identification as a sequence labeling problem and developed an LSTM-based model which learns incrementally from complex naturalistic data. We showed that *multi-modal* input is essential for our task as well

an *incremental* inference strategy with flexible access to previously observed information. Compared to our model, humans guess cautiously in the beginning, but are consistent in their predictions once they have a strong suspicion. The LSTM starts guessing earlier, leading to superior initial true-positive rates, however, at the cost of consistency.

There are many directions for future work. Beyond perpetrators, we may consider how suspects emerge and disappear in the course of an episode. Note that we have obtained suspect annotations but did not use them in our experiments. It should also be interesting to examine how the model behaves out-of-domain, i.e., when tested on other crime series, e.g., “Law and Order”. Finally, more detailed analysis of what happens in an episode (e.g., what actions are performed, by who, when, and where) will give rise to deeper understanding enabling applications like video summarization and skimming.

Acknowledgments The authors gratefully acknowledge the support of the European Research Council (award number 681760) and of the H2020 EU project SUMMA (award number 688139/H2020-ICT-2015). We also thank our annotators, the anonymous TACL reviewers whose feedback helped improve the present paper, and members of EdinburghNLP for helpful discussions and suggestions.

References

- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, Santiago, Chile.
- P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. 2013. Finding actors and actions in movies. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2280–2287, Sydney, Australia.
- John S. Boreczky and Lynn D. Wilcox. 1998. A hidden Markov model framework for video segmentation using audio and image features. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3741–3744, Seattle, Washington, USA.

- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal.
- Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47, January.
- Sachin Chachada and C.-C. Jay Kuo. 2014. Environmental sound recognition: A survey. *APSIPA Transactions on Signal and Information Processing*, 3.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.
- Timothee Cour, Chris Jordan, Eleni Miltsakaki, and Ben Taskar. 2008. Movie/script: Alignment and parsing of video and text transcription. In *Proceedings of the 10th European Conference on Computer Vision*, pages 158–171, Marseille, France.
- Steven B. Davis and Paul Mermelstein. 1990. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, pages 65–74. Morgan Kaufmann Publishers Inc., San Francisco, California, USA.
- Nevenka Dimitrova, Lalitha Agnihotri, and Gang Wei. 2000. Video classification based on HMM using text and faces. In *Proceedings of the 10th European Signal Processing Conference (EUSIPCO)*, pages 1–4. IEEE.
- Desmond Elliott and Frank Keller. 2013. Image description using visual dependency representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1292–1302, Seattle, Washington, USA.
- Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado, USA.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems* 28, pages 1693–1701. Curran Associates, Inc.
- Felix Hill, Anoine Bordes, Sumit Chopra, and Jason Weston. 2015. The Goldilocks principle: Reading children’s books with explicit memory representations. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, California, USA.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November.
- Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. 2015. Image Retrieval Using Scene Graphs. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, Boston, Massachusetts, USA.
- Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, Boston, Massachusetts.
- Douwe Kiela and Léon Bottou. 2014. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 36–45, Doha, Qatar.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining language and vision with a multimodal skip-gram model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163, Denver, Colorado, USA.
- Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. 2014. Visual semantic search: Retrieving videos via complex textual queries. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2657–2664, Columbus, Ohio, USA.
- C.S. Myers and L.R. Rabiner. 1981. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7):1389–1409.
- H.R. Naphide and T.S. Huang. 2001. A probabilistic framework for semantic video indexing, filtering, and retrieval. *IEEE Transactions on Multimedia*, 3(1):141–151.
- Luis Gilberto Mateos Ortiz, Clemens Wolff, and Mirella Lapata. 2015. Learning to interpret and describe abstract scenes. In *Proceedings of the 2015 NAACL: Human Language Technologies*, pages 1505–1515, Denver, Colorado, USA.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference*

- on *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, USA.
- Z. Rasheed, Y. Sheikh, and M. Shah. 2005. On the use of computable features for film classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):52–64.
- Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA.
- Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico.
- Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie description. *International Journal of Computer Vision*, 123(1):94–120.
- Md. Sahidullah and Goutam Saha. 2012. Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech Communication*, 54(4):543–565.
- Jitao Sang and Changsheng Xu. 2010. Character-based movie summarization. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 855–858, Firenze, Italy.
- Carina Silberer, Vittorio Ferrari, and Mirella Lapata. 2016. Visually grounded meaning representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99.
- Josef Sivic, Mark Everingham, and Andrew Zisserman. 2009. “who are you?” – learning person specific classifiers from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1145–1152, Miami, Florida, USA.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS’14*, pages 3104–3112, Cambridge, MA, USA. MIT Press.
- Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, inception-ResNet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.
- Makarand Tapaswi, Martin Bäumel, and Rainer Stiefelhagen. 2015. Aligning plot synopses to videos for story-based retrieval. *International Journal of Multimedia Information Retrieval*, (4):3–26.
- Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4640, Las Vegas, Nevada.
- Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. 2015a. Sequence to sequence – video to text. In *Proceedings of the 2015 International Conference on Computer Vision (ICCV)*, pages 4534–4542, Santiago, Chile.
- Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015b. Translating videos to natural language using deep recurrent neural networks. In *Proceedings the 2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT 2015)*, pages 1494–1504, Denver, Colorado, June.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164.
- Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 200–207, Athens, Greece.
- Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards AI-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2048–2057, Boston, Massachusetts, USA.
- Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal.
- Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation Recognition: Visual Semantic Role Labeling

- for Image Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5534–5542, Zurich, Switzerland.
- Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile.