Enhancing Visual Question Answering with Linguistic Information

by

Mehrdad Alizadeh
B.S., Shahid Beheshti University, Tehran, 2008
M.S., Amirkabir University of Technology, Tehran, 2011

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2020

Chicago, Illinois

Defense Committee:
Barbara Di Eugenio , Chair and Advisor
Natalie Parde
Cornelia Caragea
Brian Ziebart
Ahmet Enis Cetin, UIC Dept. of Electrical & Computer Engineering

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF TABLES (Continued)

TABLE **PAGE**

LIST OF FIGURES

LIST OF FIGURES (Continued)

FIGURE **PAGE**

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
NLP	Natural Lnaguage Processing
QA	Question Answering
VQA	Visual Question Answering
IR	Information Retrieval
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
MS-COCO	Microsoft Common Objects in Context
WUPS	Wu and Palmer Set
SRL	Semantic Role Labeling
RFE	Response Frame Element

SUMMARY

Visual Question Answering (VQA) concerns providing answers to natural language questions about images. Several deep neural network approaches have been proposed to model the task in an end-to-end fashion. Whereas the task is grounded in visual processing, given a complex free form question the language understanding component becomes crucial. In this work, I hypothesize that if the question focuses on events described by verbs, then the model should be aware of verb semantics, as expressed via semantic role labels, argument types, and/or frame elements. Unfortunately, no VQA dataset exists that includes verb semantic information. My first contribution is a new VQA dataset (imSituVQA) that I built by taking advantage of the imSitu annotations. The imSitu dataset consists of images manually labeled with semantic frame elements, mostly taken from FrameNet. Second, I propose a multi-task CNN-LSTM VQA model that learns to classify the answers as well as the semantic frame elements. The experiments on imSituVQA show that semantic frame element classification helps the VQA system avoid inconsistent responses and improves performance.

Semantic role labeling is an alternative solution to approximately annotate any VQA dataset of interest. I employed a PropBank based semantic role labeler to label a subset of the VQA dataset (VQA_{sub}). Then I trained the proposed multi-task CNN-LSTM model with VQA_{sub}. The results show a slight improvement over the single-task CNN-LSTM model.

CHAPTER 1

INTRODUCTION

Visual Question Answering (VQA) is a multimodal task falling at the intersection of Computer Vision and Natural Language Processing (NLP). Given an image and a question in natural language, the task of a VQA system is to provide a correct natural language response. The recent increasing interest in VQA is driven by the availability of large datasets and the success of Deep Learning in both Computer Vision and NLP (Ferraro et al., 2015). Several deep neural network approaches have been proposed to model the task in an end-to-end fashion. With all the progress so far, the task of VQA is not solved yet.

Here, I describe two problems VQA suffers from: (1) The VQA task is usually grounded in visual processing, however, if the question focuses on events described by verbs, the language understanding component becomes crucial. I hypothesize that the model should be aware of verb semantics, as expressed via semantic role labels, argument types, and/or frame elements. Unfortunately, no VQA dataset exists that includes verb semantic information. (2) End-to-end VQA models map a raw input <image, question> pair to a joint representation that is passed to a classification layer on answers. This approach ignores the semantic of answers. For example, a subset of answers can be stronger candidates for questions about location ([kitchen, office, outdoor] versus [blue, 2, pasta]). My hypothesis is that the model should be aware of the higher-level semantic of answers so as to avoid irrelevant responses and to improve gener-

alization of the model.

The next three sections describe the three main contributions of my work.

Figure 1. Word cloud of the verbs in the VQA dataset (excluding *'to be'*)

1.1 Need for a VQA dataset annotated with verb semantic information

Many large scale datasets for VQA have been created. Exploring these datasets one can realize that questions are mostly simple, short and more visually than linguistically challenging. One reason comes from the fact that usually, people from Computer Vision research communities collect the datasets. For example, the DAQUARE (Malinowski and Fritz, 2014) and CLEVER (Johnson et al., 2017) datasets include questions limited to objects and their at-

tributes. The VQA dataset (Antol et al., 2015) is among the rare datasets with open-ended free form questions. Figure 1 visualizes a word cloud of verbs excluding "to be" in the VQA dataset. I have observed that a large portion of the questions available in the VQA dataset, involves a verb other than *"to be"* (i.e., 43% of VQA dataset Figure 2). Then it becomes important for the model to be aware of the arguments the verb can take. This information can be described via semantic role labels, argument types, and/or frame elements. Semantic information about verbs includes the type of arguments a verb can take and how they can participate in the event expressed by a verb, but this information is missing in current VQA systems. I contend that, if a VQA system is aware of such semantic information, it can not only narrow down the possible answers but also avoid providing irrelevant responses. For example, the answer to the question *"What is the woman cooking in the oven?"*, should belong to the *FOOD* semantic category. However, neither do VQA datasets encode, nor has any VQA system taken advantage of this information.

The question is how to incorporate such semantic information in VQA. Traditionally in linguistics, semantic information about a verb has been captured via so-called thematic or semantic roles (Martin and Jurafsky, 2009), which may include roles such as *AGENT* or *PATIENT* as encoded in a resource such as VerbNet (Kipper et al., 2008), the abstract roles provided by PropBank (Palmer et al., 2005), or rather, the specialized frame elements provided by FrameNet (Fillmore et al., 2003). In FrameNet, verb semantics is described by frames, schematic representations of situations. Frame elements are defined for each frame and correspond to important entities present in the evoked situation. For example, the frame *Cooking-Creation* has four core

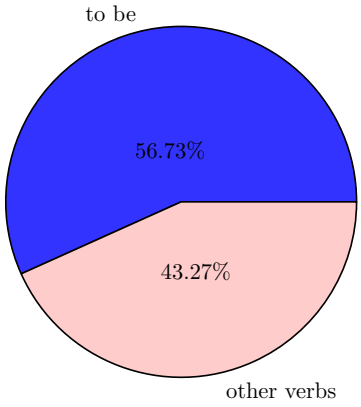


Figure 2. Distribution of verbs in the VQA dataset, *'to be'* versus *'other verbs'*

elements, namely *Produced Food, Ingredients, Heating Instrument, Container*. I took advantage of the imSitu dataset (Yatskar et al., 2016), developed for situation recognition and consisting of about 125k images. Each image is annotated with one of 504 candidate verbs and its frame elements according to FrameNet. My first contribution is the creation of the new imSituVQA dataset, by employing a semiautomatic approach in order to create question-answer pairs from the imSitu dataset.

1.2 A novel multi-task CNN-LSTM VQA model

Beside datasets, many models have been proposed and deep neural networks (Deep Learning) have been a dominant approach as of late(Wu et al., 2017). As mentioned earlier, end-to-end models do not exploit semantic information in answers. These models can be augmented

with linguistic information about answers and modified in order to respond coherently.

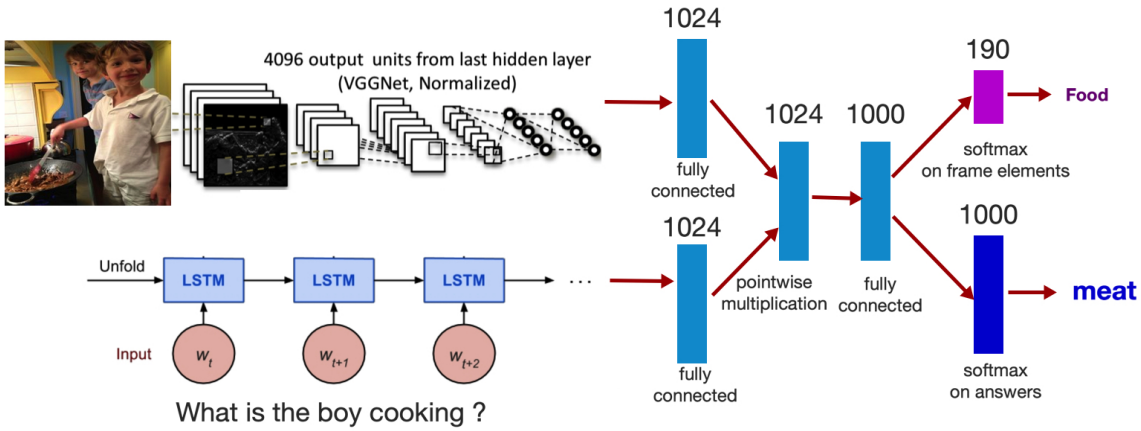


Figure 3. The proposed multi-task CNN-LSTM VQA model

(Xie et al., 2015) proposed a framework known as hyper-class augmented and regularized deep learning. The authors suggest a hyper-class augmentation formulated as multi-task learning in order to boost the recognition task in fine-grained image classification (FGIC). Inspired by their ideas, I formulate a VQA task as a multi-task learning problem. In this formulation, the system should learn to classify answers as well as hyper-classes. Figure 23 depicts an adapted version of the VQA task. Hyper-classes can be any additional information relevant to

answers. For example in the imSituVQA answers are labeled with semantic frame information. The semantic frames can be included as hyper-classes.

My second contribution is to augment a CNN-LSTM VQA model with semantic information employing the described multi-task learning approach (Figure 3). The model is trained to classify answers as well as semantic frame elements. The two classifiers share the same weights and architectures up to the classification point. The experiments on imSituVQA show that the frame element classification acts as a regularizer to improve VQA performance. Augmenting VQA with frame element information boosts accuracy up to 5%.

1.3 Automatic semantic role labeler as an alternative annotation tool

In order to train the proposed VQA model with any VQA dataset of interest semantic role labels must be available. In order to avoid costly manual annotations, automatic semantic role labeling is a quick alternative solution. I employed ClearNLP (Choi and Palmer, 2012) (PropBank based semantic role labeler) to label a subset of the VQA dataset (VQA_{sub}). *"R-"* pattern was used as an indicator of the response semantic role. For example, *"What kind of pants does the woman wear?"* is labeled as *V:wear.01 A0:woman* ***R-A1:kind***. **R-A1** indicates *A1* is the response semantic role. I used SemLink (Palmer, 2009) in order to map PropBank abstract roles to VerbNet roles. For example *A1* was mapped to *Theme*. I employed the proposed multi-task CNN-LSTM model for training and testing. The results show a slight improvement over the single-task CNN-LSTM model.

1.4 Outline

This thesis is organized as follows:

Chapter 2 reviews resources and related work in VQA, from datasets to proposed architectures and models.

Chapter 3 provides a short background on primary neural network concepts and models. The CNN-LSTM model is briefly explained since it is the primary VQA model employed in this proposal.

Chapter 4 describes the process of extracting a novel VQA dataset (imSituVQA) from the currently available imSitu dataset. The process is composed of two primary steps: (1) Question answer template generation and (2) Question answer pair realization.

Chapter 5 describes augmenting VQA with semantic frame information in a multi-task learning approach. Further evaluations and fine-grained analysis are described, and so is the performance of the proposed VQA model.

Chapter 6 generalizes the proposed idea to be working for any VQA dataset of interest. Automatic semantic role labeling is an approximate alternative to manual annotation. Manual annotation is expensive and time consuming. I employed semantic role labeling tools to label the VQA dataset (Antol et al., 2015) in order to see how the proposed multi-task CNN-LSTM works.

Chapter 7 concludes the thesis and proposes future research directions.

CHAPTER 2

RESOURCES AND RELATED WORK

In this chapter, I review various datasets and models that have been used to tackle the VQA task.

2.1 Datasets

Datasets differ based on the number of images, the number of questions, complexity of the questions, reasoning required and content information included via annotation for images, and questions. Performance of any model of interest is typically measured via *accuracy*, unless otherwise mentioned.

The DAtaset for QUestion Answering on Real-world images (DAQUAR) (Malinowski and Fritz, 2014) was among initial datasets published for the VQA task. The images are taken from NYUDepth Version2 dataset (Silberman and Fergus, 2012). The images are all of indoor scenes. NYUDepth Version2 is annotated with semantic segmentation information. DAQUAR includes 1449 images (795 training, 654 test). Question answer pairs are collected in two ways: (1) manually by human annotators with focus on colors, numbers and objects; (2) using predefined templates to generate from the NYU dataset (*"How many [object] are in [image id]?"*). In total, 12,468 question answer pairs were collected (6,794 training, 5,674 test). Unfortunately, DAQUAR is restricted as the answers are among a predefined set of 16 colors and 894 object categories. It also suffers from bias resulting from humans focusing on a few prominent objects

such as tables and chairs in the image. Beside *accuracy*, the authors proposed *WUPS* in order to measure performance. *WUPS* is defined based on *WUP* (Wu and Palmer, 1994). *WUP(a, b)* measures similarity based on the depth of two words a and b in a taxonomy such as WordNet. *WUPS* generates a score between 0 and 1. It is typically thresholded at 0.9 indicating whether an answer is correct or not. Figure 17 and section 5.2.2 provide futher details on WUPS.

Many VQA datasets utilize the Microsoft Common Objects in Context (MS-COCO) (Lin et al., 2014) image dataset. MS-COCO consists of 2.5 M instances of 91 object types for object recognition. The images are taken from complex everyday scenes of common objects in a natural context.

The COCO-QA dataset (Ren et al., 2015a) is a dataset based on the MS-COCO dataset. It was one of the first attempts at increasing the scale of the dataset for the VQA task. The <question, answer> pairs are automatically generated from MS-COCO caption annotations. The questions generally fall in four categories: *Object*, *Number*, *Color* and *Location*. For each image, there is one question with a single word answer. The dataset contains a total of 123,287 samples (72,783 training and 38,948 testing). Performance is assessed via either *accuracy* or *WUPS* score. Automatic conversion of captions results in a high repetition rate of the questions. Also since captions are describing the main information of the image, it does not provide detailed specific questions.

The VQA dataset (Antol et al., 2015) is the most widely used dataset for the VQA task. It is mostly because of the free-form and open-ended design of the questions and answers. For open-ended questions, potentially major AI capabilities are needed to answer: fine-grained

recognition (e.g., *"What kind of food is served?"*), activity recognition (e.g., *"Is this man playing tennis?"*), object detection (e.g., *"How many zebras are there?"*), commonsense reasoning (e.g., *"Does this person follow the rules?"*) and knowledge base reasoning (e.g., *"Is this a hybrid car?"*). Real images are selected from the MS-COCO dataset. Questions and answers were generated by crowd-sourced workers. For each question image pair, 10 answers were obtained from each person. Answers are usually a single word or multiple words. Almost 38% of the questions are Yes/No, 12% Number and 50% Others. The original VQA dataset has 204,721 images with 614,163 questions, 3 questions per image on average (248,349 training, 121,512 validation, 244,302 testing). The second version of the VQA 2.0 has also been proposed (Goyal et al., 2017). It extends the VQA dataset by balancing Yes/No type of questions. A machine response is evaluated via a VQA specific accuracy measure. An answer is considered correct if it matches the answers of at least three annotators.

The Visual Genome [1] QA (Krishna et al., 2017) is the largest dataset for VQA, (1.7 M question/answer pairs). It includes structured annotations known as scene graphs. These scene graphs specify visual elements, attributes, and relationships between elements. Questions were created by human subjects. Questions start with one of the 7 possible question words (*Who, What, Where, When, Why, How,* and *Which*). A major advantage of the Visual Genome QA dataset for VQA is the structured scene annotations. The diversity of the answers is also larger in comparison to the VQA dataset. The Visual7W dataset (Zhu et al., 2016) is a subset of

---

[1] This is the terminology as chosen by the authors. Here is the webpage of the project: `https://visualgenome.org/`

the Visual Genome dataset with additional annotations. Objects mentioned in the question were drawn with bounding boxes in the image in order to resolve textual ambiguity and to enable answers of a visual nature. The questions are evaluated in a multiple choice way with 4 candidate answers of which only one is correct. The dataset contains 47,300 images and 327,939 questions.

The Compositional Language and Elementary Visual Reasoning diagnostics dataset (CLEVR) (Johnson et al., 2017) was proposed to alleviate the biased problem of VQA benchmarks. In this way it prevents the models from exploiting the situation in order to answer questions without reasoning. It challenges visual reasoning capabilities such as counting, logical reasoning, comparing, and storing information in memory. It is designed so that accessing external knowledge bases and using common sense may not help in order to answer the questions. Images are annotated with ground-truth object positions and attributes (*shape, size, color, material*). Questions are generated automatically using textual templates (i.e. *"How many <Color> <Material> things are there?"*) from 90 question families. CLEVR has 100K rendered images (simple 3D shapes) and about one million questions of which 853K are unique.

The focus of many VQA datasets is on questions which require direct analysis of an image in order to answer. There are many questions that require common sense, or basic factual knowledge to be answered. FVQA (Fact-based VQA) (Wang et al., 2018) was proposed by appending supporting fact information to VQA (<image, question>,answer) samples. The supporting fact is represented as a triplet such as <Cat, CapableOf, ClimbingTrees>. 2190 images were sampled from the MS-COCO. Each image is annotated with visual concepts (objects, scenes, and

actions) using available resources and classifiers. The knowledge about each visual concept is extracted from structured knowledge bases, such as DBpedia, ConceptNet, and WebChild. Annotators created 5,826 questions in which answering each question requires information from both the image and selected supporting facts.

2.2 Models

Multi-World QA (Malinowski and Fritz, 2014) is among the popular initial approaches which do not employ deep learning in order to build a VQA system. The possibility of a response given <image, question> inputs is formulated as follows:

$$\mathsf{P}(A = a|\mathit{Question}, \mathit{World}) = \sum_{ST} \mathsf{P}(A = a|ST, \mathit{World})\mathsf{P}(ST|\mathit{Question}) \tag{2.1}$$

ST is a hidden variable (Liang et al., 2013) related to a semantic tree. This semantic tree is obtained from running a semantic parser on the question. W is the world [1], representing the image (features can be obtained from segmentation). A deterministic evaluation function evaluates $\mathsf{P}(A = a|T, W)$ and a log-linear model is trained in order to obtain $\mathsf{P}(T|Q)$. This model was named SWQA. The authors also extended SWQA to a multi-world case covering model unpredictability in class labeling and segmentation. These models were evaluated on the DAQUAR dataset.

Deep learning models typically employ Convolutional Neural Networks (CNNs) to embed the image. CNNs can be any of the popular models such as AlexNet(Krizhevsky et al., 2012),

[1] This is the terminology as chosen by the authors of this paper

GoogLeNet (Szegedy et al., 2015) , VGGNet (Simonyan and Zisserman, 2015), ResNet (He et al., 2016)and so on. Question words are initially mapped to dense vectors using word embeddings such as Glove (Pennington et al., 2014) or Word2Vec (Mikolov et al., 2013). Then, a sequence of vectors is usually passed to a Recurrent Neural Networks (RNNs) in order to embed the question. RNNs are usually LSTMs (Hochreiter and Schmidhuber, 1997) or GRUs (Chung et al., 2014)

Joint embedding is motivated by the advances of Artificial Neural Networks (Deep Learning) in both Computer Vision and NLP. Learned embeddings of image and question are joined in some fashion such as concatenation, multiplication or any complex fusion. Then after applying a number of fully connected layers, an answer can be provided via classification on a number of frequent responses or via generation by means of an RNN model. The approach is shown at the top of Figure 4.

iBOWIMG (Zhou et al., 2015) employed the last layer of the pre-trained GoogLeNet model for image classification in order to extract image features. Textual features of the question were taken using a simple bag-of-word model. The concatenation of the features is passed to a softmax layer for answer classification. The proposed model was evaluated on the VQA dataset showing comparable performance with RNN based techniques.

(Ma et al., 2016) proposed a fully CNN based model. Not only is the image embedded using the CNN model (VGGNet) but also the question is encoded using a CNN. While encoding the question via CNN, the encoded question interacts with the image representation inter-modally in order to produce an answer. The image CNN generates an image representation (ν_{im}), the

sentence CNN applies convolution with receptive field size 3 on the question (v_{qt}). Finally a multimodal CNN fuses the multimodal inputs (v_{im} and v_{qt}) together to generate their joint representation (v_{mm}). This joint representation is given to a softmax layer to predict the answer. The model was evaluated on the DAQUAR and COCO-QA datasets.

(Malinowski et al., 2017) used a pre-trained VGGNet in order to obtain the image representation. The word embeddings of the question words are fed to an LSTM in order to obtain the question representation. The answer is decoded in two ways: classification over answers or as a generation of the answer. A fully connected layer followed by a softmax layer can be used in order to predict the answer class. On the other hand, an LSTM can be employed in order to decode and generate an answer. The decoder can generate variable length answers as long as it does not generate the $<END>$ symbol.

Attention methods improve on deep learning baselines by focusing on specific regions of the input. In VQA, image features are replaced with spatial features. This allows correlating the question embedding and regions of the image. The way this mapping is defined and how it is interacting with question words has resulted in numerous attention based models. These models are commonly analyzed based on how each attends differently to image regions in order to answer. It is preferred that the model attends to relevant areas where the clue is located. This way the attention model performs much better as compared to joint embeddings. A general uniform attention model is depicted at the bottom of Figure 4

(Shih et al., 2016) proposed an attention-based model referred to as WTL (Where To Look). The image representation is obtained from a VGGNet and the question representation

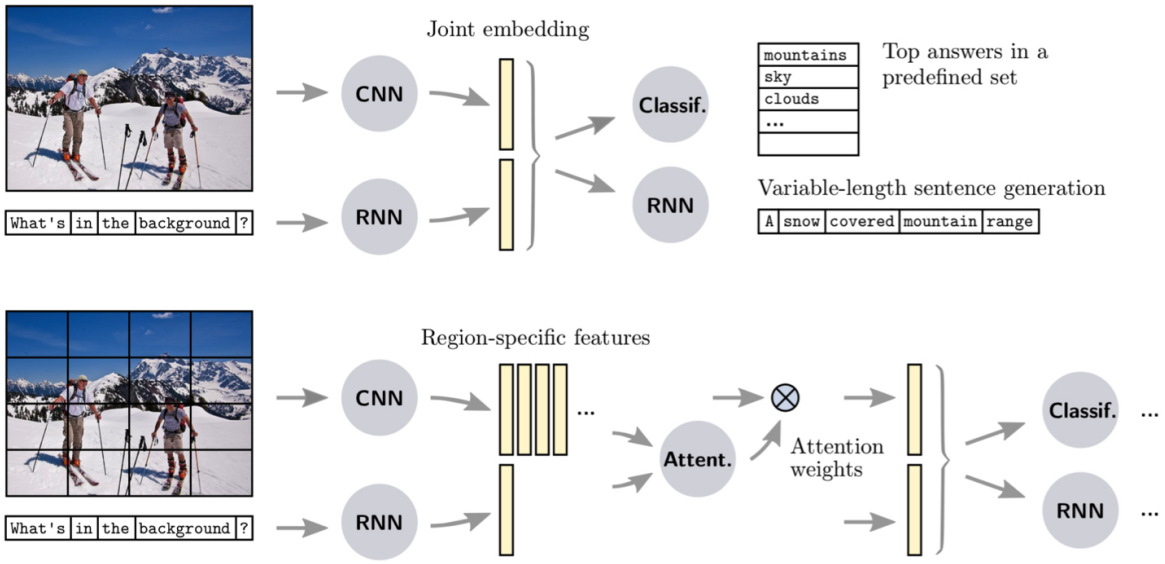


Figure 4. Common approaches in order to model a VQA task (Wu et al., 2017)

by averaging word embeddings. An attention vector is defined over a set of image features in order to capture the importance the model should assign to each region of the image. If $V = (\vec{v_1}, \vec{v_2}, ..., \vec{v_k})$ is the set of image features, and $\vec{q}$ is the question embedding, then the importance of the j_{th} region is computed as

$$g_j = (A\vec{v_j} + b_A)^T (B\vec{q} + b_B) \tag{2.2}$$

The final image embedding is an attention weighted sum of the different regions. The proposed model was evaluated on the VQA dataset.

(Yang et al., 2016) employs stacked attention networks (SAN) infering the answer repeatedly. The image feature matrix of dimensionality 512×14×14 is obtained from the last pooling layer of a pre-trained VGGNet. This indicates 196 feature vectors of dimensionality 512 for each region of the image. If V_I indicates the image feature matrix, and q is the question embedding, then passing V_I and q through a neural network with a layer with $\tanh$ activation function followed by a softmax layer results in new image features (v_i) with corresponding attention probabilities (p_i). Then having p_i as weights, a new refined image query (v_q) is defined as a weighted sum of the v_i vectors. This process is iterated having v_q as new input. Reasoning via multiple attention layers iteratively, the SAN can filter out the noise and attend to regions that are relevant to the answer.

(Anderson et al., 2018) proposed a mixture of top-down and bottom-up attention technique. This results in attention which can be computed at important image regions including the objects. The main idea is that image regions should not necessarily be of the same size since an object can be small or large, hence it may require a region in which fits. The bottom-up mechanism employs a Faster R-CNN (Ren et al., 2015b) in order to propose image regions with its features. On the other hand, the top-down mechanism figures out the attention weights of each region.

(Lu et al., 2016) adds visual attention but also models question attention. Two forms of co-attention were proposed: 1) Parallel co-attention, in which simultaneously image and question attend over each other. 2) Alternating co-attention, which sequentially switches attention

between generating image and question. The models were shown to improve performance on the VQA dataset.

Multimodal Compact Bilinear (MCB) pooling (Fukui et al., 2016) suggested a different way of combining image and question features in VQA. The idea is to calculate an approximation of the outer product of image and question features. This would allow a deeper and better interaction between vision and language modalities. The proposed idea was shown to perform very well on the COCO-VQA and the VQA datasets.

Compositional models apply multiple levels of reasoning in order to answer a question. For example, *"What is to the right of the car?"* can be decomposed into first searching the car, and then calling the instance to the right of it. Two popular compositional approaches are Neural Module Networks (NMN) (Andreas et al., 2016) and Recurrent Answering Units (RAU) (Noh and Han, 2016). Neural Module Networks (NMN) employ external textual parsers in order to decompose the task into the subtasks. Recurrent Answering Units (RAU) are trained completely end-to-end in order to learn sub-tasks.

A regular VQA system is unable to answer questions that require external information. Knowledge-based techniques are designed in order to cover this shortcoming. They utilize external structured information ranging from commonsense to encyclopedic level. The Ask Me Anything (AMA) model (Wu et al., 2016) exploits information from an external knowledge base in order to guide a visual question answering system. Initially, it extracts attributes such as object names, properties and so on from the caption of the image. The caption is obtained from a sample image captioning model trained on the MS-COCO dataset. Attributes are used to

generate queries for DBpedia (Auer et al., 2007). Each query returns a text which is embedded via Doc2Vec (Le and Mikolov, 2014). This Doc2Vec is passed as an additional input to the LSTM in order to generate the answer. The proposed model was evaluated on the VQA dataset and the COCO-QA dataset.

CHAPTER 3

BACKGROUND

This chapter provides background on Semantic Roles and Neural Networks. Semantic roles are representations expressing the semantic arguments associated with the predicate or verb of a sentence. The semantic roles are described in lexical resources such as PorpBank and FrameNet. The first part of this chapter briefly describes the idea; details on semantic Roles can be found in (Martin and Jurafsky, 2009). Artificial Neural Networks (ANN) have been studied for a long time since the 1940s. Deep learning has been a dominant approach to model a VQA task. Deep learning models typically employ Recurrent Neural Networks (RNN) to embed the question and Convolutional Neural Networks (CNN) to embed the image. The second part of this chapter explains the concepts and models used in this thesis and further details on deep learning can be found in (Goodfellow et al., 2016).

3.1 Semantic Roles

The goal of semantic roles is to achieve a common representation for sentences like *"the woman cooks pasta"* and *"cooking of pasta by the woman"*. There is a cooking event, the participants are woman and pasta, and woman is the cook. Semantic roles are representations expressing the role that arguments of a predicate take in the event. Semantic role labeling (SRL) is the task of predicting the semantic roles in a sentence.

Thematic roles are one way to capture this semantic commonality. AGENT is the thematic role that represents an abstract idea such as volitional causation. Theme prototypically represents inanimate objects that are affected in some way by the action. Table I shows more examples of important thematic roles. In the two sentences describing the event of the woman cooking pasta mentioned above, woman is Agent and pasta is Theme.

Thematic Role	Definition
AGENT	The volitional causer of an event
EXPERIENCER	The experiencer of an event
FORCE	The non-volitional causer of the event
THEME	The participant most directly affected by an event
RESULT	The end product of an event
CONTENT	The proposition or content of a propositional event
INSTRUMENT	An instrument used in an event
BENEFICIARY	The beneficiary of an event
SOURCE	The origin of the object of a transfer event
GOAL	The destination of an object of a transfer event

TABLE I. Some commonly used thematic roles with their definitions(Martin and Jurafsky, 2009)

Difficulty in defining the thematic roles, have led to different semantic role sets that use either many fewer (such as PropBank) or many more roles (such as FrameNet).

3.1.1 PropBank

The Proposition Bank (PropBank) (Palmer et al., 2005) adds semantic role labels, to the syntactic structures of the Penn TreeBank. Penn TreeBank (Marcus et al., 1993) is a parsed text corpus annotated with linguistic information such as syntactic or semantic sentence structure.

In PropBank, semantic roles are defined with regard to verb senses. The roles are given numbers rather than names: Arg0, Arg1, Arg2, Arg3, Arg4 and ArgM (M as initial for modifier). In general, Arg0 represents the AGENT, and Arg1, the PATIENT. The Arg2 is often the instrument, the Arg3 the start point, and the Arg4 the end point. Table II shows semantic roles *"Sales fell by 50% to $100 from $200"*. Among the 12 senses of *fall* in VerbNet, *fall* in this sentence is recognized as *fall.O1*, whose definition includes roles Patient (Arg1), Extent (Arg2), Start point (Arg3) and End point (Arg4).

3.1.2 FrameNet

FrameNet is based on a theory of Frame Semantics (Fillmore et al., 2003). The meanings of a sentence is described by a semantic frame. The semantic frame is a representation depicting type of event and the participants in it. In order to explain the idea I show an example: The concept of shopping usually involves a person doing the shopping (*Shopper*), the goods that

Arg#	Semantic	Words
Arg1	Logical subject, patient, thing falling	Sales
Arg2	Extent, amount fallen	by 50%
Arg3	start point	from $200
Arg4	end point, end state of arg1	to $100

TABLE II. PropBank semantic roles for "Sales fell by 50% to $100 from $200" (Martin and Jurafsky, 2009)

are being shoped (*Goods*). In the FrameNet database [1], this is represented as a frame called *Shopping*, and the *Shopper* and *Goods* are called frame elements (FEs).

3.2 Feedforward Neural Network

A *neuron* is an atomic computational unit of a neural network. It gets a weighted sum of the input vector x and outputs y after applying an activation function f such as *sigmoid*, *tanh* or *relu*. As shown in Figure 5, a Feedforward Neural Network (FNN) is composed of a series of levels (layers). Each level is composed of similar computational units and gets the output of the previous level and after applying an activation function, it passes it to subsequent level. In this way given an input, a series of matrix manipulations maps it to the output. In case

[1] `https://framenet.icsi.berkeley.edu/fndrupal/`

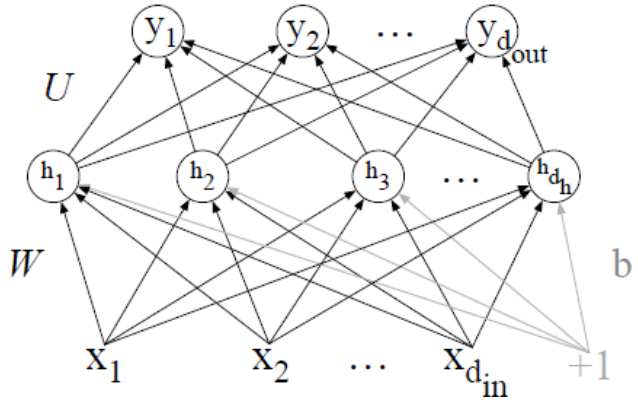


Figure 5. A basic feedforward neural network (Goodfellow et al., 2016). Input is passed through a number of hidden non-linear layers. The last layer outputs the result.

of classification, the output is a probability distribution over classes (*softmax*). For example a feedforward neural network with one hidden layer can be expressed as follows:

$$h = f(W^T x + b) \tag{3.1}$$

$$z = g(U^T h + b) \tag{3.2}$$

$$y = \mathrm{softmax}(z) : \mathrm{softmax}(z)_i = \frac{\exp z_i}{\sum_j \exp z_j} \tag{3.3}$$

Training a neural network model means adjusting weight parameters (i.e. W and U in Figure 5) so as to minimize the error (loss function) of the output. This may require passing samples to the model several times (each is called one *epoch*). In order to adjust weights, the

model requires to compute a loss function. In case of classification, the loss function is usually cross entropy (Equation 3.4). A training algorithm should adjust each weight based on its contribution to the error. In mathematical terms, this is expressed by the derivative of the loss function to each weight parameter. Backpropagation is a popular algorithm which computes derivatives and adjusts each weight. Different optimization algorithms are proposed concerning how much each weight should be updated. The batch size is the number of samples processed before the model is updated.

$$\text{cross entropy} = -\sum_{i=1}^{C} y_i \log(y_i) \tag{3.4}$$

3.3 Convolutional Neural Network (CNN)

Deep learning models typically employ Convolutional Neural Networks (CNNs) to embed the image. A variety of deep CNNs have been proposed for visual recognition tasks. Here I briefly describe how a basic CNN works. CNN is an extension of FNN introducing convolution and pooling layers (see Figure 6). A convolution layer applies a convolution function to the input. This convolution function is called *kernel* or *filter*. The main purpose of convolution is to extract features from the input image. Applying a sample m×n *kernel* to an M×N image means to move it across the image and to compute an inner product of the kernel with the intersected image. This results in newly filtered images. A *relu* unit is used in order to make sure the output is never negative. These filtered images are passed through a pooling layer in order to subsample images and reduce dimensionality. For example, max pooling selects the

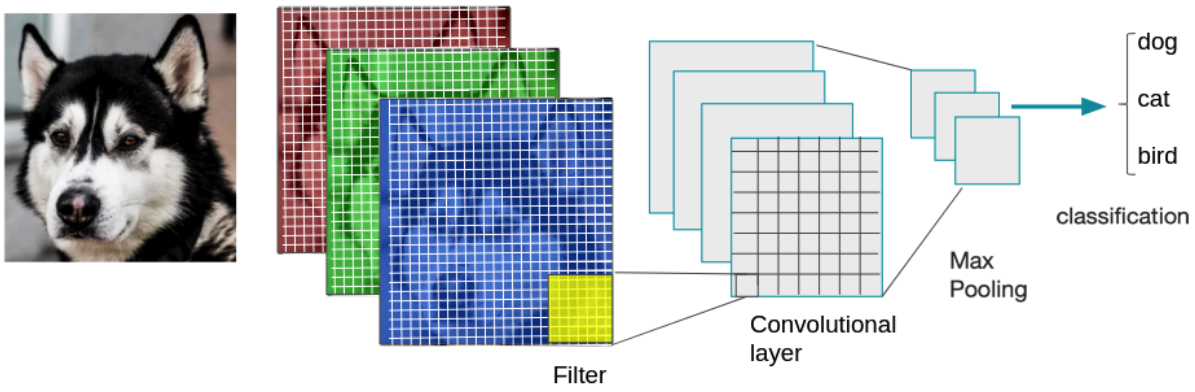


Figure 6. A basic Convolutional Neural Network (CNN) block for a classification task (Goodfellow et al., 2016)

max value of the m×n sub-image. A convolution layer with a subsequent pooling layer acts as a building block of CNNs. This building block repetition has resulted in different CNN architectures such as GoogLeNet (Szegedy et al., 2015), AlexNet(Krizhevsky et al., 2012), ResNet (He et al., 2016), VGGNet (Simonyan and Zisserman, 2015), and so on. After a stack of convolutional layers, a layer flattens the output of pooling in order to pass it to a classification layer (with likely fully connected layers in between).

3.3.1 VGGNet

VGGNet (Simonyan and Zisserman, 2015) is a very deep convolutional network proposed by the Visual Geometry Group (VGG) at the University of Oxford. As shown in Figure 7, the input is a 224×224 RGB image. The image is given to convolutional layers. Filters with size 3×3 are applied at each convolution layer. The filtered images are the same size as the input (padding = 1, stride = 1). Spatial pooling is applied by five max-pooling layers, which follow some of the convolution layers. Max-pooling is applied over a 2×2 pixel window, with stride 2. Convolutional layers are followed by three fully connected layers ending in a 1000-class softmax on ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) (Russakovsky et al., 2015) classification.

3.4 Word Embedding

Word embeddings are described as a vector representation of a word's semantics or meaning. Dense embeddings such as Glove (Pennington et al., 2014) are trained directly on a large corpus in unsupervised mode. Training is based on aggregated global word-word co-occurrence statistics. Words are transformed into a sequence of word vectors while feeding a Neural Network. This is usually done by an embedding layer. The embedding layer weights can be initialized with Glove and fine-tuned based on a new task such as VQA.

3.5 Recurrent Neural Network (RNN)

The RNN is an architecture for processing sequential data such as a sequence of words in a natural language text. The main idea is that the output of step $\mathtt{i}$ is affected by input $\mathtt{i}$

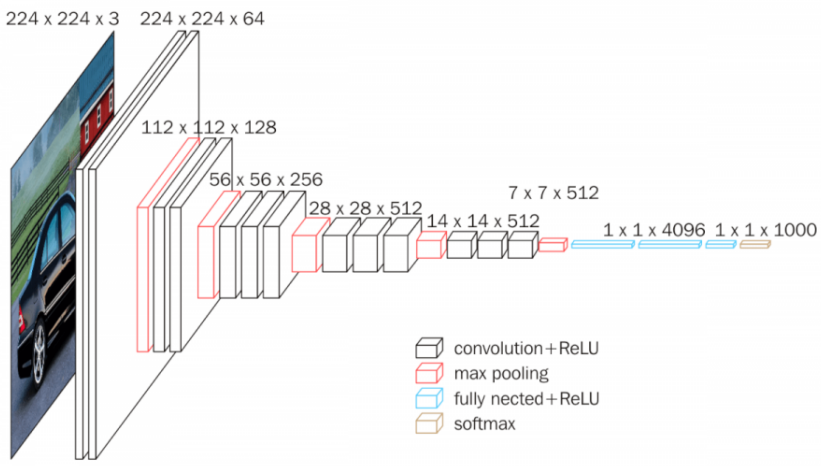


Figure 7. VGGNet: Deep Convolutional Network for large-scale image recognition (Simonyan and Zisserman, 2015). This pre-trained model is the primary source for extracting image features.

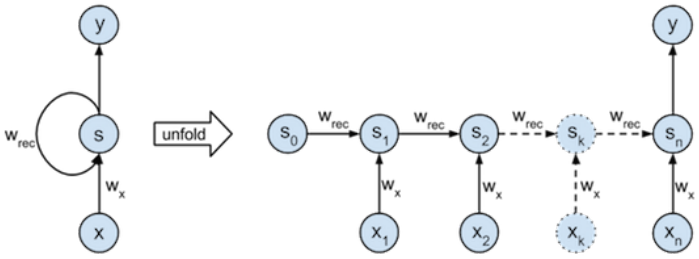


Figure 8. Recurrent Neural Network (RNN) (Goodfellow et al., 2016). It is widely used for modeling NLP tasks.

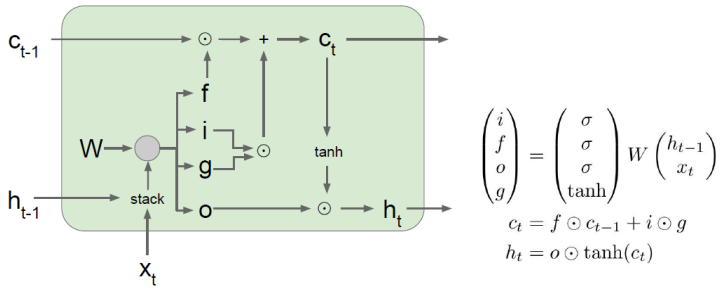


Figure 9. Lomg Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). This model does not suffer from the long short term dependencies problem. The bidirectional LSTM is the primary model used to extract question embedding.

and the output of the previous step $i-1$. Then, the sequence of vectors is usually passed to Recurrent Neural Networks (RNNs) to embed the question. Gated Recurrent (GRU) (Chung et al., 2014) or Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) are popular computational units employed in RNNs.

3.5.1 Long Short Term Memory (LSTM)

The main issue of a basic RNN is long short term dependency. An RNN with LSTM units is usually called an LSTM network. An LSTM layer is composed of 4 components: (1) an input gate, (2) cell, (3) a forget gate and (4) an output gate. A cell is responsible for memorizing values overtimes or over steps. The other three gates are responsible for regulating the flow of information to the cell and vice versa.

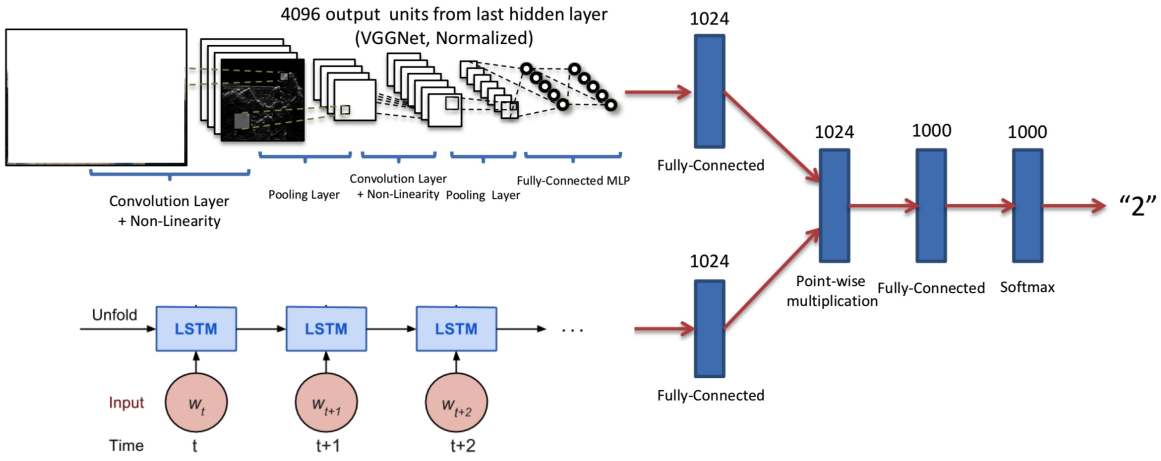


Figure 10. CNN-LSTM Model (Antol et al., 2015) is the primary VQA model used in this proposal.

3.6 Neural Network based VQA models

3.6.1 CNN-LSTM model

The CNN-LSTM Model (Figure 10) uses a two-layer LSTM to extract the question embedding and the last layer of VGGNet to extract the image features. Both the question and image features are mapped to the same dimensionality. Then the two vectors are fused via element-wise multiplication. Then, the fusion vector is given to a fully connected layer. Finally a softmax layer is employed to get a distribution over answers.

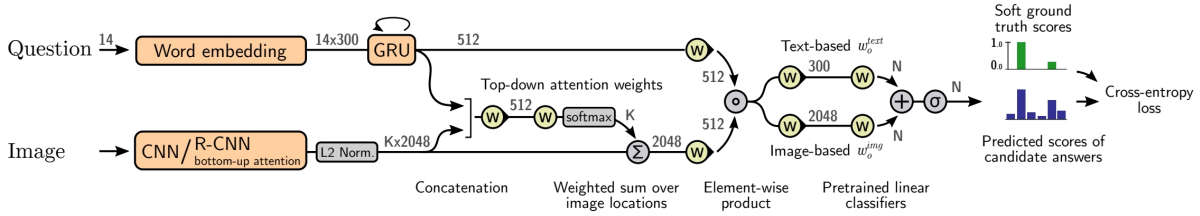


Figure 11. Bottom-Up and Top-Down Attention Model (Anderson et al., 2018). This model employed Faster R-CNN in order to attend at object level.

3.6.2 Attention model

A simple attention model can be obtained by getting the spatial features from the last convolution block of a VGGNet. As shown in Figure 7, this is a 7×7×512 tensor and a single vector can represent each of the 49 parts of the image. Then the model allows different features to be given different weights. Furthermore, each feature can interact with the question embedding. More complex is the bottom-up and top-down attention introduced in (Anderson et al., 2018). This attention is at the level of objects and other important image regions rather than uniform grid regions. The main idea is that image regions should not necessarily be of the same size since an object can be small or large, hence it may require a region in which it fits. This bottom-up mechanism employs a Faster R-CNN (Ren et al., 2015b) to propose image regions with their features. On the other hand, the top-down mechanism figures out the attention weights of each region. Figure 11 shows details of this architecture.

CHAPTER 4

IMSITUVQA: A NOVEL VQA DATASET (PREVIOUSLY PUBLISHED AS M. ALIZADEH AND B. DI EUGENIO. (LREC 2020) A CORPUS FOR VISUAL QUESTION ANSWERING ANNOTATED WITH FRAME SEMANTIC INFORMATION. PAGES 5526 – 5533)

This chapter briefly describes the imSitu dataset and explains the process of extracting a novel VQA dataset (imSituVQA) from the currently available imSitu dataset. The process is composed of two primary steps: (1) Question answer template generation: Question answer templates are generated from imSitu abstract verb definitions. (2) Question answer pair realization: The templates are filled with noun values from the imSitu annotated images.

The imSitu dataset (Yatskar et al., 2016) is collected for situation recognition task and includes about 125k images. The goal of situation recognition is predicting activities along with *substances*, , *actors*, *objects*, and *locations* and so on. imSitu utilizes linguistic resources such as FrameNet[1] (Fillmore et al., 2003) and WordNet[2] (Miller, 1995) in order to define a comprehensive space of situations. As described in Section 3.1.2, it provides representations helping to understand who ($AGENT$) did what ($ACTIVITY$) to whom ($PATIENT$), where ($PLACE$),

[1]The FrameNet database contains over 1200 semantic frames. A semantic frame is a description of a type of event the participants in it.

[2]WordNet is a lexical database of English.




fixing		**cooking**		**falling**		**buying**	
Agent	man	Agent	boy	Agent	leaf	Agent	woman
Object	roof	Food	meat	Source	tree	Goods	shoe
Part	tile	Container	wok	Goal	land	Payment	credit card
Tool	hammer	Tool	spatula			Seller	person
Place	roof	Place	kitchen			Place	shoe shop




catching		**painting**		**attaching**		**opening**	
Agent	bear	Agent	man	Agent	woman	Agent	cat
Caughtitem	fish	Item	boat	Item	fabric	Item	door
Tool	mouth	Tool	roller	Tool	hand	Tool	paw
Place	body of water	Place	outside	Place	workstation		

TABLE III. Sample imSitu (Yatskar et al., 2016) annotations of images about different events described by semantic frame.

using what (*TOOL*) and so on. A semantic frame is a descriptive structure depicting an event and the participants in it.

A sample of images from the imSitu dataset and their annotations can be found in Table III. Every situation in imSitu is described with one of **504 candidate verbs** such as *cooking*, *fixing*, *falling*, *opening*, *attaching* and so on. Each verb has a set of FrameNet related frame elements. For example S_r*(cooking)* ={ *AGENT, FOOD, CONTAINER, HEATSOURCE, TOOL, PLACE* } provides the set of semantic frame elements of the verb *cook*. The set is also expressed by an abstract definition: *"an AGENT cooks a FOOD in a CONTAINER over a HEATSOURCE using a TOOL in a PLACE"*. As another example S_r*(buying)* ={ *AGENT, GOODS, PAYMENT, PLACE* } includes a set of semantic frame elements of the verb *buy*. The abstract definition is : *"the AGENT buys GOODS with PAYMENT from the SELLER in a PLACE"*. Table III shows sample image annotations of some verbs such as *cook* and *buy*. The interested reader may refer to the imSitu online browser in order to explore the dataset. [1]

imSitu includes 190 unique frame elements, some shared among verbs such as *AGENT* and *TOOL*, while some are verb-specific such as *PICKED*$\in S_r$*(picking)*. Every image is labeled with one of the 504 candidate verbs along with frame elements filled with noun values from WordNet. If an element is not present in the image its value is *empty*. There are about 250 images per verb and 3.55 roles per verb on average.

---

[1] `http://imsitu.org`

Question Word	Frame Elements
Who	*COMPETITOR, VICTIM, LISTENER, INDIVIDUALS, MOURNER, FOLLOWER, COAGENT, VOTEFOR, PERFORMER, EXPERIENCER, TICKLED, SELLER, EATER*
Where	*PLACE, TARGET,ADDRESSEE, SURFACE, GROUND, END, SOURCE, SHELTER, SURFACE, RECIPIENTS,CONTAINER, GOAL, STAGE, SCAFFOLD*
What	*OBJECT, HUNTED, BORINGTHING,FOCUS, OCCASION, SUBSTANCE, CLOTH COMPONENTS, DEPICTED, REFERENCE, AGENTTYPE, FOOD, CENTER, CLOTH*
What item	*ITEM, SIGNEDITEM, CAUGHTITEM, TURNEDITEM, GOODS, HIDINGITEM DRENCHEDITEM, REMOVEDITEM, DEFLECTEDITEM, WRAPPEDITEM*
What part	*PART, BODYPART, YANKEDPART, VICTIMPART, ITEMPART, RECIPIENTPART, AGENTPART, OBJECTPART, COAGENTPART*
What [Frame Element]	*VEHICLE, CONTAINER, SKILL, SHAPE, PATH, LIQUID IMITATION, MATERIAL, INSTRUMENT, PHENOMENON, OBSTACLE, EVENT*
What does the [AGENT] use to	*CROWN, BRUSH, CONNECTOR, GLUE, WRAPPINGITEM, COMPONENT, LOCK, COVER, DYE, PARACHUTE, ACTION, SEALANT*

TABLE IV. A subset of frame elements and the question words they are mapped to.

4.1 Question answer template generation

The main idea behind question template generation is to ask questions about one of the frame elements of a given verb based on its abstract definition. For example a question about

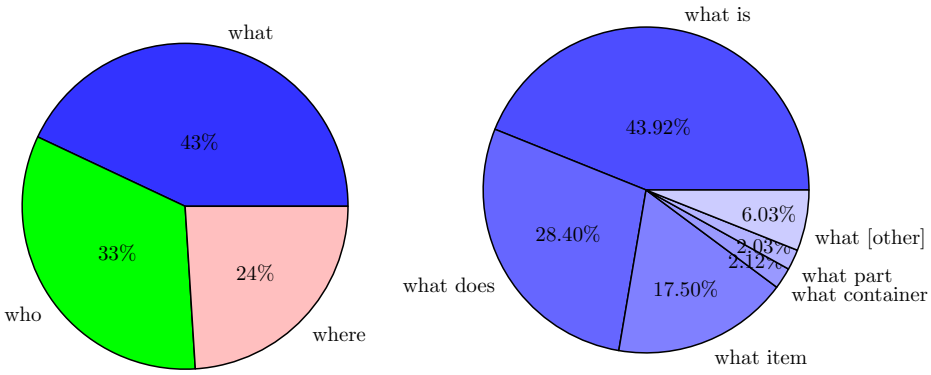


Figure 12. Distribution of questions in templates. (a) covers all questions while (b) includes questions starting with question word "what"

cooking can ask about *AGENT*, *FOOD*, *CONTAINER*, *HEATSOURCE*, *TOOL* or *PLACE*. [1] Each frame element requires a relevant question word to be used. Consequently, we mapped every frame element to a question word. For example, *AGENT* to *who*, *LOCATION* to *where*, *ITEM* , *FOOD* and *PICKED* to *what item*, *TOOL* to *what does [AGENT] use to* and so on. From 190 unique frame elements, 47 were mapped to *who*, 19 mapped to *where*, 53 mapped to *what* and the remaining were mapped to a question word starting with *what* such as *what item*. Table IV shows a subset of frame elements and the question word they are mapped to.

[1] The only exception is question template "What is the AGENT doing?" whose the response frame element is labeled with VERB.

TABLE V. A subset of Question Answer templates generated for cooking, buying, catching and opening.

Abstract definition from imSitu dataset	**Sample Generated Question Templates**	**Reponse Frame Element**
An AGENT cooks a FOOD in a CONTAINER over a HEATSOURCE using a TOOL in a PLACE.		
	Who is cooking?	AGENT
	What does the AGENT cook with TOOL?	FOOD
	What is the AGENT doing ?	VERB
	What does the AGENT use to cook in CONTAINER ?	TOOL
	Where does the AGENT cook FOOD in CONTAINER ?	PLACE
The AGENT buys GOODS with PAYMENT from the SELLER in a PLACE		
	Who is buying GOODS ?	AGENT
	What is the AGENT doing ?	VERB
	What item does the AGENT buy with PAYMENT ?	GOODS
	Who does the AGENT buy GOODS from?	SELLER
	Where does the AGENT buy GOODS ?	PLACE
An AGENT catches a CAUGHTITEM with a TOOL at a PLACE.		
	Who catches at PLACE ?	AGENT
	What is the AGENT doing ?	VERB
	What item does the AGENT catches with TOOL	CAUGHTITEM
	Where does the AGENT catches CAUGHTITEM ?	PLACE
The AGENT opens the ITEM with the TOOL at the PLACE.		
	What does the AGENT use to open ITEM ?	TOOL
	Who opens ITEM ?	AGENT
	What item does the AGENT opens ?	ITEM
	Where does the AGENT opens ITEM with TOOL	PLACE

As shown in the first column of Table V, in imSitu, each verb is described by an abstract statement including all its frame elements. Appendix B lists the definitions in a segmented format. Therefore, there are 504 abstract definitions in total. An abstract definition defines a natural form of how prepositions and punctuations are used along frame elements. For example, for *cook* the abstract definition is *"an AGENT cooks a FOOD in a CONTAINER over a HEATSOURCE using a TOOL in a PLACE"*. We can easily segment the statement to *"[an AGENT] cooks [a FOOD] [in a CONTAINER][over a HEATSOURCE] [using a TOOL] [in a PLACE]"*. Now in order to ask a question about a specific frame element, we hold out its segment. For example if we hold out *FOOD* then what remains is *"[an AGENT] cooks [X] [in a CONTAINER][over a HEATSOURCE] [using a TOOL] [in a PLACE]"*. Then, we should decide which other segments should be included in the question. The only exception is *AGENT* which will always be included (with article *the AGENT*) if not held out as response frame element. Approximately, we considered all possible subsets of segments. For example: *"[an AGENT] cooks"* and *"[an AGENT] cooks [X] [in a CONTAINER][in a PLACE]"* are two possible combinations when *FOOD* is the response frame element. Finally the relevant question word is appended at the beginning of each combination and the verb form is modified accordingly. For example: *"What does the AGENT cook?"* or *"What does the AGENT cook in CONTAINER in PLACE?"*. A subset of question templates and their response frame elements for *cooking*, *buying*, *catching* and *opening* are shown in Table V. In total, 6879 templates are generated, with on average 13.65 question-answer templates per verb. Figure 12 shows the distribution of template questions in terms of question words.

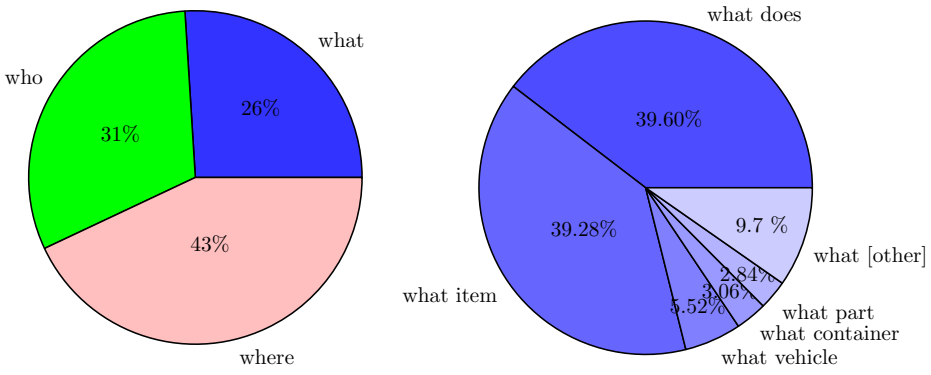


Figure 13. Distribution of questions in imSituVQA. (a) covers all questions while (b) includes questions starting with question word "what"

4.2 Question answer pair realization

The previous step generates templates for all 504 candidate verbs. As each image in imSitu is annotated with one verb, the templates of the annotated verb are considered for the image. Templates that include frame elements that are missing or empty in the image annotation are excluded. Then, given each question template and response frame element, the frame elements are filled with the noun values from the annotation. This realization process can be applied to all imSitu images. The final dataset is called imSituVQA. Each sample in imSituVQA is a <*image,question*> input pair that is labeled with an <*answer*> as output. For example given the image about *cooking* from Table III, applying the realization process on *"What does the AGENT cook in CONTAINER in PLACE?"* results in *"What does the boy cook in wok in*

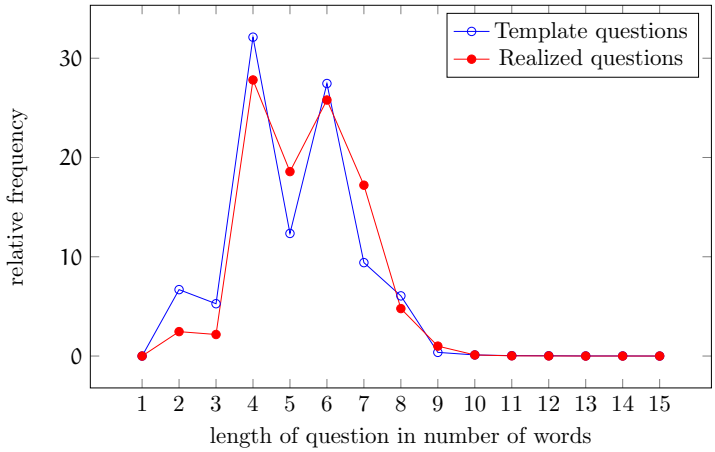


Figure 14. Distribution of template questions vs realized questions based on length.

kitchen?". Realizing the response frame element *FOOD* results in *"meat"* as answer. These three items compose a sample (<*image,question*>:<*answer*>) for the VQA task. Table VI shows VQA samples for *cooking*, *buying*, *catching* and *opening*. As can be seen, the dataset not only includes the typical question answer pairs but frame element annotations as well.

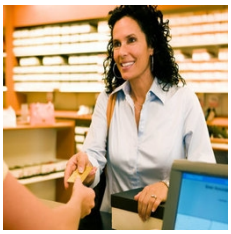
If a verb has n templates, applying an image annotation results in n real <*question, answer*> samples of the image. This way, the size of the extracted dataset is the average number of templates times the number of images. This realization process results in 254k train, 88k development and 88k test samples. For the training set, the top 10 most frequent frame element classes among the existing 190 are shown in Table VII. Table VIII also shows the top 10 frequent answers. Because 60% of answers are about *PLACE* and *AGENT*, the most frequent answers are usually values from these two frame elements. Figure 15 visualizes the relative frequency of

answers and response frame elements in terms of word clouds. The questions are mostly between 4 to 7 words. Figure 13 shows the distribution of imSituVQA questions according to the first question word. As can be seen *"Where"* is more frequent than *"Who"* and *"What"*. This derives from $PLACE$ being the most frequent frame element, twice as frequent as $AGENT$, which is the second. Figure 14 depicts the distribution of template questions and realized questions lengths in terms of the number of words. The distributions are very similar, showing the majority of questions are 4 to 7 words.

IMAGE about cooking

QUESTION	ANSWER
Who is cooking ? VERB	boy AGENT
What does the boy AGENT cook VERB with spatula ? TOOL	meat FOOD
Where does the boy AGENT cook VERB meat FOOD in wok ? CONTAINER	kitchen PLACE

IMAGE about buying



QUESTION	ANSWER
Who is buying shoes ? VERB ITEM	woman ? AGENT
Where does the woman AGENT buy VERB shoes ? GOODS	shoe store PLACE
Who does the woman AGENT buy VERB shoes ITEM from ?	person SELLER

IMAGE about catching

QUESTION	ANSWER
What is the bear AGENT doing ?	catching VERB
Where does the bear AGENT catch VERB fish ? CAUGHTITEM	body of water PLACE
What item does the bear AGENT catch VERB ?	fish CAUGHTITEM

IMAGE about opening

QUESTION	ANSWER
Who opens the door VERB ITEM	cat AGENT
What does the cat AGENT use to open VERB the door ? ITEM	paw TOOL
What item does the cat AGENT open VERB ?	door ITEM

TABLE VI. imSituVQA dataset samples about cooking, buying, catching and opening. The imSituVQA dataset includes frame element annotations for each question answer pair.

Frame element	frequency
PLACE	100,006
AGENT	49,976
ITEM	24,376
TOOL	13,908
VICTIM	3,932
TARGET	3,860
VEHICLE	3,706
DESTINATION	3,238
COAGENT	2,544
OBJECT	2,317

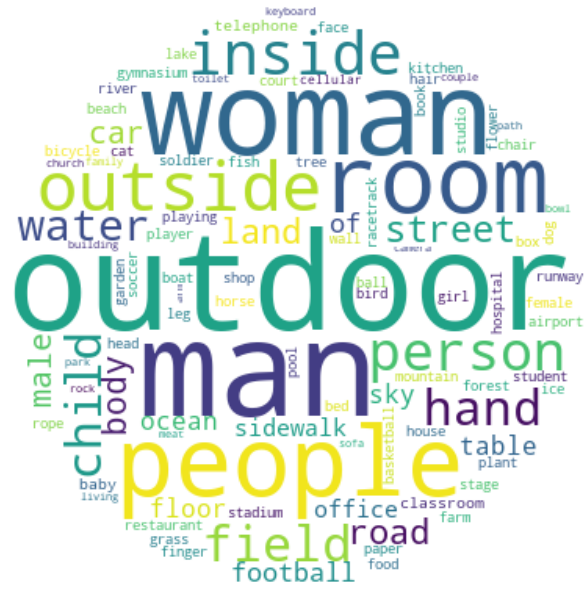
TABLE VII

Top 10 frequent frame elements in imSituVQA training samples.

Answer	frequency
outdoors	14,621
man	13,527
woman	10,763
people	9,228
room	8,323
outside	6,881
inside	6,679
person	5,625
hand	4,238
field	3,086

TABLE VIII

Top 10 frequent answers in imSituVQA training samples.



(a)

(b)

Figure 15. imSituVQA word clouds of (a) answers and (b) frame elements.

CHAPTER 5

A NOVEL MULTI-TASK VQA MODEL USING SEMANTIC FRAME INFORMATION (PREVIOUSLY PUBLISHED AS M. ALIZADEH AND B. DI EUGENIO. (ICSC 2020) AUGMENTING VISUAL QUESTION ANSWERING WITH SEMANTIC FRAME INFORMATION IN A MULTITASK LEARNING APPROACH. PAGES 37 – 44)

This chapter explains the proposed VQA model in a multi-task learning paradigm. The idea of multi-task with neural networks learning is to jointly train multiple tasks within a shared architecture up to the classification layer (Caruana, 1997). Multi-task learning has been utilized in deep learning in different applications. For example, Collobert and Weston (Collobert and Weston, 2008) proposed a neural network based multi-task learning method for NLP. This method jointly trains multiple NLP classification tasks, e.g., part-of-speech tagging, named entity tagging, semantic role labeling, etc. Seltzer and Droppo (Seltzer and Droppo, 2013) employed multi-task learning in neural networks in order to improve phoneme recognition. It has been shown that multi-task learning can boost the generalization of shared tasks. Traditional multi-task learning transfers knowledge by sharing lower level features.

(Xie et al., 2015) discusses challenges in fine-grained image classification (FGIC). The goal of FGIC is to recognize objects that are visually and semantically similar to each other. For example classifying cars to their specific models such as Chevrolet, Toyota and so on. Fine tuning

a convolutional neural network (CNN) works well for general visual recognition datasets such as ImageNet (Deng et al., 2009). But because of large intra-class and small inter-class variance, it may not work well for FGIC. The authors proposed a framework known as hyper-class augmented and regularized deep learning. They suggest a hyper-class augmentation formulated as multi-task learning in order to boost the recognition task in FGIC. Inspired by their ideas, I formulate the VQA task as a multi-task learning problem. In this formulation, the system should learn to classify answers as well as frame elements. (Xie et al., 2015) also proposed a more complex hyper-class multi-task learning called hyper-class augmented regularized (HAR) deep model. The idea is to link the output of the hyper-class classification layer to the fine-grained classification layer. For the simplicity of the implementation, I preferred the first model over the second one.

5.1 Proposed VQA model

Let $D_t = \{(x_1^t, y_1^t), ..., (x_n^t, y_n^t)\}$ be a set of training <*image, question*> paired samples with $y_i^t \in \{1, ..., C\}$ indicating the answers (e.g., *woman*, *kitchen* and *cooking*) of <*image, question*> pair x_i^t , and let $D_a = \{(x_1^a, r_1^a), ..., (x_n^a, r_n^a)\}$ be a set of auxiliary frame element information, where $r_i \in \{1, ..., R\}$ indicates the frame element of <*image, question*> pair x_a^t (e.g., *agent*, *food* and *location*). The goal is to learn a VQA model that correctly answers to an input <*image, question*> pair. In particular, the goal is to learn a prediction function given by $Pr(y|x)$, i.e., given the input *x:*<*image, question*> pair, the probability that y is the answer is computed. Similarly, $Pr(r|x)$ denotes the frame element classification model. Given the training <*image,*

question> pairs and the answers with auxiliary frame element information, our strategy is to train a multi-task deep VQA model. This model can use any arbitrary VQA architecture up to the classification layer. Then sharing common features, it branches out to two different classifiers. One classifier classifies answers, and the other one, frame elements. Figure 16 summarizes the proposed multi-task learning model. In order to train the proposed VQA model, the total loss is the average of losses from these two classifiers.

$$\text{total loss} = \frac{1}{2} \times [\text{loss}(\textit{answers classification}) + \text{loss}(\textit{frame elements classification})] \tag{5.1}$$

5.2 Evaluation

In this section, I evaluate the proposed VQA model on imSituVQA. The goal is not necessarily to optimize the hyper-parameters or the design of the feature layers but rather to focus the attention on learning strategies supported by linguistic facts.

5.2.1 Baselines

The following baselines are computed:

1. **prior ("outdoors")**: The most popular answer ("outdoors").
2. **per verb prior**: The most popular answer per verb (for example cooking ("kitchen"), buying ("man"), reading ("book")).

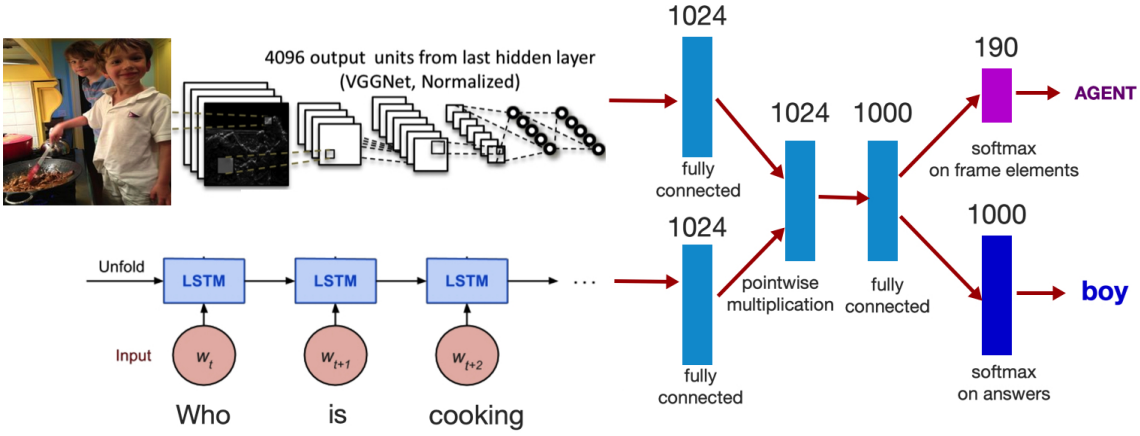


Figure 16. Proposed multi-task learning architecture for VQA

5.2.2 Experimental Setup

The proposed VQA model is evaluated by means of the CNN-LSTM-based architecture introduced in (Antol et al., 2015). Training deep models require significant time and resources. Consequently, trained models such as GLOVE (Pennington et al., 2014) and VGG-NET (Simonyan and Zisserman, 2015) are employed. GLOVE provides a good word embedding layer initialization in order to generalize well and get a performance boost. GLOVE 300-dimensional weights are utilized in order to feed question words to a bidirectional long short term memory network (LSTM). The output of the LSTM is a 300 dimension question embedding which is mapped to the 1024 dimensions by passing through a nonlinear layer. A VGG-NET-16 pre-trained model was used in order to extract image feature vectors. The 4096 image embedding is

mapped to 1024 dimensions by passing it through a nonlinear layer. The multimodal fusion of image and question embeddings occurs via pointwise multiplication, then after passing through a number of nonlinear layers, the final embedding is fed to the frame element softmax layer and the answer softmax layer. The model is trained by minimizing the sum of the two cross entropy loss functions using the rmsprop (Tieleman and Hinton, 2012) optimization algorithm. The training data is passed with a batch size of 500 in 50 epochs.

5.2.3 Results and Discussions

Table IX shows the performance evaluation on the test samples. Using the most frequent answer (prior) to answer each question results in 5.65% accuracy. Selecting the most frequent answer per verb results in 22.15% accuracy. The CNN-LSTM model trained with single answer softmax results in 38.08% accuracy. The multi-task CNN-LSTM model which includes both answer softmax and frame element softmax achieves an accuracy of 43.89%. Augmenting VQA with frame element information boosts the accuracy by up to 5%. This improvement in the generalization of the CNN-LSTM model indicates how well the multi-task approach acts as a regularizer. A chi-square test is performed in order to show statistically significant improvement of the model (Table X).

Performance can be compared in terms of **WUPS** as well. Wu-Palmer Similarity (Malinowski and Fritz, 2014) can be used as an alternative to accuracy (Wu and Palmer, 1994). It is based on how semantically the predicted answer matches the ground truth. Given a predicted answer and a ground truth answer, WUPS computes a value between 0 and 1 based on their similarity. As

shown in Figure 17, It computes similarity by using the depths of the two synonyms in WordNet, beside the depth of the LCS (Lowest Common Ancestor). For example *WUP(land,earth)* is 1.0 while *WUP(tree,water)* is 0.14. "WUPS at 0.9" applies a threshold and considers a predicted answer correct if the WUPS score is higher than 0.9. For example *WUP(dog,wolf)* is 0.93 and answering *wolf* instead of *dog* is considered correct in terms of "WUPS at 0.9". Table IX shows performance of the proposed model on the imSituVQA in terms of "WUPS at 0.9". The multi-task approach results in improvements for both accuracy and "WUPS at 0.9" of about 5%.

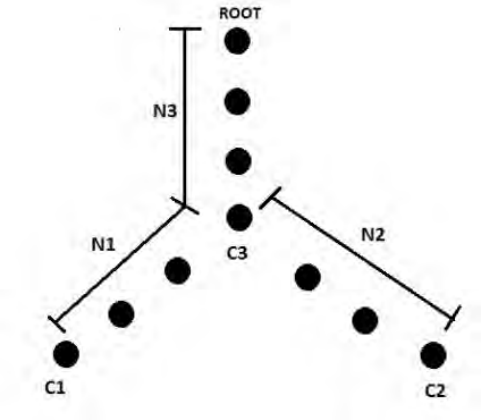


Figure 17. Wup is a WordNet based similarity measure. $\mathsf{WUP\ similarity}(C_1, C_2)$ is measured by $\frac{2\times N3}{N1+N2}$ where C_3 is the least common subsumer of C_1 and C_2. WUPS employs $\mathsf{WUP\ similarity}$ between synsets of C_1 and C_2 in order to compute a fuzzy based similarity. The detailed formulation is discussed in (Malinowski and Fritz, 2014)

	Accuracy (%)	WUPS at 0.9 (%)
prior ("outdoors")	05.68	11.87
per verb prior	22.15	27.65
CNN-LSTM	39.58	46.92
multi-task CNN-LSTM	**44.90**	**51.83**

TABLE IX. Accuracy of our VQA model on imSituVQA dataset

	Correct	Incorrect
CNN-LSTM	34905	53065
multi-task CNN-LSTM	39522	48448

TABLE X. The chi-square statistic is 496.1854. The p-value is < 0.01 and the result is significant.

Fine-grained evaluation. In order to perform fine-grained evaluation, performance per question, per verb and per role are computed. Figure 18 shows a performance comparison

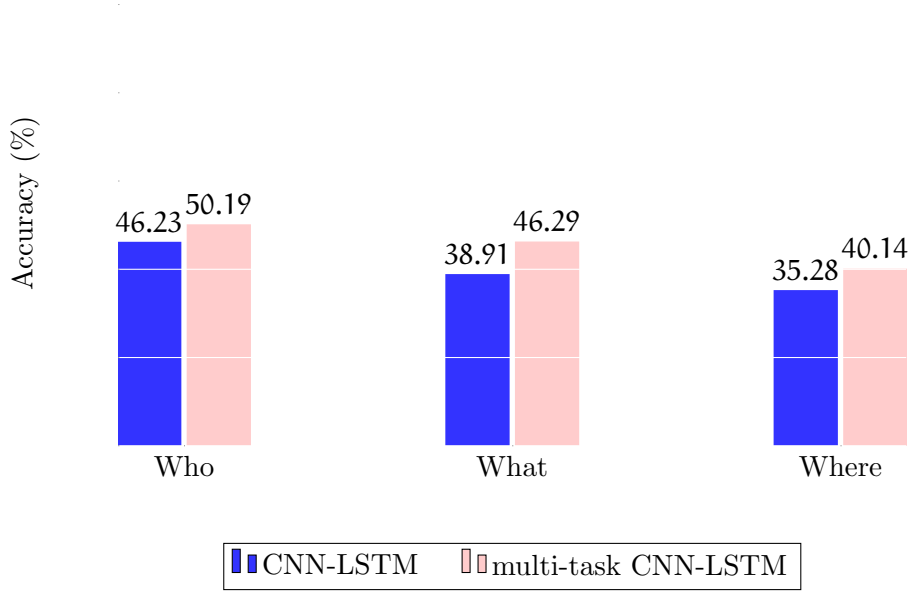


Figure 18. Evaluation by first question words

based on first question words. Multi-task CNN-LSTM performs better for who (4%), what (8%) and where (5%) when compared to CNN-LSTM. Performance per role and per verb are included in the appendix (for example *cooking* improves from 30.12% to 44.58% and *buying* from 27.42% to 64.52%). Performance per role is included in the appendix (for example the multi-task approach improves *AGENT* from 48.78% to 52.29%, *PLACE* from 34.75% to 39.52% and *ITEM* from 32.27% to 39.65%). Table XI shows a different view of the performance difference between CNN-LSTM and the multi-task version. About 55% of verbs improve by less than 10%. *whipping, buying, sketching, sketching, scooping, making* improve by more than 30% of improvement. *spanking, ejecting, farming, hitting, harvesting, moistening* decline by more than

15%. *payment, rammingitem, boaters, stage, undergoer, strapped, resource* have more than 40% of improvement. *removeditem, beneficiary, eater, planted, blocker, aspect* decline by more than 25%.

Accuracy Difference Range	Verb Frequency	Role Frequency
(-40%,-30%]		3
(-30%,-20%]	2	3
(-20%,-10%]	10	5
(-10%,0%)	67	24
0%	27	32
(0%,10%]	269	68
(10%,20%]	100	24
(20%,30%]	15	13
(30%,40%]	6	
(40%,50%]		4
(50%,60%]		2
...		
100%		1

TABLE XI. Performance evaluation grouped by performance intervals showing verb frequency and role frequency in each group.

It is very hard to extrapolate as concerns the reasons why some verbs improve and others don't; there doesn't seem to be any apparent generalizations about classes of verbs (like action verbs vs communication verbs and the like). There are subtle interplays between frequencies of slot fillers and verb samples that don't lend themselves to generalizations. For example, let's consider sketching. Table XII shows imSitu sample images about *sketching* where *MATERIAL* annotated with paper. *"What material does the AGENT sketch on?"* is one of the templates generated during imSituVQA creation process. The response frame element is *MATERIAL*. For many realized questions of this template, the answer is paper. The multi-task CNN-LSTM most of the time answers *paper* at test time while the CNN-LSTM makes more mistakes. For subset of samples about *sketching*, the multi-task CNN-LSTM model's performance is 61.54% comparing to 26.92% for the CNN-LSTM model. In general, for the samples where response frame element is *MATERIAL*, the multi-task CNN-LSTM model's performance is 50.88% comparing to 28.07% for the CNN-LSTM model.

TABLE XII. imSitu sample images about sketching where the *MATERIAL* is annotated as paper.

Frame element classification: The hyper-class augmentation model utilizes frame element classification for better representation learning of the VQA task. As discussed earlier, the accuracy of the frame element classification is 99.32%. One important reason for such high performance is the frame element dependency on the input question while it is independent of the input image. For example for the question *"who is cooking ?"* the frame element is always *AGENT* for all images about cooking. This results in a huge amount of data to train the frame element classification resulting in almost perfect performance.

It is interesting to know how frame element classification affects the predicted answer and how consistent it is with the correct answer and predicted answer. We consider the correct or predicted answer to be consistent with the frame element if there is at least one training sample labeled with both the answer and the frame element. For example <*bear, AGENT*> and <*bear, CHASEE*> are consistent but <*bear, PLACE*> and <*bear, TOOL*> are inconsistent. Figure 19 shows the frequency of distinct frame elements for a subset of answers. For example *man, car, telephone, bear* and *cafe* are fillers of 81, 37, 20, 8 and 1 distinct frame elements in the training samples respectively. An answer is consistent with the set of distinct frame elements it fills and inconsistent with others.

The almost perfect accuracy of the frame element classifier confirms its output is almost always consistent with the correct answer. Now the question is, how much does frame element classification help the predicted answer to be consistent with the semantic frame? Employing the consistency criterion, the consistency of the CNN-LSTM model is 97.56% and multi-task CNN-LSTM 99.94%. This shows a 2.38% improvement. In other words, augmenting the frame

element classification decreases inconsistency in providing final responses. Consequently, the end-user would get more reasonable answers from the system. Table XIII shows imSituVQA test samples where the multitask CNN-LSTM VQA model answers correctly while the answer of CNN-LSTM VQA model is wrong and inconsistent.

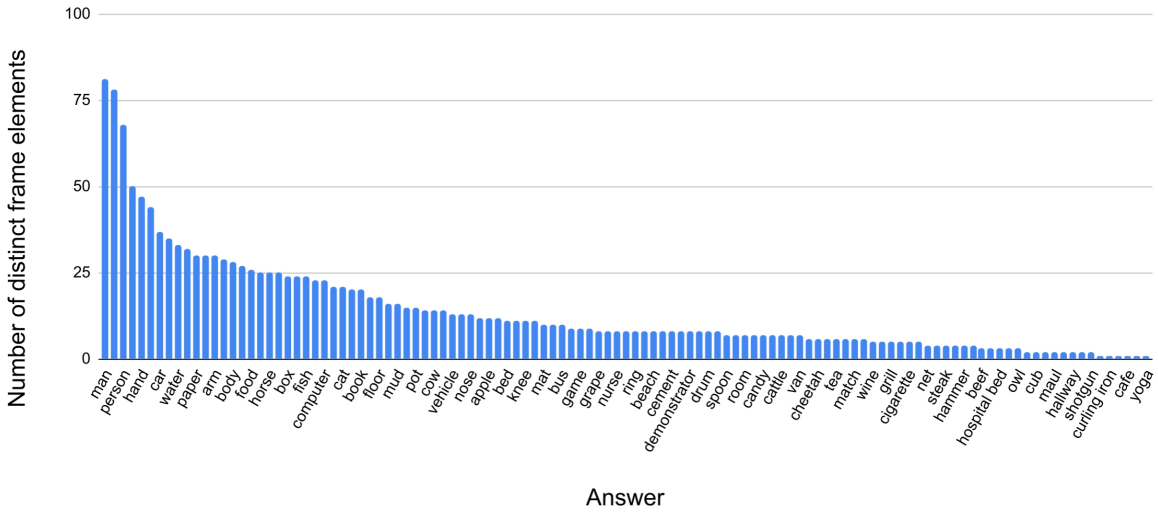


Figure 19. Distinct frame element frequency for different answers.

QUESTION	QUESTION
what container does the woman use to cook?	what does the person stitch with sewing machine?

ANSWER/RFE	CNN-LSTM	Multitask CNN-LSTM	ANSWER/RFE	CNN-LSTM	Multitask CNN-LSTM
bowl/CONTAINER	telephone	bowl	thread/FASTENER	paint	thread

QUESTION	QUESTION
what does the person use to slice seaweed?	where does the cowboy ride the bull?

ANSWER/RFE	CNN-LSTM	Multitask CNN-LSTM	ANSWER/RFE	CNN-LSTM	Multitask CNN-LSTM
knife/TOOL	hand	knife	rodeo/PLACE	adult	rodeo

TABLE XIII. Comparing the predictions of CNN-LSTM with multitask CNN-LSTM . These examples show the cases where the multitask CNN-LSTM VQA model answers correctly while the answer of CNN-LSTM VQA model is wrong and inconsistent. RFE stands for response frame element.

CHAPTER 6

AUTOMATIC SEMANTIC ROLE LABELING OF THE VQA DATASET

The imSituVQA is created based on accurate annotations. We would like to train the proposed multi-task CNN-LSTM model with currently available VQA datasets such as the VQA dataset (Antol et al., 2015) as well. The ideal solution is to manually annotate the VQA samples. However, this approach is expensive and time-consuming. An alternative approach is to use an automatic semantic labeler. In this way we can have any question of interest approximately annotated.

Semantic Role Labeling (SRL) is the process of detecting semantic arguments associated with the verb (or any word as a predicate) of a sentence and their classification into their semantic roles. In 3.1, I explained the concept of semantic roles and available resources such as FrameNet and PropBank. In this section, I describe the outputs of two semantic role labelers: (1) Open-SESAME (Swayamdipta et al., 2017) (trained on FrameNet) and (2) ClearNLP (Choi and Palmer, 2012) (trained on PropBank). As discussed later in this chapter, we found that PropBank based labeler is better suited for the multi-task CNN-LSTM VQA modeling.

6.1 FrameNet based semantic role labeler

Open-SESAME (Swayamdipta et al., 2017) is a frame-semantic parser. It automatically detects FrameNet frames and their frame-elements in a sentence. The project is a pipeline of

Target	Frame	Arguments
what.n	Entity	
be.v	Performers_and_roles	he/B-Medium
		throwing/I-Medium
throw.v	Body_movement	what/B-Depictive
		is/I-Depictive
		he/I-Depictive
		throwing/S-Message
		?/S-Depictive

TABLE XIV. The output annotation of SESAME for *"what is he throwing ?"*.

three primary tasks: *target identification* (which words or expressions evoke frames?), *frame identification* (which frame does each target evoke?), and then *argument identification* (for each frame f, and each of its possible roles in FrameNet, which span of the text provides the argument?). Table XIV shows the output of Open-SESAME on *"what is he throwing ?"*. As can be seen, three lexical units result in the evoking of three semantic frames. *what.n* evokes *Entity*, *be.v* evokes *Performers_and_roles* and *throw.v* evokes *Body_movement*. The third column shows the frame elements of each frame with its span in the text. Generally, each frame has a possible

set of frame elements but only a few of them are taking part in the sentence.

For many questions starting with *"what"*, if it is a target word then it would evoke the *Entity* frame. This is very general to be used as a discriminator in our VQA model. On the other hand, the annotations do not directly provide information about the response role to be used for the auxiliary classification task (similar to frame element classification).

6.2 PropBank based semantic role labeler

ClearNLP (Choi and Palmer, 2012) employs a dependency parser and feature-based algorithm and it is trained on PropBank. The PropBank annotations include a trace of a wh-phrase in questions. The annotation includes a *"R-A#"* pattern which provides useful information regarding the argument of the question. " *Who is sitting on the bench?*" is labeled as " *V:sit.01 A2:on* ***R-A1:who***". *"R-A1"* is an indicator of the argument of the question which is Arg1 of the first sense of the verb sit (Table II). " *What are the people watching?*" is labeled as " *V:watch.01 A0:people* ***R-A1:what***"."*R-A1"* is an indicator of the argument of the question which is Arg1 of the first sense of the verb watch (Table II). Arg1 is a proto-patient role as described in the background chapter. This *"R-A#"* pattern is the advantage of PropBank over FrameNet in the proposed multi-task VQA modeling.

6.3 Semantic Role Labeling of the VQA Dataset

Semantic role labeling tools capture different information. In general, this information might be useful for the task of VQA. But the multi-task CNN-LSTM model, requires semantic

A0 (2541)	**A1** (29105)	**A2** (1906)	**A3** (40)	**A4** (234)
Agent (1844)	Theme (14752)	Material (767)	Staring	Ending
Pivot (384)	Patient (6510)	Location (445)	Point (40)	Point (234)
Instrument (104)	Agent (2322)	Instrument (408)		
Cause (87)	Topic (2159)	Recipient (90)		
Theme (68)	Value (1036)	Destination (76)		

TABLE XV. Non-modifier abstract arguments frequency in the VQA_{sub} training samples. Each column shows the top 5 SemLink mappings of the abstract role with frequencies inside parenthesis.

labels of answers. This is the reason PropBank based semantic role labeler is preferred over FrameNet based semantic role labeler. Given the PropBank annotation, *R-* indicates which verb argument the question refers to. These references can be used as hyper-classes similar to response frame elements in imSituVQA. So the CNN-LSTM model can be augmented with semantic role information in the multi-task learning paradigm.

Since the labels provided by PropBank are abstract, I utilized SemLink (Palmer, 2009) in order to map PropBank (Palmer et al., 2005) to VerbNet (Levin, 1993). Table XV shows a more fine-grained mappings of non-modifier arguments (A0 to A4 excluding AM(Argument Modifier)). If the annotations provided by SemLink include mapping from a specific sense of a verb, then it

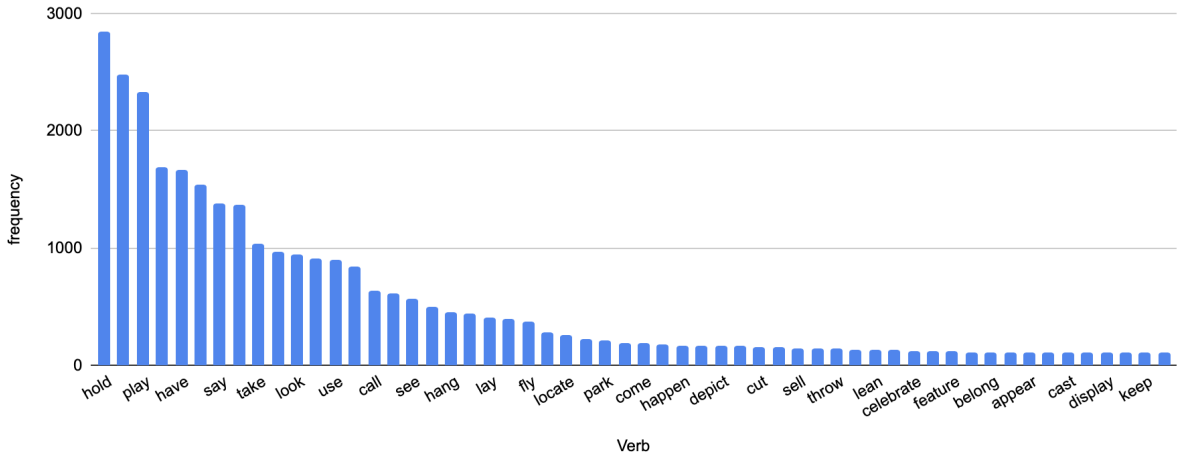


Figure 20. Verb frequencies in the VQA_{sub} training samples. hold (2844), wear (2475), play (2334), make (1688), have(1663), sit(1544), say(1380), show(1368), take(1032) and stand(973) are 10 most frequent verbs.

is used otherwise default mapping is applied. For example *Arg2* of *fall.01* is mapped to *Extent* while *Arg2* default mapping is *Instrument*.

The VQA dataset samples fall in three general categories: (1) *Yes/No* (2) *Number* (3) *Other*. Since the semantic role labels deal with "Who did what to whom" (and perhaps also "when and where"), I decided to focus on the *Other* category subset of the VQA dataset. This subset has about 120k training samples. Some of the annotations do not include *"R-A"* pattern. For example *"What food is being served?"* is labeled as *"V:serve.02 A2:food"* and *"What brand of beer does the sticker on the door feature?"* is labeled as *"V:do.02 A0:brand A1:sticker"*. *"R-A"* is essential for hyper-class augmentation modeling and the subset was filtered to include

Labeled Question	**SemLink Mapping**
Who makes the blue and red trucks?	
V:make.01 A1:truck **R-A0:who**	Agent
What is the girl dragging behind her?	
V:drag.01 A0:girl A2:behind **R-A1:what**	Theme
What kind of pants does the woman wear?	
V:wear.01 A0:woman **R-A1:kind**	Theme
From where in the room is the light coming?	
V:come.01 A1:light **R-AM-LOC:where**	Location
When will the red light turn on?	
V:turn.13 A1:light AM-MOD:will **R-AM-TMP:when**	Temporal
How does the man get the horse to move where he wants?	
V:get.04 A0:man A1:move **R-AM-MNR:how**	Manner

TABLE XVI. Samples of PropBank based semantic role labeling of the VQA dataset

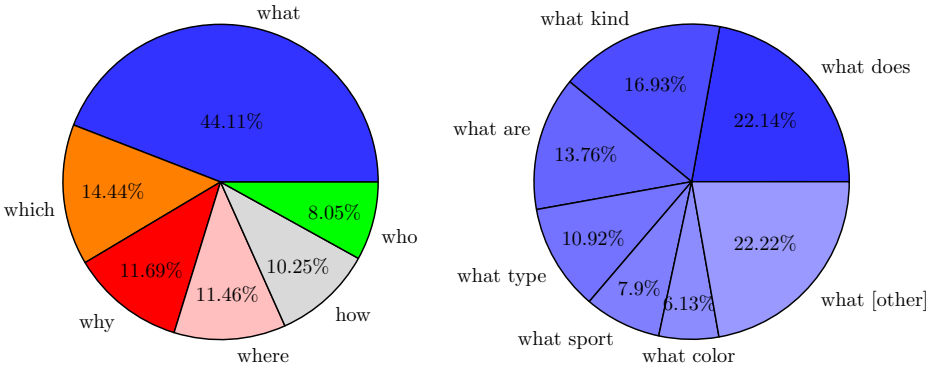


Figure 21. Distribution of questions in the filtered VQA dataset. (a) covers all questions while (b) includes questions starting with the question word "what"

this information. I filtered *"what is the AGENT doing ?"* type of questions as well. The tool labels it as *"V:do.02 A0:AGENT* ***R-A1:what****"*. The SemLink maps *A1* to *PATIENT*. Since the appropriate response would be of type *predicate* or *event* ("swimming"), I decided to filter these samples as well. Therefore, filtering out these samples resulted in 41k training samples and 21k test samples. I call the final version VQA_{sub}. Table XVI shows a sample annotation of a number of the VQA_{sub} samples. Using the *"R-A"* pattern and SemLink, the auxiliary gold standard output of the multi-task CNN-LSTM model is extracted. Table XVII shows the list of the top 10 frequent semantic roles. Table XVIII shows the list of the top 10 frequent answers. Figure 21 shows the distribution of questions in the VQA_{sub}. There are 40 unique semantic roles for the auxiliary classification task. Each VQA sample is augmented with semantic role information in order to train the multi-task CNN-LSTM VQA model.

Semantic Role	Frequency	Relative Frequency (%)
Theme	116340	39.07%
Patient	7585	18.14%
Agent	4620	11.05%
Location	2945	7.04%
Topic	2364	5.65%
Cause	1705	4.08%
Value	1143	2.73%
Material	846	2.02%
Manner	830	1.98%
Product	822	1.97%

TABLE XVII. Top 10 Semantic Roles in the VQA_{sub}. It also includes frequency and relative frequencies of each semantic role.

Given the augmented VQA_{sub} samples, the task is to train a VQA model to get a paired input of <*image, question*> for answer classification as well as semantic role classification. Figure 22 shows the updated CNN-LSTM VQA architecture. The example input question is *"What is the girl dragging behind her?"*. Interestingly the verb *"drag"* also exists in imSitu with the abstract definition: *"AGENT drags an ITEM with a TOOL on a CONTACT at a PLACE"*. The semantic role response here is *Theme* while in the context of imSitu it is *ITEM*.

Answer	Frequency	Relative Frequency (%)
tennis	725	1.73
baseball	713	1.7
wood	688	1.64
frisbee	676	1.62
right	570	1.36
left	526	1.26
skateboard	381	0.91
wii	374	0.89
grass	370	0.88
pizza	353	0.84

TABLE XVIII. Top 10 Answers in the VQA_{sub}. It also includes frequency and relative frequencies of each answer.

The answer is annotated by 10 people consequently 10 answers: 7 *plaid*, 1 *striped*, 1 *shorts* and 1 *multi-colored*. For training, the answer with the highest frequency is chosen as gold standard output. At validation time the VQA accuracy of the predicted answer is computed based on $\mathsf{accuracy}(\mathsf{answer}) = \mathsf{min}\{1, \mathsf{freq}(\mathsf{answer})/3\}$. For example *accuracy(plaid)* is 1 and *accuracy(striped)* is 0.33 for the given sample.

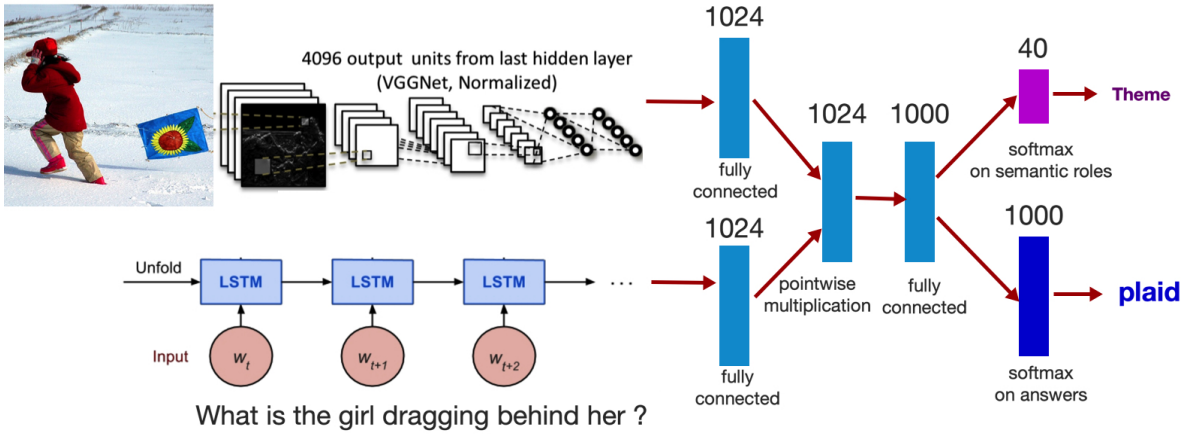


Figure 22. Multi-task CNN-LSTM VQA with semantic roles as hyper-classes

Table XIX shows the results. CNN-LSTM model accuracy is 19.89%. Multi-task CNN-LSTM model has two loss functions to be optimized and the final loss function is the weighted some of the two: $w_1 \times$ *loss(answer classification)*$+w_2 \times$ *loss(role classification)*. I experimented with three configurations (0.5,0.5), (0.8,0.2) and (0.9,0.1). (0.5,0.5) does not work well since it degrades answer classification performance. Lowering the weight of the role classification results in lower role prediction accuracy but a better answer classification until I set it to 0.8. (0.8,02) is the only setup slightly improving the performance over CNN-LSTM model.

I also performed a fine-grained analysis of the top frequent semantic roles and the results are shown in Table XX. *Product*, *Value*, *Manner* and *Topic* are improved considerably. There is a slight improvement (less than 0.15%) for *Patient*,*Agent* and *Theme*. However the performance

Model	Loss weights (answer,role)	Answer Accuracy (%)	Role Accuracy (%)
CNN-LSTM	-	19.89	-
Multi-task CNN-LSTM	(0.5,0.5)	15.02	80.57
Multi-task CNN-LSTM	(0.8,0.2)	20.08	63.20
Multi-task CNN-LSTM	(0.9,0.1)	19.57	38.23

TABLE XIX. Accuracy of the proposed CNN-LSTM VQA model on the VQA_{sub}. Three different weightings were applied in order to check the effect of multi-task paradigm.

declines for *Location* and *Cause*. Table XXI shows the analysis in terms of non-modifier abstract arguments. Interestingly the improvement is 0.8% as compared to 0.2% over all of the abstract roles. As shown in Table XV non-modifier abstract arguments are mapped to different set of VerbNet roles. A0 is mostly mapped to Agent (37%) then to Pivot (7%). The rest of the mappings are less than 2% frequent. The improved performance of the model is good for Agent but it is a little negative for A0 (-0.05). It shows the A0 samples labeled with a mapping other than Agent result in more confusion and poor performance of the model for A0. A1 is mapped mostly to Theme and Patient most of the time (35%). Agent and Topic are the next frequent mappings with 4% of the times each. The mapping is vaguer comparing to other abstract roles. But the performance is improved anyway by 0.12%. A2 is mapped to Location, Material, Instrument more than 85% of times. The improvement for A2 is 2.26 showing the model is

Semantic Role	Single-task (%)	Multi-task (%)	Improvement (%)
Theme	23.24	23.29	0.05
Patient	18.18	18.32	0.14
Agent	18.62	18.73	0.11
Location	14.33	14.21	-0.12
Topic	11.54	11.84	0.3
Cause	15.58	15.34	-0.24
Value	21.66	22.1	0.44
Manner	14	14.43	0.43
Product	32.72	33.83	1.11

TABLE XX. Fine-grained evaluation of frequent semantic roles.

Argument	Single-task (%)	Multi-task (%)	Improvement (%)
A0	18.34	18.29	-0.05
A1	20.71	20.83	0.12
A2	27.21	29.47	2.26
A3	24.55	21.49	-3.06
A4	6.95	8.48	1.53

TABLE XXI. Fine-grained evaluation of abstract arguments (excluding modifiers).

doing well for questions asking about roles mapped from A2. A3 and A4 have just one mapping each. The proposed model degrades the performance for A3 by -3.06 while improving for A4 by 1.53. Because the number of samples is low for A3, any misclassified sample would result in a significant error.

Table XXII shows the top 10 frequent verbs along with a list of their most frequent arguments. Interestingly for all of the verbs, A1 is the most frequent argument in question. I also analyzed the performance for the top frequent verbs and the results are shown in Table XXIII. Exploring the top 10 the proposed multi-task model seems to work better for concrete verbs. For example *play* and *eat* are improved while *show* and *say* are degraded. I explored the output of the two models for test samples with *play* as the primary verb, multi-task version improvement is considerable on *"what [sport] is the AGENT playing?"* type of questions. I repeated the same process for *eat*. The multi-task model performs better on *"what does the AGENT eat?"* type

Verb.sense	rank1 arg	rank2 arg	rank3 arg	rank4 arg	rank5 arg
wear.01(1227)	A1(990)	A0(114)	AM-CAU(93)	AM-MNR(16)	AM-LOC(14)
play.01(1196)	A1(1121)	AM-LOC(55)	A0(14)	AM-CAU(4)	AM-MNR(1)
hold.03(916)	A1(913)	AM-CAU(2)	AM-MNR(1)		
have.03(887)	A1(628)	A0(164)	AM-CAU(89)	AM-LOC(5)	AM-MNR(1)
sit.01(813)	A1(675)	A2(112)	AM-CAU(21)	AM-MNR(5)	
show.01(801)	A1(790)	A0(6)	AM-CAU(3)	AM-LOC(2)	
make.01(789)	A1(375)	A2(365)	A0(37)	AM-LOC(6)	AM-MNR(4)
say.01(775)	A1(769)	A2(3)	AM-CAU(2)	AM-MNR(1)	
eat.01(541)	A1(484)	A0(28)	AM-LOC(14)	AM-CAU(7)	AM-TMP(5)
stand.01(526)	A1(407)	A2(85)	AM-PRP(22)	AM-CAU(10)	AM-MNR(1)

TABLE XXII. top 10 verbs (with sense index) in the VQA_{sub} test samples along their response arguments sorted by frequency. The numbers inside parenthesis indicate the frequency.

of questions.

Table XXIV shows fine-grained analysis on first question-words. The highest improvement is for *"who"* while the worst one is *"which"*. Table XXV narrows down the questions starting with *"what"*. The highest improvement is for *"what sport"* while the worst one is *"what time"*. It might be surprising to see *what color* in the list because we expect to have *"what color is*

Verb.sense	Single-task (%)	Multi-task (%)	Improvement (%)
wear.01	26.22	26.35	0.13
play.01	34.47	35.02	0.55
hold.03	11.94	12.06	0.12
have.03	19.44	19.38	-0.06
sit.01	17.62	17.34	-0.28
show.01	18.38	18.19	-0.19
make.01	37.11	36.88	-0.23
say.01	12.34	12.31	-0.03
eat.01	26	26.25	0.25
stand.01	16.01	15.95	-0.06

TABLE XXIII. Performance of the top 10 verbs (with sense index) in the VQA_{sub} test samples.

[object]?" type of questions. However there are samples in VQA_{sub} with verbs such as *"What color light is lit on the traffic light?"* or *"What color is the frisbee the woman is putting down?"*.

Question word	Single-task (%)	Multi-task (%)	Improvement (%)
what	23.97	24.48	0.51
which	29	28.61	-0.39
why	16.33	16.04	-0.29
where	15.24	16.12	0.88
how	18.36	19.13	0.77
who	24.36	26.04	1.68

TABLE XXIV. Fine-grained evaluation of question words.

what ...	Single-task (%)	Multi-task (%)	Improvement (%)
what does	16.48	16.34	-0.14
what kind	16.48	16.72	0.24
what are	21.85	21.48	-0.37
what type	24.91	25.38	0.47
what sport	48.52	49.69	1.17
what color	24.2	25.04	0.84
what animal	18.57	19.51	0.94
what brand	28.8	29.78	0.98
what time	43.54	41.1	-2.44

TABLE XXV. Fine-grained evaluation of question words starting with what.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

In this report, I explained the need for a VQA dataset annotated with verb semantic information. I exploited imSitu dataset annotations to create a new VQA dataset with frame semantic information. This process involved two phases: (1) question, answer template generation (2) question, answer pair realization. In the first step, I exploited imSitu abstract verb definitions to generate question, answer templates. In the second step, I employed imSitu annotations to create the novel VQA dataset called imSituVQA. I also performed a distributional analysis of the imSituVQA to show the properties of the newly created dataset. This work is published in the proceedings of the *12th Language Resources and Evaluation Conference (LREC)*. Adapting ©ACL 2020, I reused many parts of the paper in my thesis (Appendix C).

The novel imSituVQA dataset is available online. [1] The imSituVQA questions are generated via a semi-automatic process. Consequently, the naturalness of the questions is not guaranteed. There are no automatic metrics and it required human judgment. One solution might be to adapt evaluation metrics from other domains. For example, (Zhang et al., 2019) proposed BERTScore, an automatic evaluation metric for text generation. The authors showed the metric is working very well for machine translation and caption generation comparing to other metrics.

---

[1] https://github.com/givenbysun/imSituVQA/

Then the question was how I could exploit this frame semantic information in a VQA system. I explored different research to figure out a solution. (Xie et al., 2015) proposed a framework known as hyper-class augmented and regularized deep learning for better fine-grained image classification. In the context of VQA, I employed frame semantic information as hyper-classes. I formulated the VQA task as a multi-task learning problem. In this formulation, the system should learn to classify answers as well as frame elements. I evaluated the proposed idea with the popular CNN-LSTM VQA modeling. This approach boosts performance and shows the benefit of using verb semantics in answering questions about images. This work is published in the proceedings of the *14th IEEE International Conference on Semantic Computing (ICSC)*. [1]. Adapting ©IEEE 2020, I reused many parts of the paper in my thesis (Appendix C).

Manual annotation is a time-consuming and expensive process. Automatic semantic role labeling is an alternative solution to employ the proposed model for any VQA dataset of interest. I employed ClearNLP (PropBank based semantic role labeler) to label a subset of the VQA dataset (VQA_{sub}). The *"R-"* pattern was used as an indicator of the response semantic role. I performed a distributional analysis of the VQA_{sub} to show and visualize its properties. I employed the proposed multi-task CNN-LSTM model for training and testing. The results show a slight improvement over the single CNN-LSTM model. I also performed several fine-grained evaluations for further analysis. A summary of this thesis including this work will be published in the *International Journal of Semantic Computing (IJSC)*.

---

[1] This paper was nominated for *Best Paper Award*

In this work, I employed exclusively the CNN-LSTM architecture proposed in (Antol et al., 2015). The two modalities are fused via multiplication. This fusion can be implemented by concatenation or bilinear pooling and so on. Different CNN models are proposed in the literature. Beside VGGNet (Simonyan and Zisserman, 2015) (pre-trained on ImageNet), I experimented with ResNet (He et al., 2016) (pre-trained on ImageNet) in order to extract image features. The results are similar. LSTM is a very popular sentence embedding approach and works better than traditional approaches such as *bag of words*. Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is one of the recent works that can be employed for question embeddings. BERT is an approach for pre-training. The model is trained for general representation that has been successfully fine-tuned for many natural language understanding applications. In future work, BERT could be explored as an alternative embedding model to LSTM.

7.1 Employing a new set of hyper-classes

Another interesting extension to this work would be employing a new set of hyper-classes. The idea is to apply hyper-class augmentation to already available VQA datasets. For example the type of task that a question refers to such as *object presence*, *object attributes*, *counting* can be included as hyper-classes.

The VQA dataset answers are categorized into *yes/no*, *number* and *other*. According to the proposed multitask CNN-LSTM modeling I added a softmax layer with three classes to the

CNN-LSTM VQA model. Then I trained the model with the VQA dataset. Unfortunately, the results show that augmentation did not help much in improving performance.

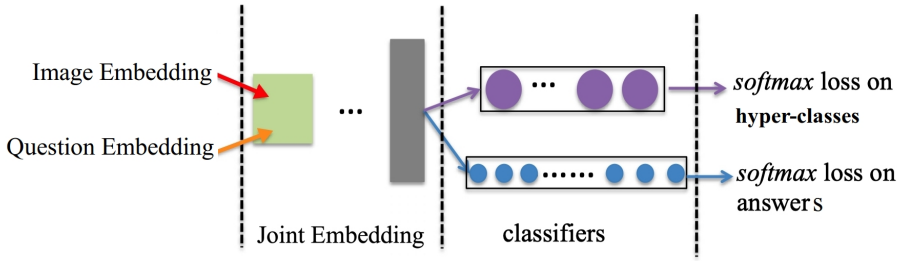


Figure 23. Hyper-class augmented VQA model using multi-task learning

The Task Driven Image Understanding Challenge (TDIUC) (Kafle and Kanan, 2017) includes 1.6 million questions categorized into 12 different unbiased (uniformly distributed) visual tasks (Figure 24). The task types can be included as hyper-classes. I also experimented with these 12 tasks as hyper-classes and could not achieve a significantly better result.

(Liu et al., 2019) proposed a Multi-Task Deep Neural Network (MT-DNN) showing the regularization effect of multi-task learning in NLP. The main goal of MT-DNN is to train a model across multiple natural language understanding (NLU) tasks. They designed the model so that 4 different tasks be optimized at the same time. Similarly, the idea of hyper-class augmentation can be extended to include more than two tasks. For example answer classification as the

primary task and 3 other hyper-class augmentation classification as auxiliary tasks.

This direction of research can be explored further especially for hyper-classes with more complex and rich semantic information.

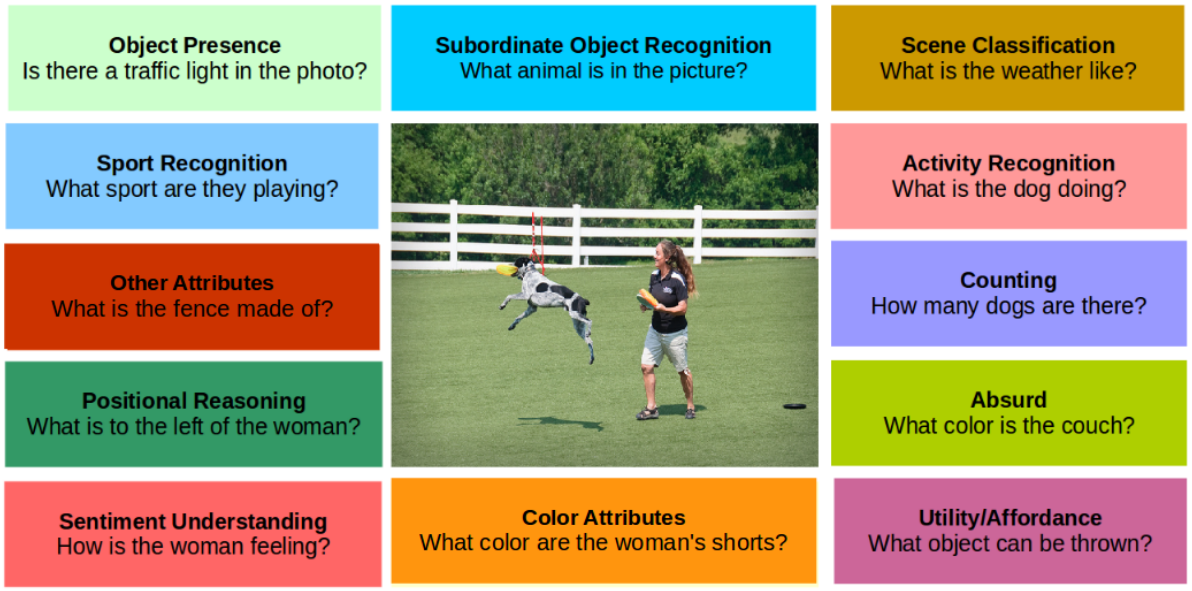


Figure 24. TDIUC task oriented dataset (Kafle and Kanan, 2017)

7.2 Employing COCO Action (COCO-a) dataset

(Ronchi and Perona, 2015) annotated an action subset of COCO image dataset with detectable interactions involving human agents as subjects. Figure 25 summarizes the process:

(1) Visual VerbNet: obtaining common visual actions by analysis of VerbNet and MS-COCO captions, **(2) Image and Subject Selection**: three types of scenes such as sport, outdoor and indoor are selected, **(3) Interactions Annotation**: each subject is annotated with the objects that the subject is interacting with and **(4) Visual Actions Annotation**: For each subject-object pair, it is labeled with possible actions and interactions involving that pair.

I was interested in exploring and enhancing the VQA from a linguistic point of view. I focused on lexical resources such as FrameNet. However, the COCO-a dataset provides interesting visual information as well. Not only does the dataset include more detailed annotations of interaction between entities and objects present in the image, but visual annotations of them in the image as well. Now the output not only can be classified based on the hyper-classes but also on detecting the object and entities as well. This multi-modal output augmentation sounds like a promising research direction to be explored.

Employing attention mechanisms in the proposed VQA model is an open research to be explored. For the imSitu dataset, it does not seem to be critical as the participants in the activity occupy a very large portion of the image. However, for coco-a images, it might be necessary to use attention because there are multiple agents and multiple interactions; Consequently, there might be cases where participants can be detected in a very small sub-image.

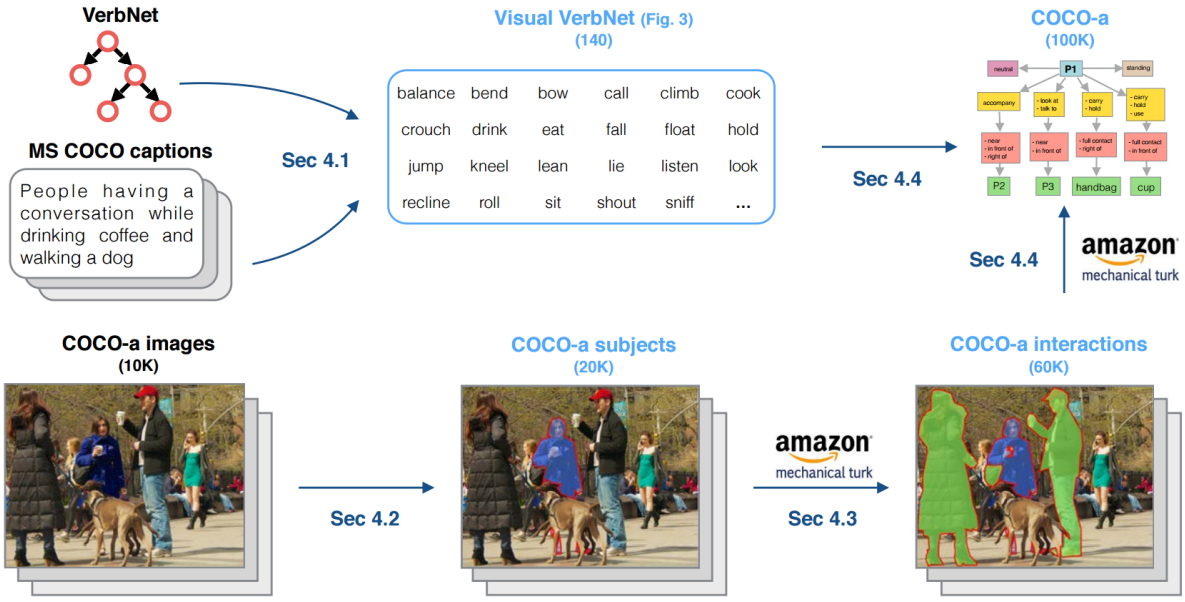


Figure 25. COCOA dataset sample annotation process (Ronchi and Perona, 2015)

APPENDICES

Appendix A

FINE-GRAINED PERFORMANCE EVALUATION ON IMSITUVQA

Performance evaluations by role

format **role** : (CNN-LSTM accuracy , multi-task CNN-LSTM accuracy)

sliceditem : (40.0 , 33.33) & **tool** : (42.84 , 51.78) & **object** : (30.69 , 39.66) & **crownedentity** : (44.0 , 52.0) & **flipped** : (48.33 , 68.33) & **placeditem** : (16.67 , 45.83) & **follower** : (33.56 , 39.6) & **part** : (7.84 , 27.45) & **agents** : (57.0 , 62.67) & **bodypart** : (38.86 , 57.77) & **bottom** : (70.0 , 60.0) & **agenttype** : (44.9 , 54.42) & **supported** : (38.81 , 29.85) & **crown** : (54.17 , 56.25) & **fluid** : (45.0 , 52.5) & **strap** : (57.14 , 71.43) & **pinned** : (34.38 , 28.13) & **pricked** : (55.41 , 62.16) & **addressee** : (34.35 , 40.46) & **brush** : (0.0 , 0.0) & **action** : (63.16 , 63.16) & **createditem** : (0.0 , 0.0) & **yankedpart** : (42.86 , 68.57) & **created** : (22.22 , 27.78) & **stake** : (73.29 , 78.08) & **feature** : (50.0 , 50.0) & **entityhelped** : (33.33 , 33.33) & **caughtitem** : (36.36 , 45.45) & **perceiver** : (28.85 , 26.92) & **victim** : (46.58 , 52.1) & **releaseditem** : (30.5 , 51.06) & **cause** : (54.65 , 58.14) & **yanked** : (29.27 , 46.34) & **turneditem** : (22.07 , 46.9) & **erased** : (27.71 , 48.19) & **sealant** : (67.74 , 70.97) & **objectpart** : (12.15 , 24.3) & **recipient** : (33.01 , 33.01) & **interviewee** : (41.61 , 44.97) & **drencheditem** :

Appendix A (Continued)

(0.0 , 0.0) & **item** : (32.27 , 39.65) & **fastener** : (50.0 , 60.87) & **paweditem** : (38.46 , 46.15) & **foodcontainer** : (21.05 , 42.11) & **experiencer** : (25.0 , 22.92) & **buyer** : (0.0 , 14.29) & **served** : (7.14 , 14.29) & **payment** : (0.0 , 100.0) & **unplugged** : (45.95 , 41.89) & **scrapeditem** : (27.27 , 30.3) & **clungto** : (30.41 , 39.86) & **boaters** : (50.0 , 100.0) & **connector** : (33.33 , 61.4) & **agentpart** : (0.0 , 0.0) & **individuals** : (61.33 , 63.33) & **container** : (34.17 , 45.47) & **glue** : (75.0 , 75.0) & **material** : (28.07 , 50.88) & **decomposer** : (15.5 , 32.56) & **tickled** : (20.83 , 25.0) & **performer** : (100.0 , 100.0) & **center** : (24.07 , 24.07) & **moisturized** : (43.94 , 50.0) & **seller** : (52.21 , 59.29) & **confronted** : (46.26 , 42.86) & **wrappingitem** : (43.64 , 40.0) & **stage** : (57.14 , 100.0) & **medium** : (71.43 , 67.53) & **end** : (25.85 , 34.01) & **food** : (32.4 , 30.67) & **items** : (26.67 , 38.0) & **victimpart** : (28.24 , 38.82) & **theme** : (20.0 , 36.0) & **place** : (34.75 , 39.52) & **vehicle** : (57.05 , 63.87) & **destroyeditem** : (0.0 , 14.29) & **brancher** : (41.33 , 46.0) & **blow** : (25.96 , 30.77) & **reference** : (24.18 , 24.18) & **gatherers** : (82.0 , 86.67) & **eater** : (57.14 , 28.57) & **component** : (71.43 , 71.43) & **surface** : (46.2 , 54.1) & **event** : (57.59 , 64.56) & **audience** : (64.34 , 64.34) & **carrier** : (40.0 , 48.24) & **scaffold** : (17.24 , 37.93) & **reciever** : (29.93 , 22.45) & **removeditem** : (25.0 , 0.0) & **substance** : (51.48 , 52.96) & **rammingitem** : (0.0 , 50.0) & **crop** : (33.33 , 33.33) & **beneficiary** : (58.33 , 33.33) & **obstacle** : (44.44 , 60.42) & **suspect** : (56.08 , 53.38) & **traveler** : (50.0 , 33.33) & **model** : (32.89 , 29.53) & **sprouter** : (71.33 , 76.0) & **coagentpart** : (25.0 , 33.33) & **liquid** : (71.11 , 69.63) & **source** : (29.65 , 40.12) & **undergoer** : (20.0 , 60.0) & **blocker** : (

Appendix A (Continued)

100.0 , 66.67) & **agent** : (48.78 , 52.29) & **shape** : (23.81 , 26.19) & **goalitem** : (47.09 , 55.03) & **signeditem** : (20.0 , 34.29) & **mass** : (42.5 , 37.5) & **pitched** : (100.0 , 100.0) & **quantity** : (75.0 , 75.0) & **heaveditem** : (60.22 , 68.82) & **goods** : (19.29 , 24.29) & **phenomenon** : (32.37 , 40.29) & **lock** : (14.29 , 42.86) & **cover** : (53.33 , 61.33) & **occasion** : (42.42 , 49.49) & **slider** : (0.0 , 0.0) & **mourner** : (42.28 , 43.62) & **smashed** : (41.67 , 25.0) & **restrained** : (45.64 , 51.68) & **itemtype** : (34.97 , 39.86) & **strapped** : (20.0 , 60.0) & **goal** : (34.0 , 30.0) & **shelter** : (84.87 , 93.28) & **chasee** : (36.0 , 33.33) & **depicted** : (16.05 , 9.88) & **contact** : (37.12 , 50.12) & **imitation** : (72.22 , 74.07) & **student** : (39.02 , 40.39) & **projectile** : (61.54 , 73.08) & **decorated** : (37.78 , 24.44) & **blocked** : (52.17 , 65.22) & **hidingitem** : (43.61 , 42.86) & **firearm** : (32.41 , 29.63) & **harvesteditem** : (0.0 , 0.0) & **rocked** : (10.0 , 10.0) & **top** : (66.0 , 66.0) & **destination** : (39.61 , 44.21) & **adressee** : (34.0 , 29.6) & **naggedperson** : (59.59 , 62.33) & **listener** : (35.21 , 41.2) & **weapon** : (36.84 , 42.11) & **baptized** : (31.51 , 32.88) & **aspect** : (33.33 , 0.0) & **path** : (0.0 , 0.0) & **hunted** : (37.84 , 30.63) & **giver** : (0.0 , 0.0) & **crusheditem** : (26.19 , 42.86) & **resource** : (40.0 , 80.0) & **deflecteditem** : (47.17 , 47.17) & **instrument** : (34.86 , 39.76) & **image** : (0.0 , 0.0) & **competition** : (44.9 , 51.02) & **components** : (50.0 , 37.5) & **target** : (41.67 , 46.17) & **dye** : (81.82 , 81.82) & **heatsource** : (12.5 , 25.0) & **reassured** : (22.82 , 24.16) & **good** : (9.09 , 18.18) & **admired** : (15.38 , 26.57) & **distributed** : (0.0 , 0.0) & **coagent** : (39.71 , 46.04) & **receiver** : (17.39 , 21.74) & **plunged** : (0.0 , 20.0) & **against** : (17.73 , 31.03) & **start** : (21.57 , 35.29) & **planted** : (33.33 ,

Appendix A (Continued)

0.0) & **message** : (100.0 , 100.0) & **ground** : (81.97 , 82.62) & **from** : (0.0 , 0.0) & **ailment** : (38.02 , 42.15) & **boringthing** : (35.48 , 44.35) & **focus** : (30.56 , 30.56) & **teacher** : (40.0 , 40.0) & **parachute** : (89.36 , 82.98) & **wrappeditem** : (25.0 , 19.23) & **cloth** : (30.14 , 58.9) & **skill** : (48.84 , 55.81) & **subject** : (100.0 , 100.0)

Performance evaluations by verb

Format **verb** : (CNN-LSTM accuracy , multi-task CNN-LSTM accuracy)

tattooing : (29.49 , 44.87) & **restraining** : (37.67 , 39.67) & **splashing** : (55.0 , 50.0) & **walking** : (40.64 , 39.58) & **sketching** : (26.92 , 61.54) & **skiing** : (43.77 , 46.8) & **inflating** : (31.17 , 48.05) & **rehabilitating** : (21.48 , 31.21) & **displaying** : (23.41 , 30.73) & **marching** : (47.33 , 49.67) & **feeding** : (27.78 , 33.33) & **hunting** : (35.0 , 34.0) & **shredding** : (27.78 , 30.56) & **chewing** : (38.51 , 39.86) & **teaching** : (51.34 , 56.71) & **flicking** : (48.15 , 50.0) & **shaving** : (58.33 , 87.5) & **ballooning** : (74.83 , 81.47) & **intermingling** : (46.42 , 51.88) & **flinging** : (28.57 , 47.62) & **sitting** : (36.45 , 41.14) & **yanking** : (42.86 , 56.25) & **putting** : (26.53 , 27.55) & **decorating** : (26.85 , 33.33) & **wilting** : (44.84 , 47.33) & **mashing** : (21.05 , 31.58) & **washing** : (50.0 , 62.5) & **kneading** : (44.67 , 54.67) & **rearing** : (49.82 , 58.36) & **urinating** : (28.76 , 32.11) & **slouching** : (41.08 , 53.54) & **moistening** : (38.64 , 18.18) & **inserting** : (38.1 , 40.48) & **lecturing** : (60.27 , 59.26) & **bubbling** : (28.23 , 33.47) & **installing** : (57.14 , 67.86) & **chopping** : (60.42 , 56.25) & **shrugging** : (

Appendix A (Continued)

66.06 , 62.9) & **parachuting** : (85.61 , 87.88) & **giving** : (31.06 , 24.24) & **cooking** : (30.12 , 44.58) & **waiting** : (35.27 , 44.18) & **saluting** : (37.98 , 43.41) & **falling** : (45.45 , 43.18) & **racing** : (47.65 , 49.66) & **ramming** : (25.0 , 45.0) & **pushing** : (10.0 , 20.0) & **wrapping** : (37.4 , 34.96) & **leaping** : (18.92 , 40.54) & **packing** : (41.0 , 48.0) & **foraging** : (35.91 , 44.3) & **operating** : (41.67 , 47.22) & **ascending** : (43.84 , 45.89) & **stretching** : (32.0 , 39.67) & **ignoring** : (43.21 , 51.57) & **whisking** : (47.83 , 63.77) & **sharpening** : (28.13 , 45.83) & **jogging** : (41.28 , 50.0) & **browsing** : (41.81 , 54.18) & **sealing** : (51.16 , 50.0) & **rafting** : (64.33 , 73.33) & **examining** : (29.17 , 38.89) & **hiking** : (54.0 , 55.0) & **clapping** : (37.33 , 38.67) & **erupting** : (67.59 , 68.97) & **hunching** : (24.16 , 30.87) & **raking** : (25.0 , 25.0) & **ailing** : (33.67 , 42.35) & **singing** : (42.11 , 43.72) & **standing** : (56.54 , 51.92) & **taxiing** : (87.63 , 87.29) & **working** : (31.1 , 33.44) & **whipping** : (16.0 , 54.0) & **crowning** : (49.54 , 52.29) & **steering** : (49.12 , 49.12) & **distributing** : (0.0 , 0.0) & **spying** : (39.85 , 45.11) & **uncorking** : (35.11 , 51.91) & **panhandling** : (43.33 , 46.0) & **weeping** : (41.8 , 44.67) & **squeezing** : (58.13 , 53.75) & **curtsying** : (30.5 , 45.39) & **pawing** : (46.15 , 48.72) & **spearing** : (36.11 , 38.89) & **whirling** : (51.9 , 57.09) & **dripping** : (32.56 , 40.7) & **carrying** : (22.03 , 23.73) & **pinching** : (23.77 , 30.49) & **flapping** : (47.83 , 51.84) & **tying** : (32.81 , 34.38) & **arranging** : (45.69 , 50.86) & **tearing** : (56.14 , 71.93) & **blocking** : (53.7 , 51.85) & **swooping** : (36.49 , 38.6) & **practicing** : (44.68 , 45.74) & **recovering** : (33.45 , 48.65) & **shushing** : (45.91 , 45.14) & **dousing** : (23.08 , 53.85) & **buttering** : (24.07 , 40.74) & **hoeing** : (70.67 , 69.0)

Appendix A (Continued)

& **making** : (20.59 , 52.94) & **unpacking** : (33.33 , 43.33) & **mowing** : (50.0 , 37.5) & **stacking** : (67.33 , 62.38) & **stinging** : (51.06 , 63.83) & **heaving** : (52.69 , 58.6) & **pulling** : (14.29 , 21.43) & **counting** : (27.61 , 33.67) & **breaking** : (27.19 , 29.49) & **docking** : (30.56 , 36.11) & **scolding** : (41.16 , 47.28) & **complaining** : (29.96 , 37.91) & **drying** : (38.37 , 50.0) & **poking** : (9.09 , 31.82) & **tickling** : (18.37 , 22.45) & **bandaging** : (30.0 , 33.85) & **recuperating** : (37.24 , 40.34) & **butting** : (43.0 , 46.67) & **communicating** : (32.33 , 31.33) & **begging** : (31.74 , 38.7) & **jumping** : (16.13 , 35.48) & **hoisting** : (0.0 , 8.33) & **pasting** : (34.29 , 51.43) & **lighting** : (36.22 , 43.31) & **shelving** : (62.96 , 57.41) & **wagging** : (33.33 , 40.23) & **lifting** : (27.27 , 40.91) & **surfing** : (0.0 , 0.0) & **gluing** : (30.36 , 57.14) & **twisting** : (46.0 , 48.0) & **hauling** : (29.17 , 25.0) & **destroying** : (42.31 , 50.0) & **crawling** : (30.77 , 28.32) & **punting** : (65.55 , 68.9) & **erasing** : (44.83 , 65.52) & **bathing** : (27.66 , 46.81) & **serving** : (20.75 , 34.91) & **telephoning** : (41.52 , 50.87) & **miming** : (48.93 , 57.14) & **congregating** : (41.28 , 43.62) & **smiling** : (46.0 , 46.8) & **drinking** : (26.35 , 35.14) & **scoring** : (56.42 , 62.16) & **pouting** : (38.98 , 48.43) & **yawning** : (45.65 , 45.29) & **soaking** : (25.0 , 25.0) & **glowing** : (20.31 , 16.8) & **writing** : (55.56 , 72.22) & **coaching** : (21.33 , 30.67) & **brawling** : (31.94 , 31.6) & **spoiling** : (41.06 , 48.67) & **pooing** : (43.1 , 51.85) & **sucking** : (38.59 , 41.95) & **unveiling** : (30.42 , 34.97) & **flipping** : (39.45 , 49.54) & **peeing** : (39.13 , 39.8) & **educating** : (47.78 , 52.22) & **bothering** : (51.39 , 50.0) & **sweeping** : (18.75 , 25.0) & **baking** : (33.08 , 40.0) & **spinning** : (37.25 , 25.49) & **dyeing** : (52.69 , 56.99) & **slipping** : (34.67 , 45.0)

Appendix A (Continued)

& **ducking** : (29.19 , 35.23) & **shaking** : (57.14 , 76.98) & **signing** : (51.19 , 59.52) & **braiding** : (51.0 , 63.0) & **stooping** : (30.93 , 31.62) & **celebrating** : (37.07 , 43.88) & **catching** : (48.72 , 56.41) & **juggling** : (29.0 , 38.33) & **preaching** : (48.15 , 56.9) & **scratching** : (50.0 , 57.74) & **shearing** : (75.0 , 91.67) & **extinguishing** : (40.63 , 54.69) & **driving** : (54.33 , 58.67) & **puckering** : (44.0 , 48.0) & **peeling** : (37.68 , 40.58) & **hanging** : (25.76 , 28.79) & **bulldozing** : (50.0 , 54.8) & **sliding** : (28.0 , 24.0) & **igniting** : (37.24 , 40.31) & **turning** : (31.99 , 41.08) & **clenching** : (45.38 , 60.5) & **mourning** : (30.04 , 36.4) & **grimacing** : (45.66 , 47.92) & **fastening** : (25.0 , 15.0) & **plowing** : (54.55 , 59.09) & **repairing** : (19.23 , 42.31) & **training** : (40.4 , 42.42) & **potting** : (51.09 , 54.35) & **blossoming** : (48.76 , 56.18) & **cheering** : (31.52 , 35.14) & **nailing** : (10.0 , 20.0) & **wading** : (43.67 , 43.33) & **leading** : (30.2 , 37.58) & **biting** : (36.91 , 40.94) & **strapping** : (27.27 , 40.91) & **sneezing** : (54.93 , 63.38) & **drooling** : (35.57 , 43.29) & **striking** : (37.5 , 50.0) & **knocking** : (41.67 , 53.0) & **encouraging** : (30.54 , 24.5) & **soaring** : (62.08 , 64.09) & **flossing** : (57.21 , 64.41) & **manicuring** : (52.94 , 62.75) & **squinting** : (41.94 , 48.39) & **boating** : (47.16 , 39.36) & **typing** : (43.62 , 50.67) & **lacing** : (30.2 , 28.19) & **farming** : (58.33 , 41.67) & **photographing** : (45.54 , 58.42) & **speaking** : (39.39 , 45.12) & **mining** : (31.25 , 45.0) & **selling** : (17.65 , 21.01) & **winking** : (47.2 , 44.0) & **bouncing** : (27.59 , 29.31) & **leaking** : (33.63 , 43.36) & **cheerleading** : (39.04 , 49.32) & **autographing** : (18.3 , 28.1) & **chasing** : (35.33 , 31.0) & **boarding** : (41.47 , 55.52) & **nagging** : (54.88 , 57.58) & **scraping** : (29.07 , 37.21) & **recording** : (32.53 , 43.84) & **providing**

Appendix A (Continued)

: (18.18 , 22.73) & **tipping** : (42.37 , 37.29) & **nuzzling** : (46.0 , 47.33) & **gathering** : (50.0 , 52.36) & **reassuring** : (29.1 , 33.44) & **patting** : (53.73 , 73.13) & **talking** : (37.25 , 41.61) & **interrogating** : (23.08 , 30.77) & **emptying** : (41.67 , 58.33) & **slithering** : (57.09 , 62.63) & **skidding** : (38.0 , 47.33) & **flaming** : (29.37 , 28.57) & **dining** : (43.48 , 52.51) & **hurling** : (33.33 , 52.78) & **scooping** : (16.67 , 50.0) & **weeding** : (34.33 , 44.33) & **disembarking** : (32.78 , 39.8) & **sewing** : (33.33 , 53.7) & **videotaping** : (26.99 , 31.83) & **colliding** : (42.0 , 43.67) & **socializing** : (43.14 , 48.16) & **subduing** : (24.75 , 26.42) & **giggling** : (24.18 , 32.23) & **sprinting** : (51.35 , 51.35) & **drawing** : (25.71 , 34.76) & **voting** : (37.25 , 39.26) & **swarming** : (35.21 , 41.2) & **attacking** : (27.03 , 28.38) & **signaling** : (51.14 , 60.23) & **dissecting** : (20.71 , 22.14) & **emerging** : (31.11 , 31.11) & **applauding** : (21.27 , 27.61) & **grilling** : (22.26 , 27.55) & **spanking** : (56.82 , 40.91) & **frisking** : (36.0 , 39.0) & **gasping** : (40.8 , 39.2) & **crashing** : (31.86 , 39.82) & **coloring** : (28.26 , 39.13) & **detaining** : (40.0 , 53.67) & **riding** : (43.81 , 46.49) & **clawing** : (41.92 , 50.86) & **scrubbing** : (35.37 , 36.59) & **rinsing** : (51.92 , 59.62) & **fording** : (44.18 , 44.86) & **fueling** : (24.07 , 50.0) & **grieving** : (27.41 , 34.75) & **sprouting** : (50.0 , 54.17) & **tilting** : (45.45 , 62.12) & **frowning** : (48.35 , 47.52) & **spraying** : (5.56 , 11.11) & **competing** : (40.0 , 39.67) & **frying** : (29.59 , 36.73) & **imitating** : (29.0 , 31.0) & **stitching** : (41.88 , 55.63) & **sowing** : (52.01 , 65.1) & **spilling** : (0.0 , 21.43) & **covering** : (35.03 , 49.68) & **rowing** : (41.33 , 40.67) & **dipping** : (35.94 , 32.81) & **licking** : (23.91 , 29.63) & **calling** : (46.82 , 53.18) & **shelling** : (29.1 , 42.14) & **overflowing** : (29.35 ,

Appendix A (Continued)

34.81) & **wheeling** : (20.37 , 24.07) & **discussing** : (37.41 , 40.48) & **circling** : (55.48 , 53.0) & **constructing** : (31.25 , 40.63) & **burying** : (22.22 , 25.93) & **throwing** : (27.56 , 34.65) & **drenching** : (50.0 , 60.0) & **injecting** : (43.33 , 50.0) & **pricking** : (53.29 , 55.26) & **exterminating** : (21.67 , 24.33) & **camping** : (59.44 , 67.87) & **tackling** : (68.33 , 62.33) & **clearing** : (50.0 , 50.0) & **interviewing** : (37.67 , 42.67) & **plummeting** : (54.0 , 68.0) & **wrinkling** : (40.67 , 41.33) & **wetting** : (58.82 , 58.82) & **reading** : (30.41 , 44.59) & **weighing** : (51.06 , 60.64) & **deflecting** : (44.63 , 47.93) & **opening** : (53.57 , 52.14) & **brewing** : (32.26 , 43.37) & **lapping** : (57.58 , 54.55) & **vacuuming** : (40.43 , 57.45) & **kicking** : (31.61 , 34.48) & **phoning** : (54.7 , 61.74) & **picking** : (40.0 , 28.57) & **staring** : (31.51 , 32.53) & **folding** : (30.46 , 45.03) & **wringing** : (20.74 , 26.76) & **smearing** : (33.33 , 33.33) & **betting** : (33.45 , 37.54) & **smashing** : (25.53 , 21.28) & **packaging** : (25.84 , 33.56) & **nipping** : (24.0 , 24.0) & **stroking** : (24.07 , 35.19) & **pouring** : (12.5 , 12.5) & **pedaling** : (61.67 , 61.67) & **tuning** : (30.28 , 41.28) & **dusting** : (20.75 , 18.87) & **chiseling** : (33.45 , 37.16) & **attaching** : (25.0 , 35.42) & **dropping** : (21.05 , 31.58) & **baptizing** : (26.01 , 30.41) & **instructing** : (26.67 , 36.0) & **resting** : (28.57 , 33.93) & **shopping** : (27.85 , 33.89) & **combing** : (66.49 , 73.94) & **coughing** : (59.67 , 67.9) & **releasing** : (32.55 , 43.29) & **misbehaving** : (26.14 , 31.44) & **cresting** : (50.51 , 48.81) & **rocking** : (30.26 , 34.36) & **dialing** : (49.16 , 67.0) & **paying** : (44.57 , 50.86) & **guarding** : (29.11 , 30.38) & **queuing** : (46.55 , 48.28) & **grinning** : (34.48 , 35.63) & **leaning** : (18.85 , 31.56) & **slapping** : (37.04 , 38.89) & **trimming** : (0.0 , 0.0) & **biking** : (32.54 , 36.61

Appendix A (Continued)

) & **piloting** : (48.98 , 59.86) & **welding** : (34.62 , 26.92) & **stuffing** : (29.76 , 48.81) & **tilling** : (48.48 , 50.51) & **arching** : (25.67 , 35.33) & **rubbing** : (15.38 , 16.67) & **sleeping** : (44.14 , 52.76) & **skipping** : (42.81 , 51.17) & **pressing** : (32.09 , 36.82) & **clinging** : (35.91 , 40.94) & **swinging** : (37.58 , 45.3) & **pruning** : (10.0 , 10.0) & **pinning** : (29.35 , 29.35) & **dragging** : (16.67 , 33.33) & **barbecuing** : (34.34 , 29.97) & **parading** : (61.07 , 59.06) & **fetching** : (32.43 , 37.84) & **locking** : (50.36 , 54.74) & **shoveling** : (42.86 , 35.71) & **glaring** : (39.53 , 44.96) & **confronting** : (40.0 , 39.33) & **cramming** : (36.67 , 53.33) & **disciplining** : (36.61 , 46.88) & **microwaving** : (56.55 , 57.93) & **running** : (46.67 , 51.0) & **cleaning** : (21.54 , 33.85) & **spitting** : (25.84 , 28.09) & **climbing** : (33.33 , 28.57) & **slicing** : (47.22 , 38.89) & **dancing** : (36.82 , 44.77) & **flexing** : (60.0 , 60.0) & **studying** : (33.22 , 35.23) & **massaging** : (48.08 , 53.85) & **handcuffing** : (49.32 , 56.42) & **assembling** : (27.4 , 32.88) & **lathering** : (48.68 , 68.42) & **shivering** : (44.32 , 48.48) & **mending** : (24.11 , 34.82) & **distracting** : (58.0 , 64.33) & **fishing** : (6.25 , 18.75) & **praying** : (37.93 , 43.3) & **kneeling** : (46.42 , 52.22) & **loading** : (40.0 , 60.0) & **decomposing** : (20.82 , 34.29) & **exercising** : (53.25 , 69.48) & **arresting** : (46.28 , 46.28) & **shouting** : (41.48 , 55.19) & **crying** : (41.87 , 43.9) & **unloading** : (25.0 , 50.0) & **harvesting** : (40.0 , 20.0) & **unplugging** : (38.05 , 39.39) & **molding** : (33.8 , 35.92) & **checking** : (35.48 , 35.48) & **descending** : (25.0 , 30.56) & **saying** : (29.76 , 28.72) & **aiming** : (37.25 , 42.11) & **pouncing** : (26.51 , 31.21) & **waddling** : (41.41 , 43.1) & **performing** : (49.12 , 63.16) & **taping** : (20.0 , 27.86) & **laughing** : (39.85 , 34.32) & **immersing** : (50.0 ,

Appendix A (Continued)

53.85) & **gardening** : (28.72 , 35.14) & **punching** : (31.65 , 48.2) & **twirling** : (24.75 , 35.45) & **diving** : (49.66 , 54.08) & **sniffing** : (38.93 , 40.6) & **filling** : (27.78 , 27.78) & **grinding** : (21.74 , 29.35) & **filming** : (26.53 , 41.84) & **asking** : (38.19 , 40.97) & **measuring** : (25.35 , 42.25) & **eating** : (48.48 , 49.49) & **buckling** : (71.05 , 60.53) & **hitchhiking** : (44.82 , 61.2) & **bowing** : (28.11 , 29.18) & **tripping** : (25.0 , 28.0) & **officiating** : (47.65 , 57.05) & **submerging** : (75.68 , 78.38) & **waving** : (24.0 , 34.0) & **crafting** : (28.68 , 32.35) & **drumming** : (71.15 , 76.92) & **plunging** : (35.92 , 28.16) & **swimming** : (36.36 , 34.34) & **carting** : (36.0 , 41.0) & **watering** : (18.75 , 31.25) & **stampeding** : (0.0 , 0.0) & **curling** : (57.14 , 75.32) & **burning** : (35.0 , 37.31) & **perspiring** : (48.18 , 47.77) & **clipping** : (28.57 , 28.57) & **snuggling** : (38.33 , 43.67) & **carving** : (28.17 , 33.1) & **tugging** : (26.1 , 37.97) & **calming** : (37.76 , 41.5) & **hitting** : (39.13 , 19.57) & **apprehending** : (36.05 , 46.94) & **retrieving** : (37.1 , 35.48) & **camouflaging** : (45.39 , 51.06) & **placing** : (24.0 , 33.33) & **towing** : (34.67 , 34.67) & **prowling** : (48.48 , 52.53) & **launching** : (50.0 , 62.5) & **crushing** : (29.79 , 36.17) & **pitching** : (100.0 , 100.0) & **offering** : (43.53 , 42.35) & **branching** : (44.44 , 47.22) & **adjusting** : (52.54 , 67.8) & **landing** : (56.67 , 59.33) & **moisturizing** : (43.57 , 45.71) & **tasting** : (28.93 , 26.45) & **wiping** : (30.26 , 25.0) & **sprinkling** : (13.64 , 18.18) & **buying** : (27.42 , 64.52) & **applying** : (32.14 , 39.29) & **protesting** : (43.67 , 45.33) & **stumbling** : (37.04 , 38.72) & **stapling** : (37.5 , 52.5) & **mopping** : (52.01 , 59.06) & **stirring** : (37.5 , 25.0) & **skating** : (39.33 , 51.0) & **gnawing** : (38.26 , 35.57) & **floating** : (51.9 , 50.0) & **shooting** : (46.33 , 50.0) & **commuting** : (

Appendix A (Continued)

54.67 , 66.0) & **planting** : (18.18 , 18.18) & **fixing** : (33.33 , 42.86) & **milking** : (0.0 , 0.0) & **gambling** : (59.93 , 80.48) & **brushing** : (31.94 , 43.06) & **vaulting** : (0.0 , 28.57) & **crouching** : (35.92 , 40.14) & **embracing** : (43.69 , 40.96) & **whistling** : (26.8 , 46.39) & **helping** : (31.25 , 50.0) & **smelling** : (35.69 , 38.38) & **unlocking** : (18.75 , 31.25) & **ejecting** : (33.33 , 16.67) & **building** : (45.0 , 50.0) & **rotting** : (31.06 , 26.89) & **painting** : (10.94 , 29.69) & **admiring** : (21.33 , 29.0) & **waxing** : (28.77 , 52.05) & **stripping** : (26.09 , 21.74) & **prying** : (42.31 , 42.31) & **buttoning** : (40.27 , 42.95)

Appendix B

IMSITU VERB ABSTRACT DEFINITIONS

This appendix includes segmented abstract definitions sorted by verbs.

A.

[an AGENT] adjusts [an ITEM's FEATURE] [using TOOL] [at PLACE]

[an AGENT] is admiring [the ADMIRED] [at PLACE]

[the VICTIM's VICTIMPART] is ailing [with the CAUSE] [at the PLACE]

[an AGENT] aims [an ITEM] [at TARGET] [in PLACE]

[an AGENT] applauds [an ADDRESSEE] [at PLACE]

[AGENT] is applying [SUBSTANCE] [to DESTINATION] [using TOOL] [in PLACE]

[AGENT] apprehended [VICTIM] [in PLACE]

[AGENT] arches [BODYPART] [at PLACE]

[the AGENT] arranges [ITEM] [with TOOL] [in PLACE]

[the AGENT] arrested [the SUSPECT] [in PLACE]

[an AGENT] ascends [at PLACE]

[an AGENT] asks [an ADDRESSEE] [at PLACE]

[an AGENT] assembles [the GOALITEM] [with COMPONENTs using TOOL] [at PLACE]

[an AGENT] attaches [ITEM] [to DESTINATION] [with GLUE] [using TOOL] [at PLACE]

[an AGENT] attacks [VICTIM] [using WEAPON] [at PLACE]

[an AGENT] autographs [an ITEM] [for RECEIVER] [at PLACE]

Appendix B (Continued)

B.

[an AGENT] bakes [FOOD] [in FOODCONTAINER] [by applying heat with HEATSOURCE] [at PLACE]

[an AGENT] balloons [at PLACE]

[AGENT] bandages [VICTIM] [at PLACE]

[AGENT] baptizes [BAPTIZED] [at PLACE]

[an AGENT] barbecues [FOOD] [at PLACE]

[an AGENT] bathes [COAGENT] [using TOOL] [and SUBSTANCE] [in PLACE]

[the AGENT] is begging [the GIVER] [for ITEM] [in PLACE]

[AGENT] bets [at PLACE]

[an AGENT] bikes [at PLACE]

[AGENT] is biting [ITEM] [in PLACE]

[the BLOCKER] blocked [the BLOCKED] [with TOOL] [in PLACE]

[an AGENT] blossoms [in PLACE]

[the AGENT] boards [VEHICLE] [at PLACE]

[the BOATERS] boat [on VEHICLE] [in PLACE]

[the AGENT] bothers [the VICTIM] [by do an ACTION] [in PLACE]

[the AGENT] bounces [an ITEM] [against SURFACE] [in PLACE]

[an AGENT] bows [at PLACE]

[AGENT] braids [ITEM] [at PLACE]

[BRANCHER] branches [at PLACE]

[an AGENT] brawls [at PLACE]

Appendix B (Continued)

[the AGENT] breaks [the ITEM] [using TOOL] [at PLACE]

[an AGENT] brews [TARGET] [at PLACE]

[an AGENT] browses [for GOALITEM] [at PLACE]

[an AGENT] brushes [TARGET] [with TOOL] [using SUBSTANCE] [at PLACE]

[the AGENT] bubbles [in PLACE]

[an AGENT] buckles [an ITEM] [using FASTENER] [into CONTAINER] [at PLACE]

[an AGENT] builds [GOALITEM] [from COMPONENTS] [using TOOL] [in PLACE]

[the AGENT] bulldozes [the OBJECT] [at PLACE]

[AGENT] is burning [TARGET] [in PLACE]

[AGENT] buries [an ITEM] [into DESTINATION] [using TOOL] [at PLACE]

[an AGENT] butters [an ITEM] [using TOOL] [in PLACE]

[AGENT] butts [TARGET] [at PLACE]

[an AGENT] buttons [an ITEM] [in PLACE]

[the AGENT] buys [GOODS] [with PAYMENT] [from the SELLER] [in PLACE]

C.

[an AGENT] calls [using TOOL] [at PLACE]

[AGENT] is calming [EXPERIENCER] [in PLACE]

[AGENT] is camouflaging [into HIDINGITEM] [in PLACE]

[an AGENT] camps [on/in SHELTER] [at PLACE]

[the AGENT] caresses [the RECIPIENTPART] [with the AGENTPART] [at PLACE]

[an AGENT] carries [an ITEM] [on their AGENTPART] [at PLACE]

Appendix B (Continued)

[the AGENT] cartes [the ITEM] [in TOOL] [at PLACE]

[AGENT] carved [SUBSTANCE] [with TOOL] [in PLACE]

[an AGENT] catches [CAUGHTITEM] [with TOOL] [at PLACE]

[an AGENT] celebrates [an OCCASION] [at PLACE]

[an AGENT] chases [the CHASEE] [at PLACE]

[the AGENT] checks [the PATIENT's ASPECT] [with the TOOL] [in the PLACE]

[the AGENT] cheers [in PLACE]

[an AGENT] cheerleads [for the SUPPORTED] [at PLACE]

[an AGENT] chews [an ITEM] [in PLACE]

[the AGENT] chisels [the ITEM] [at the PLACE]

[AGENT] circles [CENTER] [in PLACE]

[the AGENT] claps [their AGENTPART] [in PLACE]

[AGENT] is clawing [VICTIM] [in PLACE]

[AGENT] is cleaning [SOURCE] [with TOOL] [in PLACE]

[an AGENT] clears [an ITEM] [from SOURCE] [using TOOL] [in PLACE]

[an AGENT] clenched [an ITEM] [with the AGENTPART] [at PLACE]

[the AGENT] climbs [an OBSTACLE] [with TOOL] [at PLACE]

[AGENT] is clipping [ITEM] [from SOURCE] [with TOOL] [in PLACE]

[an AGENT] coaches [STUDENT] [to be good at SKILL] [at PLACE]

[the AGENT] collides [with the ITEM] [at PLACE]

[the AGENT] colors [ITEM] [with TOOL] [in PLACE]

Appendix B (Continued)

[the AGENT] combs [the TARGET] [with TOOL] [at PLACE]

[an AGENT] communicates [to the ADRESSEE] [at PLACE]

[the TRAVELER] commutes [in VEHICLE] [in PLACE]

[AGENT] is competing [in COMPETITION] [in PLACE]

[the AGENT] complains [to the LISTENER] [in PLACE]

[AGENT] confronts [CONFRONTED] [in PLACE]

[the INDIVIDUALS] congregate [at the PLACE]

[an AGENT] constructs [CREATEDITEM] [with TOOL] [from the COMPONENTS] [at PLACE]

[an AGENT] cooks [FOOD] [in CONTAINER] [over HEATSOURCE] [using TOOL] [in PLACE]

[the AGENT] coughs [in PLACE]

[an AGENT] counts [the ITEMTYPE] [at PLACE]

[the AGENT] covers [the ITEM] [with COVER] [at PLACE]

[an AGENT] crafts [CREATED] [with INSTRUMENT] [in PLACE]

[the AGENT] crammed [the THEME] [into the CONTAINER] [in the PLACE]

[the AGENT] crashes [the ITEM] [into the AGAINST] [at PLACE]

[an AGENT] crawls [at PLACE]

[an AGENT] crests [at PLACE]

[the AGENT] crouches [in the PLACE]

[the AGENT] crowns [the CROWNEDENTITY] [with CROWN] [at the PLACE]

[an AGENT] crushed [the CRUSHEDITEM] [with TOOL] [at PLACE]

[an AGENT] cries [in PLACE]

Appendix B (Continued)

[AGENT] curling [TARGET] [with TOOL] [in PLACE]

[an AGENT] curtsies [at PLACE]

D.

[the AGENT] dances [in PLACE]

[DECOMPOSER] decomposes [at PLACE]

[an AGENT] decorates [the DECORATED] [with an ITEM] [at PLACE]

[the AGENT] deflected [the DEFLECTEDITEM] [to the DESTINATION] [at the PLACE]

[the AGENT] [uses TOOL] to descend [from SOURCE] [at PLACE]

[an AGENT] destroyed [the DESTROYEDITEM] [with TOOL] [at PLACE]

[an AGENT] detains [VICTIM] [at PLACE]

[an AGENT] dials [on an ITEM] [at PLACE]

[the AGENT] dined [on the FOOD] [in the PLACE]

[an AGENT] dips [an ITEM] [into SUBSTANCE] [at PLACE]

[an AGENT] disciplines [VICTIM] [with TOOL] [in PLACE]

[AGENTS] discuss [at PLACE]

[the AGENT] disembarks [from VEHICLE] [in PLACE]

[the AGENT] displays [the ITEM] [to the TARGET] [using TOOL] [at PLACE]

[AGENT] is dissecting [ITEM] [using TOOL] [in PLACE]

[an AGENT] distracts [VICTIM] [at PLACE]

[the AGENT] [uses TOOL] to distribute [DISTRIBUTED] [to RECIPIENTS] [at PLACE]

[the AGENT] docks [VEHICLE] [with CONNECTOR] [in PLACE]

Appendix B (Continued)

[AGENT] is dousing [UNDERGOER] [with LIQUID] [in PLACE]

[an AGENT] drags [an ITEM] [with TOOL] [on CONTACT] [at PLACE]

[an AGENT] draws [REFERENCE] [using TOOL] [in PLACE]

[AGENT] is drenching [DRENCHEDITEM] [with LIQUID] [using TOOL] [in PLACE]

[the AGENT] drinks [LIQUID] [from CONTAINER] [at PLACE]

[the AGENT] drips [the FLUID] [from the SOURCE] [to the DESTINATION] [in the PLACE]

[AGENT] is driving [ITEM] [in PLACE]

[the AGENT] drools [on END] [in PLACE]

[an AGENT] dropped [an ITEM] [from the START] [to the END] [at PLACE]

[the AGENT] drums [on the ITEM] [with the TOOL] [in the PLACE]

[AGENT] dries [ITEM] [using TOOL] [at PLACE]

[an AGENT] ducks [to avoid BLOW] [in PLACE]

[an AGENT] dusts [SOURCE] [using TOOL] [at PLACE]

[the AGENT] dyes [the MATERIAL] [with DYE] [in PLACE]

E.

[an AGENT] eat [FOOD] [from CONTAINER] [using TOOL] [at PLACE]

[TEACHER] educates [STUDENT] [on SUBJECT] [at PLACE]

[an AGENT] ejects [an ITEM] [from SOURCE] [toward DESTINATION] [at PLACE]

[the AGENT] embraces [the COAGENT] [at PLACE]

[the AGENT] emerges [from SOURCE] [to DESTINATION] [in PLACE]

[an AGENT] empties [an ITEM] [from CONTAINER] [into DESTINATION] [using TOOL] [in PLACE]

Appendix B (Continued)

[the AGENT] encourages [the RECIEVER] [in PLACE]

[the AGENT] erases [the ERASED] [from SOURCE] [in PLACE]

[an AGENT] erupts [with SUBSTANCE] [in PLACE]

[the AGENT] examines [the ITEM] [using TOOL] [at PLACE]

[the AGENT] exercises [the AGENTPART] [in PLACE]

[the AGENT] is exterminating [the PLACE] [with the INSTRUMENT]

[the AGENT] extinguishes [ITEM] [using TOOL] [in PLACE]

F.

[the AGENT] falls [from SOURCE] [to GOAL] [at PLACE]

[the FARMER] farms [the ITEM] [from the GROUND] [using TOOL] [at PLACE]

[an AGENT] fastens [an ITEM] [into DESTINATION] [with CONNECTOR] [using TOOL] [at PLACE]

[the AGENT] feeds [FOOD] [from SOURCE] [to the EATER] [in PLACE]

[the AGENT] fetches [the ITEM] [from SOURCE] [and brings it to DESTINATION,] [in PLACE]

[AGENT] fills [DESTINATION] [with ITEM] [from SOURCE] [at PLACE]

[the AGENT] films [PERFORMER] [using TOOL] [at PLACE]

[the AGENT] fish [from SOURCE] [using TOOL] [at the PLACE]

[the AGENT] fixes [the OBJECT's OBJECTPART] [with TOOL] [in PLACE]

[an AGENT] flames [at PLACE]

[the AGENT] flapped [its BODYPART] [in PLACE]

[an AGENT] flexes [their AGENTPART] [at an ADDRESSEE] [at PLACE]

[the AGENT] flicks [the OBJECT] [in the OBJECTPART] [with their AGENTPART] [at PLACE]

Appendix B (Continued)

[AGENT] flings [ITEM] [toward GOAL] [using TOOL] [in PLACE]

[the AGENT] flips [the FLIPPED] [with TOOL] [in PLACE]

[the AGENT] is floating [in MEDIUM] [using TOOL] [at PLACE]

[an AGENT] flosses [COAGENT] [in PLACE]

[an AGENT] folds [CLOTH] [into SHAPE] [at PLACE]

[the AGENT] forages [for ITEM] [in PLACE]

[the AGENT] fords [using TOOL] [in PLACE]

[an AGENT] frisks [VICTIM] [at PLACE]

[the AGENT] frowns [in PLACE]

[an AGENT] fries [FOOD] [in CONTAINER] [at PLACE]

[AGENT] fuels [RECIPIENT] [with TOOL] [in PLACE]

G.

[the AGENT] gambles [STAKE] [in PLACE]

[the AGENT] gardens [with the help of TOOL] [in PLACE]

[the AGENT] is gasping [in PLACE]

[the GATHERERS] gathered [in PLACE]

[the AGENT] giggled [at the TARGET] [at PLACE]

[an AGENT] gives [an ITEM] [to the RECIPIENT] [at PLACE]

[an AGENT] glares [at PERCEIVER] [in PLACE]

[the AGENT] glows [at PLACE]

[the AGENT] glues [ITEM] [to GOAL] [with CONNECTOR] [in PLACE]

Appendix B (Continued)

[an AGENT] gnaws [an ITEM] [at PLACE]

[an AGENT] grieves [at PLACE]

[an AGENT] grills [FOOD] [at PLACE]

[the AGENT] grimaced [because of the CAUSE] [at the PLACE]

[the AGENT] grinds [an ITEM] [with TOOL] [against SURFACE] [in PLACE]

[the AGENT] grinned [in the PLACE]

[an AGENT] guards [an ITEM] [with WEAPON] [at PLACE]

H.

[AGENT] handcuffs [VICTIM] [at PLACE]

[the AGENT] hangs [the ITEM] [from the SCAFFOLD] [in the PLACE]

[the AGENT] harvest [the HARVESTEDITEM] [using TOOL] [from the GROUND] [at the PLACE]

[CARRIER] hauls [an ITEM] [with TOOL] [in PLACE]

[the AGENT] heaves [the HEAVEDITEM] [using TOOL] [at PLACE]

[AGENT] helped [ENTITYHELPED] [with TOOL] [in PLACE]

[the AGENT] hikes [at PLACE]

[an AGENT] hitchhikes [at PLACE]

[the AGENT] hits [the VICTIM] [on the VICTIMPART] [with TOOL] [in PLACE]

[an AGENT] hoes [the GROUND] [in PLACE]

[an AGENT] hoists [an ITEM] [up from SOURCE] [using TOOL] [at PLACE]

[the AGENT] to hug [HUGGED] [at PLACE]

[an AGENT] hunches [over SURFACE] [in PLACE]

Appendix B (Continued)

[the AGENT] hunts [the HUNTED] [in the PLACE]

[the AGENT] hurls [the OBJECT] [from START] [to END] [at PLACE]

I.

[an AGENT] ignites [the ITEM] [with TOOL] [at PLACE]

[the AGENT] ignores [the BORINGTHING] [in the PLACE]

[AGENT] imitates [MODEL] [in PLACE]

[the AGENT] immerses [an ITEM] [in LIQUID] [in PLACE]

[AGENT] [is using TOOL] to inflate [OBJECT] [in PLACE]

[an AGENT] injects [SUBSTANCE] [from SOURCE] [into DESTINATION] [at PLACE]

[the AGENT] inserts [the OBJECT] [into CONTAINER] [at PLACE]

[an AGENT] installs [COMPONENT] [into DESTINATION] [using TOOL] [at PLACE]

[an AGENT] instructs [STUDENT] [at PLACE]

[AGENTS] intermingle [in PLACE]

[the AGENT] interrogated [the ADDRESSEE] [using the TOOL] [in the PLACE]

[the AGENT] interviews [the INTERVIEWEE] [at PLACE]

J.

[an AGENT] jogs [at PLACE]

[an AGENT] juggles [ITEMS] [in PLACE]

[an AGENT] jumps [from SOURCE] [over/through an OBSTACLE] [and will end up at DESTINATION] [at PLACE]

K.

[an AGENT] kicks [VICTIM] [in the VICTIMPART] [at PLACE]

Appendix B (Continued)

[the AGENT] kissed [the COAGENT's COAGENTPART] [with his/her AGENTPART] [at PLACE]

[the AGENT] kneads [an ITEM] [at PLACE]

[an AGENT] kneels [on the CONTACT] [at PLACE]

[an AGENT] knocks [on an ITEM] [at PLACE]

L.

[AGENT] laces [ITEM] [at PLACE]

[AGENT] lands [on DESTINATION] [at PLACE]

[an AGENT] laps [FOOD] [from CONTAINER] [at PLACE]

[an AGENT] lathers [SUBSTANCE] [into DESTINATION] [at PLACE]

[the AGENT] laughes [at PLACE]

[an AGENT] launches [an ITEM] [from SOURCE] [toward DESTINATION] [at PLACE]

[the AGENT] is leading [the FOLLOWER] [in PLACE]

[SUBSTANCE] leaks [from SOURCE] [onto/into DESTINATION] [at PLACE]

[an AGENT] leans [an ITEM] [against AGAINST] [at PLACE]

[the AGENT] leaps [from the SOURCE] [over/through an OBSTACLE] [to the DESTINATION] [in the PLACE]

[the AGENT] lectures [the AUDIENCE] [in PLACE]

[an AGENT] licks [an ITEM] [at PLACE]

[the AGENT] lifts [ITEM] [from START] [to END] [in PLACE]

[an AGENT] lights [an ITEM] [on fire using TOOL] [at PLACE]

[the AGENT] loads [DESTINATION] [with an ITEM] [using TOOL] [at PLACE]

[an AGENT] locks [an ITEM] [with TOOL] [in PLACE]

Appendix B (Continued)

M.

[an AGENT] makes [GOALITEM] [by manipulating COMPONENT] [using TOOL] [at PLACE]

[AGENT] manicures [ITEM] [using TOOL] [at PLACE]

[the AGENT] marches [in PLACE]

[an AGENT] mashes [an ITEM] [with TOOL] [in PLACE]

[an AGENT] massages [the COAGENTPART] [of the COAGENT] [at PLACE]

[an AGENT] measures [an OBJECTs' QUANTITY] [using TOOL] [at PLACE]

[the AGENT] mends [ITEM] [with TOOL] [in PLACE]

[an AGENT] microwaves [FOOD] [in CONTAINER] [at PLACE]

[an AGENT] milks [SOURCE] [with TOOL] [into DESTINATION] [in PLACE]

[an AGENT] mimes [an IMITATION] [at PLACE]

[an AGENT] mines [the RESOURCE] [with TOOL] [at PLACE]

[an AGENT] misbehaves [at PLACE]

[an AGENT] moistens [an ITEM] [with LIQUID] [at PLACE]

[the AGENT] [uses TOOL] to moisturize [MOISTURIZED] [at PLACE]

[an AGENT] molds [SUBSTANCE] [into GOALITEM] [in PLACE]

[the AGENT] is mopping [the SURFACE] [in the PLACE]

[the MOURNER] mourns [at the PLACE]

[an AGENT] mows [an ITEM] [with TOOL] [in PLACE]

N.

[the AGENT] nags [the NAGGEDPERSON] [at PLACE]

Appendix B (Continued)

[the AGENT] [uses the TOOL] to nail [ITEM1] [and ITEM2] [together in the PLACE]

[an AGENT] nips [an ITEM] [at PLACE]

[an AGENT] nuzzles [an ITEM] [at PLACE]

O.

[AGENT] offers [ITEM] [to BENEFICIARY] [at PLACE]

[the AGENT] officiates [an EVENT] [in PLACE]

[the AGENT] opens [the ITEM] [with the TOOL] [at the PLACE]

[the AGENT] operates [an ITEM] [with TOOL] [in PLACE]

[AGENT] is overflowing [from SOURCE] [in PLACE]

P.

[AGENT] packages [ITEM] [in PLACE]

[an AGENT] pack [an ITEM] [into CONTAINER] [at PLACE]

[an AGENT] paints [an ITEM] [with TOOL] [at PLACE]

[AGENT] panhandles [TARGET] [at PLACE]

[an AGENT] parades [in PLACE]

[the AGENT] [used CONNECTOR] to paste [an ITEM] [to an OBJECT] [in PLACE]

[AGENT] pats [ITEM] [using TOOL] [at PLACE]

[the AGENT] pawed [the PAWEDITEM] [using his/her AGENTPART] [at PLACE]

[an AGENT] pays [SELLER] [for GOOD] [at PLACE]

[the AGENT] pedals [his VEHICLE] [in the PLACE]

[AGENT] is peeing [in TARGET] [in PLACE]

Appendix B (Continued)

[an AGENT] peels [an ITEM] [with TOOL] [at PLACE]

[the AGENT] performing [the EVENT] [on the STAGE] [using TOOL] [at PLACE]

[AGENT] perspires [in PLACE]

[the AGENT] phones [using the TOOL] [at the PLACE]

[the AGENT] photographes [an ITEM] [with TOOL] [in PLACE]

[an AGENT] picks [the CROP] [from the SOURCE] [in PLACE]

[the AGENT] pilots [VEHICLE] [from START] [to END] [in PLACE]

[the AGENT] pinches [the OBJECT] [in the OBJECTPART] [at PLACE]

[AGENT] is pinning [the PINNED] [onto DESTINATION] [in PLACE]

[the AGENT] [uses TOOL] to pitch [the PITCHED] [at PLACE]

[AGENT] placed [PLACEDITEM] [in DESTINATION] [along ALREADYPLACEDITEM] [in PLACE]

[the AGENT] [use TOOL] to plant [PLANTED] [in PLACE]

[AGENT] is plowing [with the INSTRUMENT] [in PLACE]

[the AGENT] is plummeting [from the START] [toward the DESTINATION] [in the PLACE]

[an AGENT] plunges [PLUNGED] [into DESTINATION] [at PLACE]

[the AGENT] poked [the OBJECT] [in its OBJECTPART] [using TOOL] [in PLACE]

[AGENT] poos [onto DESTINATION] [at PLACE]

[an AGENT] pots [an ITEM] [in CONTAINER] [at PLACE]

[an AGENT] pounces [onto the DESTINATION] [at PLACE]

[an AGENT] pours [SUBSTANCE] [from SOURCE] [to DESTINATION] [with TOOL] [in PLACE]

[an AGENT] pouts [at PLACE]

Appendix B (Continued)

[an AGENT] practices [SKILL] [using TOOL] [at PLACE]

[the AGENT] prays [at PLACE]

[an AGENT] preaches [to ADDRESSEE] [in PLACE]

[the AGENT] presses [ITEM] [in PLACE]

[the AGENT] pricks [the PRICKED] [with the TOOL] [in the PLACE]

[an AGENT] protests [at PLACE]

[an AGENT] provides [RECIPIENT] [with an ITEM] [from SOURCE] [in PLACE]

[AGENT] is prowling [for TARGET] [in PLACE]

[an AGENT] prunes [REMOVEDITEM] [from SOURCE] [using TOOL] [at PLACE]

[the AGENT] pries [the ITEM] [from the FROM] [using TOOL] [at PLACE]

[the AGENT] puckers [his/her AGENTPART] [at PLACE]

[the AGENT] pulls [an ITEM] [with TOOL] [at PLACE]

[an AGENT] pumps [SUBSTANCE] [from SOURCE] [to DESTINATION] [using TOOL] [at PLACE]

[AGENT] is punching [VICTIM's BODYPART] [in PLACE]

[an AGENT] punts [ITEM] [at PLACE]

[an AGENT] pushes [an ITEM] [with an AGENTPART] [at PLACE]

[an AGENT] puts [an ITEM] [into DESTINATION] [in PLACE]

Q.

[the AGENT] is queueing [in PLACE]

R.

[the AGENT] races [against the COMPETITOR] [at PLACE]

Appendix B (Continued)

[an AGENT] rafts [at PLACE]

[an AGENT] rakes [an ITEM] [from SOURCE] [into DESTINATION] [at PLACE]

[an AGENT] rams [the VICTIM] [with RAMMINGITEM] [at PLACE]

[an AGENT] reads [an ITEM] [at PLACE]

[AGENT] is rearing [in PLACE]

[an AGENT] reassures [the REASSURED] [at PLACE]

[the AGENT] records [PHENOMENON] [in PLACE]

[the AGENT] recovers [from an AILMENT] [at PLACE]

[the AGENT] recuperates [at the PLACE]

[AGENT] rehabilitates [ITEM] [at PLACE]

[an AGENT] releases [RELEASEDITEM] [from PLACE]

[the AGENT] repairs [ITEM's PROBLEM] [using TOOL] [in PLACE]

[the AGENT] rests [ITEM] [on GOAL] [in PLACE]

[an AGENT] restrained [the RESTRAINED] [in PLACE]

[the AGENT] retrieves [the OBJECT] [from START] [in PLACE]

[an AGENT] rides [then VEHICLE] [at PLACE]

[AGENT] is rinsing [OBJECT] [using TOOL] [in PLACE]

[the AGENT] rocks [ROCKED] [in CONTAINER] [in PLACE]

[AGENT] rots [in CONTAINER] [at PLACE]

[an AGENT] rows [VEHICLE] [at PLACE]

[the AGENT] rubs [ITEM] [with AGENTPART] [in PLACE]

Appendix B (Continued)

[AGENT] runs [at PLACE]

S.

[AGENT] is saluting [TARGET] [in PLACE]

[an AGENT] scolds [VICTIM] [in PLACE]

[AGENT] is scooping [ITEM] [from SOURCE] [using TOOL] [in PLACE]

[the AGENT] scores [in PLACE]

[the AGENT] scrapes [the SCRAPEDITEM] [with TOOL] [at the PLACE]

[the AGENT] scratches [the OBJECT] [using TOOL] [at PLACE]

[an AGENT] scrubs [an ITEM] [with TOOL] [at PLACE]

[an AGENT] seals [an ITEM] [with SEALANT] [at PLACE]

[SELLER] sells [an ITEM] [to BUYER] [at PLACE]

[the AGENT] serves [an ITEM] [to the SERVED] [at PLACE]

[the AGENT] sews [the ITEM] [with the TOOL] [in the PLACE]

[AGENT] shakes [ITEM] [using TOOL] [at PLACE]

[the AGENT] sharpens [ITEM] [with TOOL] [in PLACE]

[an AGENT] shaves [COAGENT'sBODYPART using TOOL] [with the help of SUBSTANCE] [at PLACE]

[an AGENT] shears [an ITEM] [from SOURCE] [at PLACE]

[the AGENT] shells [the OBJECT] [in PLACE]

[the AGENT] shelves [an ITEM] [on DESTINATION] [in PLACE]

[an AGENT] shivers [at PLACE]

[the AGENT] shoots [PROJECTILE] [from the FIREARM] [at TARGET] [in PLACE]

Appendix B (Continued)

[the AGENT] shops [for GOODS] [in PLACE]

[an AGENT] shouts [at an ADDRESSEE] [in PLACE]

[the AGENT] shovels [the ITEM] [from the SOURCE] [in the PLACE]

[an AGENT] shreds [an ITEM] [using TOOL] [at PLACE]

[an AGENT] shrugs [at an ADDRESSEE] [at PLACE]

[the AGENT] shushes [the TARGET] [at the PLACE]

[the AGENT] signals [the MESSAGE] [to the RECIPIENT] [using the TOOL] [in the PLACE]

[the AGENT] signs [the SIGNEDITEM] [with the TOOL] [at the PLACE]

[the AGENT] sings [in the PLACE]

[an AGENT] sits [on CONTACT] [at PLACE]

[an AGENT] skates [by using VEHICLE] [at PLACE]

[the AGENT] sketches [an IMAGE] [on MATERIAL] [with TOOL] [in PLACE]

[AGENT] is skidding [in PLACE]

[an AGENT] skis [in PLACE]

[an AGENT] skips [over an OBSTACLE] [at PLACE]

[AGENT] slaps [VICTIM] [in the VICTIMPART] [with TOOL] [in PLACE]

[the AGENT] sleeps [in the PLACE]

[the AGENT] slices [the SLICEDITEM] [using TOOL] [at the PLACE]

[the AGENT] slides [the SLIDER] [on SURFACE] [to DESTINATION] [at PLACE]

[an AGENT] slips [onto DESTINATION] [at PLACE]

[an AGENT] slithers [in PLACE]

Appendix B (Continued)

[an AGENT] slouches [on the CONTACT] [at PLACE]

[an AGENT] smashes [the SMASHED] [with TOOL] [against AGAINST] [at PLACE]

[the AGENT] smears [an ITEM] [on SURFACE] [with TOOL] [at PLACE]

[the AGENT] smells [ITEM] [in PLACE]

[the AGENT] smiles [in PLACE]

[the AGENT] sneezed [at the PLACE]

[an AGENT] sniffs [an ITEM] [in PLACE]

[an AGENT] snuggles [with COAGENT] [at PLACE]

[an AGENT] soaks [an ITEM] [in SUBSTANCE] [in CONTAINER] [at PLACE]

[an AGENT] soares [in PLACE]

[the AGENT] socializes [with COAGENT] [in PLACE]

[an AGENT] sows [with TOOL] [at PLACE]

[the AGENT] spanks [the VICTIM] [with the TOOL] [in the PLACE]

[the AGENT] speaks [to COAGENT] [in PLACE]

[the AGENT] spears [the VICTIM] [in PLACE]

[an AGENT] spills [SUBSTANCE] [from SOURCE] [onto DESTINATION] [at PLACE]

[an AGENT] spins [MATERIAL] [with TOOL] [in PLACE]

[the AGENT] spits [an ITEM] [on the TARGET] [at PLACE]

[an AGENT] splashes [DESTINATION] [with SUBSTANCE] [using TOOL] [in PLACE]

[the AGENT] spoils [at PLACE]

[an AGENT] sprays [SUBSTANCE] [onto DESTINATION] [from SOURCE] [using TOOL] [in PLACE]

Appendix B (Continued)

[an AGENT] spreads [SUBSTANCE] [onto SURFACE] [using TOOL] [at PLACE]

[an AGENT] sprinkles [an ITEM] [from SOURCE] [onto DESTINATION] [at PLACE]

[an AGENT] sprints [at PLACE]

[SPROUTER] sprouts [at PLACE]

[an AGENT] spies [on TARGET] [with TOOL] [in PLACE]

[an AGENT] squeezes [an ITEM] [with TOOL] [at PLACE]

[the AGENT] squints [at the ITEM] [in the PLACE]

[an AGENT] stacks [TOP] [onto BOTTOM] [in PLACE]

stampede [in PLACE]

[AGENT] is standing [in PLACE]

[an AGENT] staples [ITEM] [onto SURFACE] [using TOOL] [in PLACE]

[the AGENT] stares [at ITEM] [in PLACE]

[the AGENT] steers [the VEHICLE] [with the TOOL] [in the PLACE]

[the AGENT] stings [the VICTIM] [on the VICTIMPART] [in PLACE]

[AGENT] stirs [ITEM] [in CONTAINER] [using TOOL] [at PLACE]

[the AGENT] stitches [using the TOOL] [and the FASTENER] [in PLACE]

[AGENT] is stooping [at PLACE]

[an AGENT] straps [the STRAPPED] [into DESTINATION] [using STRAP] [at PLACE]

[AGENT] is stretching [ITEM] [in PLACE]

[an AGENT] strikes [the AGENTPART] [of COAGENT] [using TOOL] [at PLACE]

[AGENT] is stripping [REMOVEDITEM] [from SOURCE] [using TOOL] [in PLACE]

Appendix B (Continued)

[the AGENT] is stroking [the OBJECT] [on the PART] [in the PLACE]

[AGENT] is studying [in PLACE]

[the AGENT] stuffs [the ITEM] [in the DESTINATION] [at the PLACE]

[an AGENT] stumbles [onto DESTINATION] [at PLACE]

[an AGENT] subdues [TARGET] [in PLACE]

[the AGENT] submerges [the OBJECT] [in SUBSTANCE] [at PLACE]

[an AGENT] sucks [on an ITEM] [in PLACE]

[AGENT] is surfing [PATH] [using TOOL] [at PLACE]

[an AGENTTYPE] swarms [at PLACE]

[the AGENT] sweeps [the SURFACE] [with the BRUSH] [in the PLACE]

[the AGENT] swims [in PLACE]

[an AGENT] swings [on CARRIER] [at PLACE]

[AGENT] is swooping [in PLACE]

T.

[an AGENT] tackles [VICTIM] [in PLACE]

[the AGENT] talks [to the LISTENER] [in MANNER] [in PLACE]

[an AGENT] tapes [an ITEM] [to DESTINATION] [at PLACE]

[the AGENT] tastes [the ITEM] [with the TOOL] [in the PLACE]

[AGENT] tattooed [TARGET] [with TOOL] [in PLACE]

[the AGENT] taxies [on the GROUND] [at the PLACE]

[the TEACHER] to teach [the STUDENT] [at PLACE]

Appendix B (Continued)

[the AGENT] tears [the ITEM] [with the TOOL] [in the PLACE]

[AGENT] telephones [at PLACE]

the [AGENT] [throws an ITEM] [towards DESTINATION] [at PLACE]

[the AGENT] tickled [TICKLED] [with an OBJECT] [in PLACE]

[an AGENT] tills [soil with TOOL] [at PLACE]

[an AGENT] tilts [an ITEM] [with their AGENTPART] [at PLACE]

[an AGENT] tips [an ITEM] [with its AGENTPART] [in PLACE]

[AGENT] tows [ITEM] [onto DESTINATION] [using TOOL] [at PLACE]

[AGENT] is training [STUDENT] [in PLACE]

[AGENT] is trimming [ITEMPART] [of ITEM] [with TOOL] [in PLACE]

[an AGENT] trips [over an ITEM] [onto DESTINATION] [in PLACE]

[the AGENT] tugs [the ITEM] [in the PLACE]

[AGENT] tunes [OBJECT] [with TOOL] [in PLACE]

[AGENT] is turning [TURNEDITEM] [in PLACE]

[an AGENT] twirls [COAGENT] [in PLACE]

[AGENT] is twisting [AGENTPART] [at PLACE]

[AGENT] is typing [with TOOL] [in PLACE]

U.

[the AGENT] uncorks [CONTAINER] [using TOOL] [in PLACE]

[an AGENT] unloads [an ITEM] [from SOURCE] [using TOOL] [at PLACE]

[an AGENT] unlocks [CONTAINER] [by opening LOCK] [using TOOL] [in PLACE]

Appendix B (Continued)

[the AGENT] unpacks [ITEM] [from CONTAINER] [in PLACE]

[the AGENT] [uses TOOL] to unplug [UNPLUGGED] [at PLACE]

[the AGENT] unveils [the OBJECT] [in PLACE]

[the AGENT] urinates [onto the TARGET] [at PLACE]

V.

[the AGENT] [uses TOOL] to vacuum [the SURFACE] [at PLACE]

[an AGENT] vaults [from START] [over OBSTACLE] [to END] [using TOOL] [at PLACE]

[an AGENT] videotapes [the DEPICTED] [at PLACE]

[the AGENT] votes [for VOTEFOR] [at PLACE]

W.

[the AGENT] to waddle [at PLACE]

[AGENT] wades [through SUBSTANCE] [at PLACE]

[an AGENT] wags [the AGENTPART] [at an ADDRESSEE] [in PLACE]

[an AGENT] waits [at PLACE]

[an AGENT] walks [at PLACE]

[an AGENT] washes [an ITEM] [of DIRT] [using TOOL] [in PLACE]

[AGENT] is watering [RECIPIENT] [with TOOL] [in PLACE]

[the AGENT] waves [the AGENTPART] [in the PLACE]

[an AGENT] waxes [COAGENT's BODYPART] [at PLACE]

[an AGENT] weeds [with TOOL] [at PLACE]

[an AGENT] weeps [at PLACE]

Appendix B (Continued)

[an AGENT] weighs [the MASS] [with TOOL] [in PLACE]

[the AGENT] welds [the ITEM] [to the SURFACE] [in the PLACE,] [using the TOOL]

[the AGENT] wets [the OBJECT] [with LIQUID] [using TOOL] [in PLACE]

[an AGENT] wheels [an ITEM] [on CARRIER] [at PLACE]

[AGENT] whips [ITEM] [using TOOL] [at PLACE]

[AGENT] whirls [at PLACE]

[the AGENT] whisks [ITEM] [in CONTAINER] [in PLACE]

[the AGENT] whistles [with TOOL] [in PLACE]

[the AGENT] wilts [in the PLACE]

[an AGENT] winks [at the ADDRESSEE] [at PLACE]

[an AGENT] wipes [SUBSTANCE] [from SOURCE] [with TOOL] [at PLACE]

[an AGENT] works [on FOCUS] [to achieve goal in PLACE]

[AGENT] is wrapping [WRAPPEDITEM] [with WRAPPINGITEM] [in PLACE]

[the AGENT] wrings [the ITEM] [at the PLACE]

[the AGENT] wrinkles [his/her AGENTPART] [at PLACE]

[AGENT] writes [on TARGET] [using TOOL] [at PLACE]

Y.

[an AGENT] yanks [the YANKED] [by the YANKEDPART] [in PLACE]

[an AGENT] yawns [at PLACE]

Appendix C

COPY RIGHT

C.1 LREC 2020 paper copyright

https://lrec2020.lrec-conf.org/en/submission2020/authors-kit/

The Language Resource and Evaluation Conference (LREC) proceedings are published by the European Language Resources Association (ELRA). They are available online from the conference website.

ELRA's policy is to acquire copyright for all LREC contributions. In assigning your copyright, you are not forfeiting your right to use your contribution elsewhere. This you may do without seeking permission and is subject only to normal acknowledgement to the LREC proceedings. The LREC 2020 Proceedings are licensed under CC-BY-NC, the Creative Commons Attribution-NonCommercial 4.0 International License.

Your submission of a finalized contribution for inclusion in the LREC Proceedings automatically assigns the above-mentioned copyright to ELRA.

C.2 IEEE ICSC 2020 paper copyright

https://ieeexplore.ieee.org/abstract/document/9031456

Appendix C (Continued)

CITED LITERATURE

[Anderson et al., 2018] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6.

[Andreas et al., 2016] Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2016). Deep compositional question answering with neural module networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Antol et al., 2015] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. (2015). VQA: Visual Question Answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

[Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *Lecture Notes in Computer Science*, 4825:722–735.

[Caruana, 1997] Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

[Choi and Palmer, 2012] Choi, J. D. and Palmer, M. (2012). Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 363–367. Association for Computational Linguistics.

[Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*.

[Collobert and Weston, 2008] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

[Ferraro et al., 2015] Ferraro, F., Mostafazadeh, N., Huang, T.-H., Vanderwende, L., Devlin, J., Galley, M., and Mitchell, M. (2015). A survey of current datasets for vision and language research. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 207–213.

[Fillmore et al., 2003] Fillmore, C. J., Johnson, C. R., and Petruck, M. R. (2003). Background to FrameNet. *International journal of lexicography*, 16(3):235–250.

[Fukui et al., 2016] Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M. (2016). Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 457–468. ACL.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

[Goyal et al., 2017] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017). Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Johnson et al., 2017] Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997.

[Kafle and Kanan, 2017] Kafle, K. and Kanan, C. (2017). An analysis of visual question answering algorithms. In *International Conference on Computer Vision (ICCV)*.

[Kipper et al., 2008] Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2008). A Large-Scale Classification of English Verbs. *Journal of Language Resources and Evaluation*, 42(1):21–40.

[Krishna et al., 2017] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances In Neural Information Processing Systems (NIPS)*, pages 1097–1105.

[Le and Mikolov, 2014] Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

[Levin, 1993] Levin, B. (1993). *English Verb Classes and Alternations:˜A Preliminary Investigation*. University of Chicago Press.

[Liang et al., 2013] Liang, P., Jordan, M. I., and Klein, D. (2013). Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer.

[Liu et al., 2019] Liu, X., He, P., Chen, W., and Gao, J. (2019). Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

[Lu et al., 2016] Lu, J., Yang, J., Batra, D., and Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems (NIPS)*, pages 289–297.

[Ma et al., 2016] Ma, L., Lu, Z., and Li, H. (2016). Learning to answer questions from image using convolutional neural network. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 3, page 16.

[Malinowski and Fritz, 2014] Malinowski, M. and Fritz, M. (2014). A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances In Neural Information Processing Systems (NIPS)*, pages 1682–1690.

[Malinowski et al., 2017] Malinowski, M., Rohrbach, M., and Fritz, M. (2017). Ask your neurons: A deep learning approach to visual question answering. *International Journal of Computer Vision*, 125(1-3):110–135.

[Marcus et al., 1993] Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

[Martin and Jurafsky, 2009] Martin, J. H. and Jurafsky, D. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances In Neural Information Processing Systems (NIPS)*, pages 3111–3119.

[Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Noh and Han, 2016] Noh, H. and Han, B. (2016). Training recurrent answering units with joint loss minimization for VQA. *arXiv preprint arXiv:1606.03647*.

[Palmer, 2009] Palmer, M. (2009). Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the Generative Lexicon Conference*, pages 9–15. GenLex-09, Pisa, Italy.

[Palmer et al., 2005] Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Ren et al., 2015a] Ren, M., Kiros, R., and Zemel, R. (2015a). Exploring models and data for image question answering. In *Advances In Neural Information Processing Systems (NIPS)*, pages 2953–2961.

[Ren et al., 2015b] Ren, S., He, K., Girshick, R., and Sun, J. (2015b). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances In Neural Information Processing Systems (NIPS)*, pages 91–99.

[Ronchi and Perona, 2015] Ronchi, M. R. and Perona, P. (2015). Describing common human visual actions in images. In Xianghua Xie, M. W. J. and Tam, G. K. L., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 52.1–52.12. BMVA Press.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

[Seltzer and Droppo, 2013] Seltzer, M. L. and Droppo, J. (2013). Multi-task learning in deep neural networks for improved phoneme recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6965–6969.

[Shih et al., 2016] Shih, K. J., Singh, S., and Hoiem, D. (2016). Where to look: Focus regions for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4613–4621.

[Silberman and Fergus, 2012] Silberman and Fergus (2012). Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision (ECCV)*.

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Computational and Biological Learning Society*.

[Swayamdipta et al., 2017] Swayamdipta, S., Thomson, S., Dyer, C., and Smith, N. A. (2017). Frame-semantic parsing with softmax-margin segmental RNNs and a syntactic scaffold. *arXiv preprint arXiv:1706.09528*.

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

[Tieleman and Hinton, 2012] Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2):26–31.

[Wang et al., 2018] Wang, P., Wu, Q., Shen, C., Dick, A., and van den Hengel, A. (2018). FVQA: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2413–2427.

[Wu et al., 2017] Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., and van den Hengel, A. (2017). Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40.

[Wu et al., 2016] Wu, Q., Wang, P., Shen, C., Dick, A., and van den Hengel, A. (2016). Ask me anything: Free-form visual question answering based on knowledge from external sources. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4622–4630.

[Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

[Xie et al., 2015] Xie, S., Yang, T., Wang, X., and Lin, Y. (2015). Hyper-class augmented and regularized deep learning for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2645–2654.

[Yang et al., 2016] Yang, Z., He, X., Gao, J., Deng, L., and Smola, A. (2016). Stacked attention networks for image question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–29.

[Yatskar et al., 2016] Yatskar, M., Zettlemoyer, L., and Farhadi, A. (2016). Situation recognition: Visual semantic role labeling for image understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5534–5542.

[Zhang et al., 2019] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

[Zhou et al., 2015] Zhou, B., Tian, Y., Sukhbaatar, S., Szlam, A., and Fergus, R. (2015). Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*.

[Zhu et al., 2016] Zhu, Y., Groth, O., Bernstein, M., and Fei-Fei, L. (2016). Visual7w: Grounded question answering in images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4995–5004.

VITA

NAME: Mehrdad Alizadeh

EDUCATION: B.Sc. Computer Science, Shahid Beheshti University, Tehran, Iran, 2008

M.Sc. Artificial Intelligence, Tehran Polytechnic, Tehran, Iran, 2011

HONORS: Fifty for the Future Award from Illinois Technology Foundation

PUBLICATIONS:

1. **M. Alizadeh** and B. Di Eugenio. (2020). Incorporating Verb Semantic Information in Visual Question Answering through Multitask Learning Paradigm. *Accepted in the International Journal of Semantic Computing (IJSC).*

2. **M. Alizadeh** and B. Di Eugenio. (2020). Augmenting Visual Question Answering with Semantic Frame Information in a Multitask Learning Approach. *In Proceedings of the 14th IEEE International Conference on Semantic Computing (ICSC)*, San Diego, CA, February 3-5. ***(Nominated for Best Paper Award)***

3. **M. Alizadeh** and B. Di Eugenio. (2020). A Corpus for Visual Question Answering Annotated with Frame Semantic Information. *In Proceedings of the 12th Language Resources and Evaluation Conference (LREC).* Marseille, France, May 11-16.

4. O. AlZoubi, B. Di Eugenio, D. Fossati, N. Green, and **M. Alizadeh**. (2020). Learning Recursion: Insights from the ChiQat Intelligent Tutoring System. *In Proceedings of the 12th International Conference on Computer Supported Education (CSEDU).* May 2-4.

5. R. Harsley, N. Green, **M. Alizadeh**, S. Acharya, D. Fossati, B. Di Eugenio, O. AlZoubi. Incorporating Analogies and Worked Out Examples as Pedagogical Strategies in a Computer Science Tutoring System. *Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE)*. Memphis, TN, USA, 2016

6. N. Green, D. Fossati, B. Di Eugenio, R. Harsley, O. AlZoubi, **M. Alizadeh**. Student Behavior with Worked-out Examples in a Computer Science Intelligent Tutoring System. *3rd International Conference on Educational Technologies.* Florianópolis, Santa Catarina, Brazil, 2015

7. B. Di Eugenio, N. Green, O. AlZoubi, **M. Alizadeh**, R. Harsley, D. Fossati. Worked-out Examples in a Computer Science Intelligent Tutoring System. *The 16th Annual Conference on Information Technology Education.* Chicago, IL, 2015

8. O. AlZoubi, D. Fossati, B. Di Eugenio, N. Green, **M. Alizadeh**, R. Harsley. A Hybrid Model for Teaching Recursion. *The 16th Annual Conference on Information Technology Education.* Chicago, IL, 2015

9. N. Green, O. AlZoubi, **M. Alizadeh**, B. Di Eugenio, D. Fossati, R. Harsley. A Scalable Intelligent Tutoring System Framework for Computer Science Education. *7th International Conference on Computer Supported Education (CSEDU'15).* May 2015

10. **M. Alizadeh**, B. Di Eugenio, R. Harsley, N. Green, D. Fossati, O. AlZoubi. A Study of Analogy in Computer Science Tutorial Dialogues. *7th International Conference on Computer Supported Education (CSEDU'15).* May 2015