Evaluating Captioning Models using Markov Logic Networks

1st Monika Shah
University of Memphis
mshah2@memphis.edu

2nd Somdeb Sarkhel
Adobe Research
sarkhel@adobe.com

3rd Deepak Venugopal
University of Memphis
dvngopal@memphis.edu

***Abstract*—Multimodal problems such as caption generation advances AI as a whole since they require integration of several key domains such as computer vision, NLP and knowledge representation. In this paper, we develop a new approach to evaluate captioning models by verifying them using Markov Logic Networks (MLNs). Specifically, we compile an MLN from training data and perform probabilistic inference to estimate uncertainty in a generated caption. To *reify* the caption, we leverage advances in Natural Language Inference (NLI) models and convert a caption into a query for the MLN. Further, we add visual context into the MLN distribution using an attention-based Multiple Instance Learning model and evaluate a caption based on this augmented distribution. We perform experiments using MSCOCO on several state-of-the-art benchmarks and show that our approach can evaluate captioning models just as effectively as methods that require human-generated captions.**

***Index Terms*—Visual Captioning, Markov Logic Networks, Attention, probabilistic theorem proving**

I. Introduction

Visual captioning has emerged as a prototypical multi-modal problem that requires integration of natural language understanding, computer vision and knowledge representation.

Typical caption evaluation methods rely on human judgement [20] or automated comparison to reference captions [14] to evaluate the quality of generated captions. However, these approaches are less scalable since in some cases they require several captions for the same image in order to measure if the generated caption is similar to human consensus. More recently, there has been a push towards techniques that do not require reference captions for evaluation. For instance, CLIPScore [6] relies on using a pre-trained deep model to measure similarity between the image and text. However, there are limitations to such approaches since the evaluation method may not be interpretable. In this paper, we propose a symbolic approach to evaluate captions. Specifically, similar to verification using automated theorem proving, here, we develop a verification method using *probabilistic theorem proving* [4] in Markov Logic Networks (MLNs) [2] - the equivalent of proving entailment in logical knowledge bases - to quantify uncertainty in generated captions. Fig. 1 illustrates our main idea. Assume that we have compiled an MLN as shown in the figure, where first-order formulas in the MLN represent relations observed in the data and the weights on formulas parameterize a probability distribution induced by the MLN. Given a test caption, we compute the likelihood of the caption in the MLN distribution. This is in general a *weighted model counting* or *discrete integration* problem where we sum probabilities over *possible worlds*. A world in this example denotes the presence/absence of key relations (such as `Hold`(Child,Bottle), `Frontof`(Child, Bottle), etc.) in a possible caption that can be generated for the image. Thus, if the generated caption occurs in high-probability worlds, it is more likely to be in agreement with the distribution of the compiled MLN and thus has smaller uncertainty. As an illustration of this, the probabilities of the worlds corresponding two example captions are visualized in Fig. 1.

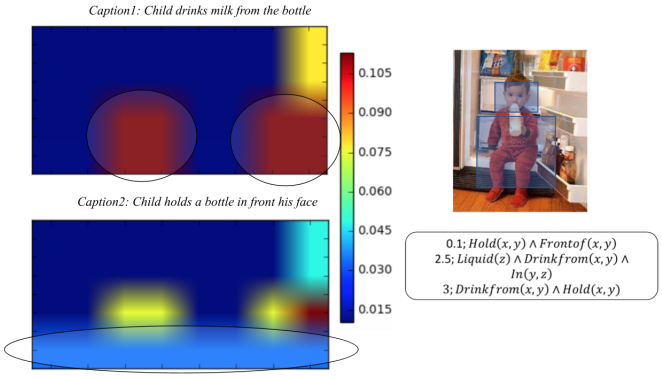


Fig. 1: An example to visualize the probabilities in the MLN distribution. The MLN for the image consists of weighted formulas. For the first caption, the atoms `Drinkfrom`(Bottle, Child) and `Liquid`(Milk) and `In`(Milk, Bottle) all follow from the caption. For the second caption, `Hold`(Child, Bottle) and `Frontof`(Child, Bottle) follow from the caption. The visualization shows the probabilities of all possible worlds in the MLN distribution (each pixel in the image corresponds to probability of a world). The marked circles show the worlds where atoms that follow from the caption are true. Summing over them yields a measure of uncertainty of the caption in the MLN distribution.

In our verification model, we compile MLNs by connecting

This research was sponsored by NSF IIS award #2008812 and NSF award #1934745. The opinions, findings, and results are solely the authors' and do not reflect those of the funding agencies.

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

triplets extracted from training data into a first-order representation which are then parameterized through Max-likelihood estimation. However, a challenging problem with compiling such an MLN for verification is that typically, the weights in an MLN are static. This is problematic since we would ideally want the MLN weights to dynamically change depending upon the visual context observed in a test image. Therefore, during verification/inference, we augment the compiled MLN based on visual features. Specifically, we learn an attention-based Deep Multiple Instance Learning (MIL) [8] model that pools lower level visual artifacts (such as object vectors) to learn a representation for higher level concepts (relationships between objects). We then augment the distribution of the MLN based on outputs of the MIL model. Thus, the distribution of the MLN dynamically changes to reflect the visual context of the relation mentions in the generated captions. To estimate the likelihood of a caption in this distribution, we *reify* the caption to represent a query in the compiled MLN. Specifically, we use a pre-trained Natural Language Inference (NLI) model that reifies the caption based on relations that entail/contradict/remain-neutral given the caption. We compute the likelihood of the reified caption as a measure of uncertainty of the captioning model.

We perform experiments using the well-known MSCOCO dataset [11] and compare the performance of state-of-the-art captioning systems such as SGAE [21] and attention-on attention transformer models (AoANet) [7]. Through detailed experiments and user studies, we demonstrate that our approach evaluates captions similar to metrics that require reference captions.

II. Background

A. Markov Logic Networks

An MLN is a set of pairs (f, w_f) where f is a formula in first-order logic and w_f is a real number. We *ground* the formulas by substituting variables with constants/objects from its *domain*. MLNs assume Herbrand semantics, i.e., there is a finite number of objects in the domain. The ground MLN represents a probability distribution over *possible worlds* (a world ω is a `True`/`False` or 0/1 assignment to all possible ground atoms in the MLN) as a *log-linear* model. Specifically,

$$P(\omega) = \frac{1}{Z} \exp\left(\sum_f w_f n_i(\omega)\right) \tag{1}$$

where $n_i(\omega)$ is the number of groundings of f that evaluate to TRUE given ω and Z is the normalization constant.

B. Related Work

The standard evaluation metrics used for captioning are typically based on comparing generated sentences to reference sentences. Since measuring semantic similarity between sentences is typically a challenging problem by itself, many of the evaluation methods such as ROGUE [17] and BLEU [14] are borrowed from NLP-evaluation in related tasks (e.g. summarization). Other methods such as METEOR [1] aim to develop metrics that are more correlated with human consensus. CIDEr [20] measures the generated sentence with human consensus on how best to describe an image. However, this requires each image to have several descriptions related to the same image in order to reliably measure consensus. Metrics that do not need human-annotated captions have also been explored recently. Madhyastha et al. [12] developed VIFIDEL, an evaluation metric that measures visual fidelity. Specifically, this metric compares a representation of the generated caption with the visual content. That is, if the caption misses out some details in the image then it is penalized and rewarded when it describes all aspects (e.g. objects) present in the image. However, one drawback with this approach is that often in images it is not necessary to represent all details, i.e., humans tend to focus on key aspects of an image to describe it. Therefore, the scores generated through VIFIDEL had lower correlations with human-based scores. Hessel et al. [6] developed an approach called CLIPScore which does not use reference captions for evaluation. For evaluation, CLIPScore presents the image and generated caption to a pre-trained cross-modal model and measures similarity between the two. Thus, the representation learned by the model helps determine the alignment between the image features and the language features in the caption. On the other hand, since our approach is symbolic, it is easier for a user to interpret our scoring. Thus, using the compiled MLN for validation does not require the step of generating a representation (using a deep network) which may not always be easy to interpret. More recently, THUMB [10] proposed rubrics for human-evaluation protocol in image captioning focused on transparency of evaluation. Our approach using symbolic AI models for evaluation is a step along this direction.

III. MLN Learning

We learn the MLN structure from captions in the training data. Specifically, we use bottom-up structure learning [13] to learn formulas from triplets (subject, object and predicate) extracted from captions using the textual scene graph parser [18]. A *gliteral* (ground literal) is a triplet extracted from the parser. We constrain connections between gliterals as follows. A connection between two gliterals exists if and only if there is an object (or constant) shared between them. For example, $\texttt{On}(Bike, Person)$ and $\texttt{On}(Street, Light)$ are non-shared gliterals, but $\texttt{On}(Bike, Person)$ and $\texttt{Near}(Person, Light)$ are connected since they share a common object. Thus, we can form a connected chain of gliterals of size k by connecting the k-th gliteral to the $k-1$-th gliteral.

We learn the MLN structure by iterating over each instance in the training data and extracting the gliteral chains of size at most k. Once we have extracted the chains from the training data, we form conjunctive first-order formulas from these gliteral chains. For example, $\texttt{On}(Bike, Person) \wedge \texttt{Ride}(Person, Bike)$ and $\texttt{On}(Horse, Person) \wedge \texttt{Ride}(Person, Horse)$ share the same first-order logic structure and the candidate MLN formula that act as a template for them is $\texttt{On}(x, y) \wedge \texttt{Ride}(y, x)$. As is

the case in MLNs, we assume Herbrand semantics where we restrict the domain of each variable to a finite set. In our case, for example, if we consider $\texttt{Ride}(y, x)$, the domain of x, Δ_x is restricted to be the set of objects that are observed in the training data, i.e., $X \in \Delta_x$ if there exists some $Y \in \Delta_y$ such that Y rides X. Note that in theory, we can use other logical connectives to connect the literals (such as $\Rightarrow$). However, it has been shown in MLNs that it is often better to use conjunctions to make weight learning more robust [2] as compared to implications that are more common in logical knowledge bases.

Once we form the set of conjunctive formulas, we learn their weights using pseudo-log likelihood (PLL) learning. Specifically, we start with a weight initialization of the ratio of the number of times a formula is satisfied by the training data and the total number of possible ground instantiations of the formula. For instance, if $\texttt{On}(x, y) \wedge \texttt{Ride}(y, x)$ is satisfied m times in the data, its weight is initialized to $m/(|\Delta_x||\Delta_y|)$ since there are a total of $|\Delta_x||\Delta_y|$ of possible groundings of the formula. We learn the weights of the formulas by maximizing the PLL which is an efficient learning approach. Specifically, let X denote all the possible ground atoms in the MLN and let x be the assignment (either 1/0) to these atoms based on the triplets in the training data. We maximize PLL of X given by the following equation.

$$logP_w(X = x) = \sum_{l=1}^{n} logP_w(X_l = x_l | MB(X_l)) \quad (2)$$

where X_l represents a single ground atom, $MB(X_l)$ is the Markov blanket of X_l, i.e., all atoms that occur in at least one formula that X_l occurs in, $X_l = x_l$ is an assignment (0/1) to the atom X_l. Thus, the PLL computes the probability of each atom conditioned on assignments to all other atoms. The weight update can be carried out efficiently since the gradient of the PLL function is given by,

$$\frac{\partial P}{\partial w_i} = \sum_{l=1}^{n} n_i(x) - P_w(X_l = 0 | MB(X_l)) n_i(x_{[X_l=0]}) - P_w(X_l = 1 | MB(X_l)) n_i(x_{[X_l=1]}) \quad (3)$$

where w_i is the weight of the i-th formula, $n_i(x)$ is the number of groundings of the i-th formula that are true (in the training data), $n_i(x_{[X_l=0]})$ is the number of groundings of the i-th formula that are true when the assignment to X_l is equal to 0 and the other assignments are unchanged. Thus, to obtain the gradient, we compute the difference between the number of true groundings and the expected number of true groundings for the current weights. To maximize the PLL, we update the i-th weight in iteration t as,

$$w_i^{(t+1)} = w_i^{(t)} + \epsilon \frac{\partial P}{\partial w_i}$$

where ϵ is the learning rate. We continue updating all the weights until we reach a fixed point. Note that since the values $n_i(x_{[X_l=0]})$, $n_i(x_{[X_l=1]})$ and $n_i(x)$ do not change as the weights are changing, they can be pre-computed in advance and therefore, PLL learning is a highly efficient weight learning approach for MLNs.

IV. Attention-based Multiple Instance Learning

One of the limitations in MLNs is that the weight for a ground formula is fixed at a single value. However, in our case, the same formula in the context of different images can have varying degrees of importance. For example, a grounding such as $\texttt{On}(Person, Bike) \wedge \texttt{Ride}(Bike, Person)$ can be very important if we are captioning an image where the rider and the vehicle are prominently seen but less important when they are obscured from view. To address this limitation, we augment the MLN with visual context from the image. Specifically, we learn an attention-based Multiple Instance Learning (MIL) model that pools information from the visual context and relates it to atoms in the MLN.

To train our model, we use relation mentions in the caption to weakly relate feature vectors extracted from the image. For example, the features corresponding to $Bike$ and $Person$ can be related through the $\texttt{Ride}$ relation if the caption mentions this relation. However, the labeling might not be exact since the object extraction can be noisy and the relation mention in the caption may be referencing other objects in the image. To address this, similar to the approaches in [8], we use weak supervision and predict a label over *bags* of instances instead of individual instances.

Let $\mathbf{O}_I$ represent the set of object feature-vectors extracted from I. In our experiments, the objects are identified and localized using Faster R-CNN [15]. ResNet-101 [5] performs object detection in the image and the visual features are extracted from this consisting of region of interest pooling for each of the bounding boxes. Let $< e, e', p >$ represent a triplet extracted from the textual scene graph parser applied to the caption, where e, e' are the object mentions and p is the predicate mention. $\vec{O} = O \oplus O'$ is an instance in the *positive bag* for p, where $O, O' \in \mathbf{O}_I$ and O has the label e, O' has the label e'. $\vec{O} = O \oplus O'$ is an instance in the *negative bag* for p, where $O, O' \in \mathbf{O}_I$ and O does not have the label e or O' does not have the label e'. As is standard practice in MIL, we assume that for a positive bag at least one instance in the bag is positive. In our case, this means that at least one of the object-pair vectors must be related by the predicate p. For a negative bag, the assumption is that no instance within the bag must be positive, i.e., none of the object-pair vectors are related through p.

Given positive and negative bags for a predicate, the MIL pooling function learns a representation for the bag. A requirement is for the pooling function to be *permutation-invariant*, i.e., the bag representation must be invariant to the order of object-pairs within the bag. The pooling function specified in [8] is shown to be effective even with a small number of bags, which fits our case since some predicates occur less frequently than others in the captions.

Let $\mathbf{O}^{(i)}$ be the i-th bag for a predicate. Let the instances in $\mathbf{O}^{(i)}$ be $O_1 \ldots O_n$. The MIL pooling function combines the representations of $O_1 \ldots O_n$ into a representation for the bag

Fig. 2: Illustrative example for attentions in identifying predicates. The left image shows the gated-attentions on object pairs and the right image show the regular attention (brighter the intensity in the box larger is the attention). For $\texttt{Hold}(boy, umbrella)$, in the gated attention model, the objects that are relevant to the predicate are attended to more than in the model that uses regular attentions.

$\mathbf{z}$. In the case of attention-based pooling, the bag representation is given by the following equation.

$$\mathbf{z} = \sum_{k=1}^{n} \alpha_k O_k$$

$$\alpha_k = \frac{\exp(\mathbf{w}^\top \tanh(\mathbf{V} O_k^\top))}{\sum_{j=1}^{n} \exp(\mathbf{w}^\top \tanh(\mathbf{V} O_j^\top))}$$

Here, $\mathbf{w} \in \mathbb{R}^{L\times 1}$ and $\mathbf{V} \in \mathbb{R}^{L\times M}$ are learnable parameters, where M is the dimensionality of the instances in the bag (the object-pair features) and L determines the flexibility we have to represent the bag. The $\tanh(\cdot)$ function is an element-wise non-linearity. In our case, we learn the parameters using a fully connected layer on top of $\mathbf{z}$ which outputs the bag label. α_k encodes the importance of the k-th object pair in determining the label for that bag. Specifically, the model has a likelihood given by,

$$L(\theta) = \prod_{i=1}^{N} P(y^{(i)} | \mathbf{O}^{(i)}, \theta)$$

where $y^{(i)}$ is the bag label for the i-th bag. The negative-log likelihood is optimized by minimizing the negative log-likelihood as,

$$\ell(\theta) = -\sum_{i=1}^{N} y^{(i)} \log(P_\theta(\mathbf{O}^{(i)}) + (1 - y^{(i)} \log(1 - P_\theta(\mathbf{O}^{(i)})$$

where $P_\theta(\mathbf{O}^{(i)})$ is the output of the attention-based model parameterized by θ. In our case, note that we learn a set of models, where the model parameterized by θ_r corresponds to the predicate of type r. The overall loss over all predicates is simply equal to $\sum_r \ell(\theta_r)$. For each predicate, we balance the negative bags with the number of positive bags when training the model. Given a new bag $\mathbf{O}$, the most likely predicate can be estimated as $r = \arg\max_{r'} P_{\theta_{r'}}(\mathbf{O})$.

A. Gated Attention

For the attention-based model, we see that the contribution from each object-pair in the bag is encoded within the operation $\tanh(\mathbf{V} O_k^\top)$. The output vector from this operation (which is an element-wise operation) is then parameterized with $\mathbf{w}$. The problem here is that, if the range of values for $\mathbf{V} O_k^\top$ is between -1 and 1, then the $\tanh(\cdot)$ function acts like a linear function. This means that, for a bag, the information passed on by each of the object-pairs within that bag will only vary linearly. In the case of complex relationships in an image, this becomes more problematic. Therefore, in [8], a gating mechanism is used to control the information that can flow out of a bag. This is similar to the approach in LSTMs where gating allows us to ensure that information does not vanish across distant time steps.

In our case, we want finer-grained control of the information the model obtains from each of the instances within a bag. Particularly, suppose in an image, we have several object-pairs that are likely to be a related by the same predicate, the gating mechanism will allow the most relevant object-pair to more significantly attend to the output bag label. Technically, as is defined in [8], this is achieved by combining a sigmoid with the tanh function to define the attention weight for each instance in a bag in MIL. That is, for each of the object-pairs in the bag, we perform an element-wise multiplication of the tanh non-linearity with the output of a sigmoid with learnable parameters. Specifically,

$$\alpha_k = \frac{\exp(\mathbf{w}^\top (\tanh(\mathbf{V} O_k^\top)) \bigodot \text{sigm}(\mathbf{U} O_k^\top))}{\sum_{j=1}^{n} \exp(\mathbf{w}^\top (\tanh(\mathbf{V} O_j^\top) \bigodot \text{sigm}(\mathbf{U} O_j^\top)))} \quad (4)$$

where $\mathbf{U} \in \mathbb{R}^{L\times M}$ are learnable parameters for the sigmoid function. In this case, for each object-pair, we compute a sigmoid non-linearity with parameters $\mathbf{V}$ and a tanh non-linearity with parameters $\mathbf{U}$ and perform an element-wise multiplication to obtain a vector representing the attention given to the object-pair. Multiplying this with $\mathbf{w}$ yields the attention value. The effect of the gating mechanism is illustrated in Fig. 2. Specifically, the gating mechanism allows more information to be passed on from each of the object pairs in a bag. As a result, relevant relationships are likely to get greater attention in the model. As seen in the figure, the relationship that is more important to the caption has more attention when we use the gating mechanism in the MIL model as compared to the regular attention model.

V. Caption Verification

A. Probabilistic Theorem Proving

Theorem Proving is the fundamental task in logical knowledge bases. The classical Davis-Putnam method uses the resolution rule to prove that a query Q is entailed by a knowledge base. However, in the presence of uncertainty, as is well-known, logical reasoning is brittle. In MLNs, we perform Probabilistic Theorem Proving (PTP) to estimate uncertainty in a query. In fact, it can be shown that if the weights of the MLN are ∞ (also called hard formulas), then PTP is equivalent to theorem proving in knowledge bases [4].

In PTP, we compute $P(Q|\mathcal{M})$, where Q is a query (logical statement) and $\mathcal{M}$ is the MLN. This is formulated as a weighted model counting problem. Specifically, for a world

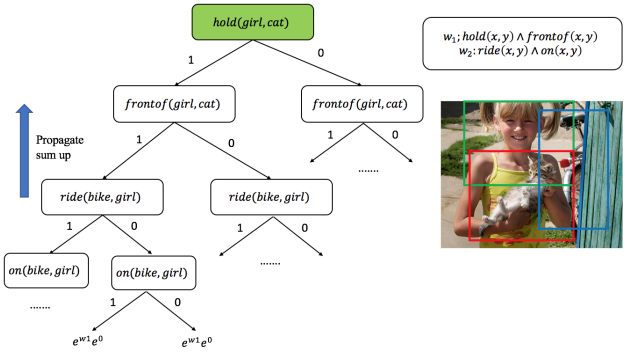


Fig. 3: Illustrating Probabilistic Theorem Proving (PTP) on an example image. We condition on each atom and each leaf represents the probability of a possible world.

ω (`True` or `False` assignment to all atoms in the MLN), we compute its weight as $\exp(\sum_f n_i(f)w_f)$, where f is a formula in $\mathcal{M}$ with weight w_f and $n_i(\omega)$ is the number of satisfied groundings of f in ω. We sum the weights over all worlds where Q is `True` to compute $Z(Q|\mathcal{M})$, similarly the weights of worlds where Q is `False` yields $Z(\neg Q|\mathcal{M})$ and we can compute the probability using the ratio, $\frac{Z(Q|\mathcal{M})}{Z(Q|\mathcal{M})+Z(\neg Q|\mathcal{M})}$. Note that the denominator is also equal to the partition function of $\mathcal{M}$, i.e., $Z(\mathcal{M})$. Fig. 3 illustrates the PTP-tree for the query marked in green. As shown here, we branch on each atom, where a branch indicates that the atom is assigned a `True` (1) or `False` (0) value. The leaf nodes evaluate the weight of a world and non-leaf nodes sum the weights propagating them upwards. Efficient strategies based on structure of the MLN such as decomposition into independent components and dynamic programming to store intermediate reusable results can be used to make the evaluation more efficient [4].

B. Reification

To apply PTP to verify a caption, we reify the caption to convert it into a logical query. Specifically, let C_I be the caption for image I. We ground the MLN $\mathcal{M}$ with image I to obtain the ground MLN $\mathcal{M}_I$ as follows. For a formula f in $\mathcal{M}$, let $\Delta_{x_1} \ldots \Delta_{x_n}$ be the domains of variables in f. We ground f with objects $X_1 \in \Delta_{x_1} \ldots X_n \in \Delta_{x_n}$ where $X_1 \ldots X_n$ are objects that are detected and localized in I. Essentially all the other ground formulas in $\mathcal{M}$ other than those in $\mathcal{M}_I$ are considered to be false for I and thus can be removed to reduce the number of groundings without affecting the MLN distribution [2]. Let $Q_1 \ldots Q_n$ be the atoms in $\mathcal{M}_I$. The reified caption is represented as a conjunction over atoms in $\mathcal{M}_I$. Specifically, we want to test if an atom in $\mathcal{M}_I$ is semantically equivalent to a relationship specified in C_I. We formulate this as a Natural Language Inference (NLI) problem. Specifically, we consider the *premise* $P = C_I$ and the *hypothesis* $H = Q_i$ and infer if H entails/contradicts/is-neutral-given P. To determine this, we use a pre-trained, state-of-the-art NLI model such as RoBERTa. Thus, the reified caption is given by $\mathbf{Q} = \wedge_{i=1}^{n} t_{Q_i}$, where t_{Q_i} is a positive literal of Q_i if Q_i entails C_I, t_{Q_i} is a negative literal of Q_i if Q_i contradicts C_I and $t_{Q_i} = \emptyset$ if Q_i is undetermined given C_I.

C. Inference

In general, computing $P(\mathbf{Q}|\mathcal{M}_I)$ exactly is intractable since to compute the normalization constant in the probability distribution, we need to sum over all possible worlds (a problem that is $\#P$-complete). Instead, we use Gibbs sampling [3] to estimate this probability. Further, we augment the distribution with outputs of the MIL model to add visual context into the distribution. Specifically, we associate each atom with a Bernoulli distribution with p equal to the probability of that atom as inferred from the MIL model. Thus, the MIL-augmented distribution becomes, $P(\mathbf{Q}|\mathcal{M}_I) \prod_{i=1}^{n} P_\theta(Q_i)$, where $P_\theta(Q_i)$ is the probability inferred for Q_i using the MIL model. To sample from this distribution, we start with an assignment to all atoms in $\mathcal{M}_I$. In each iteration, we select an atom Q and sample its assignment from the conditional distribution, $P(Q|Q_-, \mathcal{M}_I) \prod_{i=1}^{n} P_\theta(Q_i)$, where Q_- is an assignment to all atoms other than Q. This distribution is computationally easy to sample from as follows. First, we consider $Q =$ `True` and for all ground formulas in $\hat{\mathcal{M}}_I$ satisfied by the assignments (Q_-, $Q =$ `True`), we sum their exponentiated weights. For all ground formulas unsatisfied by the assignments to the atoms, we sum $exp(0)$ (i.e. they have 0 weight). The total sum yields the conditional probability $P(Q|Q_-, \mathcal{M}_I)$. We multiply this by the Bernoulli probability densities for each atom in the set $\{Q_-, Q\}$. Similarly, we compute the conditional probability for $\neg Q$ and then sample Q from its conditional distribution. We repeat this process over several iterations and it can be shown that after a period called the *burn-in* of the sampler (time to forget the starting state), we will be sampling from the target distribution. We can now estimate the $\hat{P}(\mathbf{Q}|\hat{\mathcal{M}}_I)$ by a simple Monte-Carlo estimate. That is, we count the number of samples that are consistent with $\mathbf{Q}$ and normalize it with the total number of samples. The estimate is unbiased, i.e., as we increase the number of samples the estimated $\hat{P}(\mathbf{Q}|\mathcal{M}_I) \prod_{i=1}^{n} P_\theta(Q_i)$ approaches the true distribution.

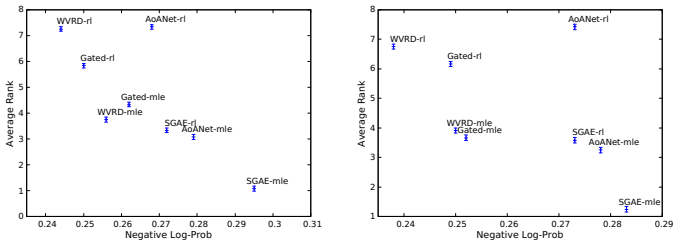
VI. Experiments

A. Setup

We used the MSCOCO image captioning benchmark dataset with Karpathy's train, test, validation split [9] in our evaluation. The training data consists of 113K images with 5K images in validation set and 5K images in test set. The number of captions per image is equal to 5. We compared the following state-of-the-art approaches in our evaluation. `SGAE` [21], `AoANet` [7] and `WeakVRD` [19]. Further, we also added another approach based on gated attentions which we denote as `Gated`. In the `Gated` model, we augment the SGAE model with the relation with maximum probability that is output from the gated attention model. In each of the captioning systems, the final captions are generated from LSTMs using two well-known standard approaches. One uses cross-entropy loss (we refer to this as `MLE`) and the other optimizes the

	Validation Set								Test Set							
	Human-Annotated Evaluation						MLN-based Evaluation		Human-Annotated Evaluation						MLN-based Evaluation	
	B1	**B4**	**ME**	**RG**	**CD**	**SP**	**Log-Prob**	**Var**	**B1**	**B4**	**ME**	**RG**	**CD**	**SP**	**Log-Prob**	**Var**
SGAE-mle	77.0	36.6	27.6	56.8	113.6	20.6	-0.295	0.094	77.4	36.6	27.7	56.9	114.5	20.8	-0.283	0.092
WVRD-mle	78.1	38.4	28.2	58.0	119.0	21.1	-0.256	0.088	78.0	37.4	28.2	58.0	119.0	21.1	-0.250	0.087
AoANet-mle	78.0	37.2	28.4	57.4	116.6	21.3	-0.279	0.092	77.2	36.9	28.4	57.2	116.6	21.6	-0.278	0.091
Gated-mle	78.0	**38.9**	28.4	57.9	116.6	21.7	-0.262	0.090	78.0	**38.9**	28.1	57.5	116.4	21.2	-0.252	0.087
SGAE-rl	79.5	36.6	27.9	57.8	122.9	21.3	-0.272	0.091	79.6	36.6	27.9	57.8	123.9	21.4	-0.273	0.090
WVRD-rl	**80.8**	**38.9**	28.8	58.7	**129.6**	22.3	**-0.244**	0.085	**80.8**	37.1	28.8	58.7	**129.6**	22.3	**-0.238**	0.084
AoANet-rl	80.2	**38.9**	**29.2**	**58.8**	127.7	**22.4**	-0.268	0.090	80.2	**38.9**	**29.2**	**58.8**	**128.9**	**22.4**	-0.273	0.089
Gated-rl	80.6	38.4	28.4	58.7	126.3	21.9	-0.250	0.087	80.7	38.6	28.4	58.7	127.0	22.0	-0.249	0.0858

TABLE I: Comparison Results for the MSCOCO dataset for Karpathy split. B1, B4, ME, RG, CD, and SP denote BLEU-1, BLEU-4, METEOR, ROUGE, CIDEr-D and SPICE scores respectively.



(a) (b)

Fig. 4: Comparing the average ranking (using 6 metrics that compare generated and human-annotated captions) of a model with the average negative log-probabilities computed from the compiled MLNs for the generated captions. Error bars indicate variance in the estimated probabilities. As we go to the top left, it indicates better model performance. (a) shows results are shown for the validation split and (b) for test split.

CiderD metric [20] within a reinforcement learning framework performed using self-critical sequence learning [16] (we refer to this as `RL`). To learn the MLN model, we use 200 predicates based on the most frequently occurring triplets in captions in the training data extracted using the textual scene graph parser [18]. In all, we learned around 6K rules in the MLN. For the gated attention-based MIL model, we used the open-source implementation from [8] with the following configuration. We set the dimensionality for the bag representation as 500 and the hidden layer size as 128. For the Gibbs-sampler, we set the burn-in as 500 samples and measured convergence through the Gelman-Rubin statistic to reach a stopping point. For caption reification, we used the open-source, pre-trained RoBERTa model for NLI. We performed our experiments on a 62.5 GiB RAM, 64-bit Intel® Core™ i9-10885H CPU @ 2.40GHz × 16 processor with a NVIDIA Quadro GPU with 16GB RAM. Our code and data is available here[1].

B. Results

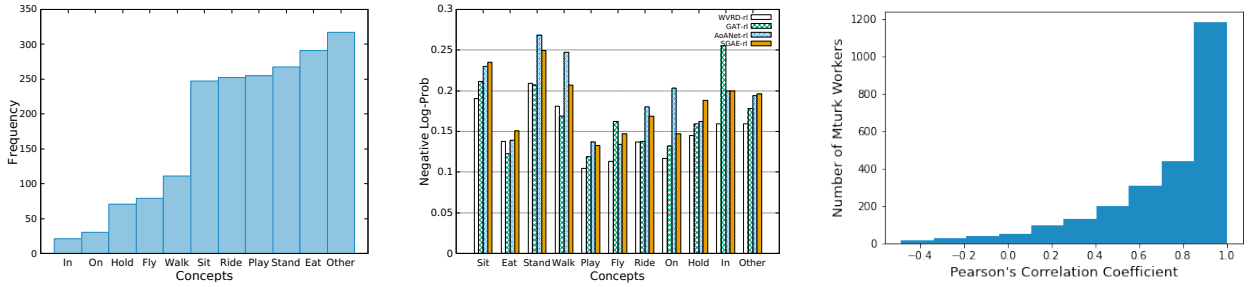
Through our evaluation, we want to answer the following questions, i) what is the correlation between metrics computed with human-labeled captions and our approach which does not require human-generated captions? ii) are there some types of concepts on which models exhibit larger/smaller uncertainty in captioning? and iii) do humans explain images in a manner that is consistent with our model?

1) Uncertainty Scores: We computed the accuracy based on 6 standard caption evaluation metrics that compare image captions written by humans with those generated by the model [21]; BLEU, METEOR, ROUGE, CIDEr-D and SPICE. For a fair evaluation, we ran each of the models in our own configuration and computed these metrics over the validation and test sets. These scores are shown in Table I. For each model, we estimate the log-probability from the Gibbs sampler. We show the average log-probability over all images in the validation and test sets along with the variance.

To get a broader view, we compute the *ranking* of a method based on its position according to each metric and then compute average of all rankings. A larger average ranking indicates that the method scored well over all metrics that require human-annotations. Fig. 4 plots this average ranking against the average negative log-probability generated by the MLN. Thus, a high-rank and low negative log-probability indicates that the model did well in both evaluations. As we see from the figure, the two evaluations are largely correlated, i.e., as the average-rank decreases, the uncertainty of the model increases as shown in Figs. 4 (a) and (b). The performance of the models over the test and validation splits are mostly similar. However, in the case of `AoANet`, the uncertainty score was lower but the average-rank was high. One of the reasons for this is the architecture of `AoANet` adds more specificity into its captions. Specifically, in one of their human evaluations, their captions were considered more descriptive than other captions. However, the added specificity also means that more general relationships may not always entail the caption. Thus the uncertainty over such captions will be higher since the distribution learned from the training captions may not encode the same level of specificity.

2) Concept Uncertainty: We asked Amazon Mechanical Turk (AMT) workers to label a concept that best describes an image. We used 3 workers for each image in the test data and instructed each worker to choose exactly one out of 10 different concepts: `Sit`, `Eat`, `Stand`, `Walk`, `Play`, `Fly`, `Ride`, `On`, `In` and `Hold`. We also added an option of choosing `Other` in case none of the concepts were perceived to be applicable to the image. We selected the dominant concept for an image based on what 2 or more workers selected (in case of a tie we inspected the image and selected the most relevant

[1]https://github.com/Monikshah/MLNCaptionVerification



(a) (b) (c)

Fig. 5: (a) shows the distribution of concepts over test images as labeled by AMT workers. (b) shows the negative log-probabilities (smaller values are better) of each of the models (we only show results for the RL variants of models here) for each of the concepts. (c) The x-axis shows the correlation between human rankings and those obtained using MLNs. The y-axis shows the number of AMT workers whose rankings had the correlations specified in the x-axis.

Fig. 6: Illustrative examples showing captions with high (red) and low (green) uncertainty. Key relations are shown above each image. In the top row of images, these relations are entailed by the caption and in the bottom row they are not entailed (shown as grey).

concept manually). The distribution of concepts is shown in Fig. 5 (a). Fig. 5 (b) shows the results obtained for images corresponding to each category. As shown here, some types of concepts have smaller uncertainty due to the nature of the concept. In particular, we observed that *play* and *eat* have the smallest uncertainty. One possible reasoning is that an action such as *sit* is generic and can also imply other actions such as read/watch/etc., whereas an action such as *eat* is much more specific and thus has smaller uncertainty.

3) Human Explanation: We ran a second AMT user study to evaluate if the MLN distribution yields probabilities that are explainable by humans. Specifically, for an image I, we showed the user a query $\mathbf{Q} = \{Q_i\}_{i=1}^{k}$, where each Q_i is an atom in the MLN grounded on I, i.e., $\mathcal{M}_I$ (we converted the atom to a natural language sentence to make it more readable). We asked the user to rank each $Q_i \in \mathbf{Q}$ by order of importance in describing/explaining the image. Thus, an atom that is less relevant to I was ranked lower than the atom that is more relevant to I. Then, we computed the probability of $\mathbf{Q}$ in $\mathcal{M}_I$, i.e., we computed $P(Q_1) \ldots P(Q_k)$ and ranked the atoms according to their probabilities, i.e., lower probability atoms were ranked lower than high probability atoms. We then measured the correlation between the two rankings to verify how closely the distribution induce by the MLN matched human interpretation of the image. We ran this study for around 1500 test images and had 2 workers perform the task

for each image (to account for noise). In Fig. 5 (c), we plot the histogram of the Person's correlation coefficients obtained when we correlate the human ranking with the probabilities. As shown in the figure, the correlations for majority of the images were quite high and only a small number of cases showed low or negative correlations. This indicates that the distribution learned is consistent with human perception.

4) Qualitative Analysis: In Fig. 6 we show example captions and for each one, we mark whether the caption has high (red) or low (green) uncertainty. We also indicate the *dominant atom*, i.e., the one that had highest probability (measured using Gibbs sampling) and whether or not that atom was entailed/not by the caption (non-entailed atoms are shown in grey color). As seen by the examples, this allows us to explain the uncertainty score assigned to the caption. As seen here, in most cases since the dominant atom captures the most important relationship in the image, captions that entail it have low uncertainty and those don't entail this have higher uncertainty. On the other hand, there are some cases where the dominant atom is not as accurate. For instance, consider the example that shows the bear on the book. The dominant atom here was not the most important aspect of this image and in this case, though the caption is a reasonable description, the uncertainty is high for this caption. This happens when the caption is *off-distribution*, i.e., different from the typical distribution that we learn from the training data. We plan to explore these cases in future work.

VII. Conclusion

In this paper, we developed a new approach to evaluate captioning models based on Markov Logic Networks (MLNs) by estimating the uncertainty in a caption based on the relationships specified in it. To do this, we reified a caption using pre-trained NLI models and performed probabilistic inference to evaluate captions from several state-of-the-art models. We showed that our results approximates evaluations that require human-generated captions. In future, we plan to extend our approach to other multi-model problems such as VQA.

References

[1] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Workshop on Statistical Machine Translation*, 2014.

[2] Pedro Domingos and Daniel Lowd. Markov logic: An interface layer for artificial intelligence. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–155, 2009.

[3] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[4] Vibhav Gogate and Pedro Domingos. Probabilistic theorem proving. *Communications of the ACM*, 59(7):107–115, 2016.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[6] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[7] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019.

[8] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *ICML*, 2018.

[9] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[10] Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah Smith. Transparent human evaluation for image captioning. In *NAACL-HLT*, 2022.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[12] Pranava Madhyastha, Josiah Wang, and Lucia Specia. VIFIDEL: evaluating the visual fidelity of image descriptions. In *ACL*, 2019.

[13] Lilyana Mihalkova and Raymond J Mooney. Bottom-up learning of markov logic network structure. In *ICML*, 2007.

[14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[16] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[17] Lin CY ROUGE. A package for automatic evaluation of summaries. In *ACL Workshop on Text Summarization*, 2004.

[18] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Workshop on vision and language*, 2015.

[19] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. In *ACL*, 2020.

[20] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[21] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019.