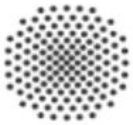


Universität Stuttgart

Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Master Thesis

Integrating Common Sense Knowledge in Visual Question Answering Systems

Adrika Mukherjee

Study program:	M.Sc. Infotech
Examiner:	Prof. Dr. Thang Vu
Advisor:	Pascal Tilli
start date:	15.01.2021
end date:	15.09.2021

Acknowledgement

I want to express my sincere gratitude towards my Supervisor Pascal Tilli. He guided me in the right direction and provided me with necessary support from time to time.

I would also like to thank Prof. Dr. Thang Vu for his valuable input during the entire duration of my thesis.

Abstract

Visual Question Answering (VQA) is a long-standing problem in the junction of Natural Language Processing and Computer Vision. It is a task that requires developing a system to answer a natural language question about an image. This thesis explores knowledge-based VQA Systems. It is different from vanilla VQA in the sense that it requires additional external knowledge on top of visual attributes from the given image to produce the correct answer.

The use of entire knowledge bases for training like ConceptNet [1] , ATOMIC [2], WebChild [3] etc. is common in knowledge-based VQA architectures e.g CONCEPT-BERT [4]. Narasimhan et al. [5] and Ziaeefard et al. [6], proposed different knowledge retrieval and filtering methods which can be used to augment knowledge, but they are neither inerrant nor generalizable. These approaches have a drawback, given that a lot of irrelevant and unwanted facts generate noise, and the model cannot determine the fact required to answer the question.

The common sense knowledge-based dataset, FVQA [7] has only one triple mapped to each question. This knowledge triple is essential to answer the given question. However, there is a need for having a more generalized system that can handle questions that require more than one fact to answer a question. Again, there may not be any existing fact in the Knowledge Base (KB) that is sufficient to answer the question. A more fundamental understanding of different objects around us is required. To this end, a new architecture is proposed which uses the automatic knowledge graph construction method formulated in COMET [8] to generate graph structures or set of fact triples relevant to a given image and question. This set of acquired triples is used to answer the given question for an image. Aditionally, This thesis work comprehensively studies existing approaches for knowledge-based VQA architectures and analyze their shortcomings. Finally, a novel pipeline-based architecture to integrate common sense knowledge into VQA systems, using an automatic knowledge graph construction mechanism, is proposed.

COMET [8], trained on ConceptNet[1] is adopted as the source of general knowledge. The performance of the model is assessed on the challenging FVQA [7] dataset. This dataset was build keeping in mind a particular fact from a KB, associated with each Image-question pair which is essential to answer the question. Hence the model's generalisability and degree of common sense grasping capability is rightly estimated by using FVQA dataset since COMET is not entitled to generate the exact triple required to answer the question. A set of experiments were conducted by using various knowledge graph embedding techniques, different modality combinations (image, knowledge, and question), and, combination of different modules in the pipeline. A comparison between two different attention-based models for final training was made. Finally, the best combination is compared with other SOTA approaches. The outcomes are equivalent to the performance of existing methods even though an actual KB or filtered triples from an actual KB was not always used for training. An accuracy of 61.92% was achieved when the Stacked-attention-based model [9] is applied for training with image, question, and knowledge features. Conclusively, a meticulous analysis of the model and future possibilities for extensions are reported.

Contents

Abbreviations

CNN Convolution Neural Network

COMET Commonsense Transformers for Automatic Knowledge Graph Construction

GAT Graph Attention Networks

GCN Graph Convolution Network

KB Knowledge Base

KG Knowledge Graph

LDA Linear Discriminant Analysis

LSTM Long Short-Term Memory

MLP Multi Layer Perceptron

PCA Principal Component Analysis

RDF Resource Description Framework

RNN Recurrent Neural Network

SAN Stacked Attention Network

SN Semantic network

SOTA State of the art

SPARQL SPARQL Protocol and RDF Query Language

VQA Visual Question Answering

1 Introduction

1.1 Knowledge-based VQA System

In the traditional VQA approach, no external knowledge is required to answer a large majority of questions. The most common questions such as "How many mangoes are there in the image?" "Which animal is this?" and "What color is the cup?" can be answered just by looking at the image. While these are ideally plausible tasks for a vanilla VQA system, they do not examine the models' capability to contemplate a scene or pull on knowledge outside of the image. Knowledge-based VQA requires external knowledge to answer a question. An example of image question pair that requires external knowledge to be answered correctly is shown in Figure 1.

Question: What can the red object on the ground be used for?
Answer: Firefighting
Support Facts: Fire hydrants can be used for fighting fires

Figure 1: Image, question pair from FVQA dataset that requires external common sense knowledge to answer the question, taken from [7]

1.2 Research Question

Acquiring knowledge for VQA is a taxing task, given that there are different types of knowledge that humans possess surrounding the world varying from everyday life knowledge, language, and culture, cuisine and lifestyle Etc. Thus, the goal of combining visual recognition with information extraction from sources outside the image has given rise to several important research questions, including What external knowledge sources should be adopted given that there is a huge set of information out in the open? How to integrate this

knowledge into existing VQA systems? How to improve the general knowledge understanding of a model such that it delivers on real time scenarios where the question asked may not be directly mapped to a fact in the KB?. In existing works, predefined knowledge bases like ConceptNet [1] , ATOMIC [2], WebChild [3], DBPedia [16] is used to generate dataset like FVQA [7] and KB-VQA [13]. On the contrary, OK-VQA [14] has garnered much praise owing to its questions based on open-domain knowledge. Although OK-VQA is built without keeping any database in mind, most questions in this dataset are related to world knowledge and hence require DBPedia [16]. In this thesis, the focus is narrowed down to common sense knowledge which an adult can answer with no external World or General knowledge. These questions are fundamental but still require some external support and cannot fully be answered by the embedded image and question features.

One of the most interesting research question pertaining to this setting is, What is the most plausible way to build a generalized system that integrates common sense knowledge and Which KB can be used for it and how it can be used? There is no limit to knowledge but there is a limitation of hardware so its not feasible to integrate multiple high dimensional knowledge bases. ConceptNet [1] is a well-known source for common sense knowledge; however, is it necessary to combine the entire knowledge base with each question and image data point during training? Are the various filtering and knowledge retrieval approaches described in [5], [10] stable, reproducible, and most importantly applicable on a dataset build on open domain knowledge? Will the model be bombarded with noisy data while trying to filter triples from existing KB? If so then, how to mitigate this problem?

1.3 Overview

The goal of this thesis is to find a suitable method to integrate knowledge such that there is minimum static knowledge base dependency. The flow of knowledge should be dynamic and intelligently generated by a human-level knowledge producing model trained on some of the already available knowledge bases like [1] [2] [3]. These models can complete or generate new knowledge with the natural language understanding of nodes, which is developed using the knowledge base it is trained on. The knowledge generated by these models is at par with human-level reasoning. This model can be exploited to generate subgraphs based on visual concepts, questions and embed these subgraphs as features to predict the answers. This approach eliminates the requirement of an existing static knowledge base for integrating knowledge into VQA systems. For example, an automatic knowledge graph construction method completes a triple by generating objects (an entity) given a subject (another entity) and relation (the link between the two entities). The subject can be generated using visual concepts like the scene, action, or object detected from the image; the question can produce the relation. The knowledge completion model then generates complete triples based on the subject-relation input combination. A triple is a set of two entities with a relation between them. The circles denote entities, and the dashed line denotes the relation in Figure 2. It gives an idea of how a set of triples is represented using a graph data structure.

Using finite knowledge bases for knowledge integration has its limitations (what if the fact required to answer the question is not present in the existing KB) which set forth a sturdy research ground for this thesis. Previous methods use biased filtering-based knowl-

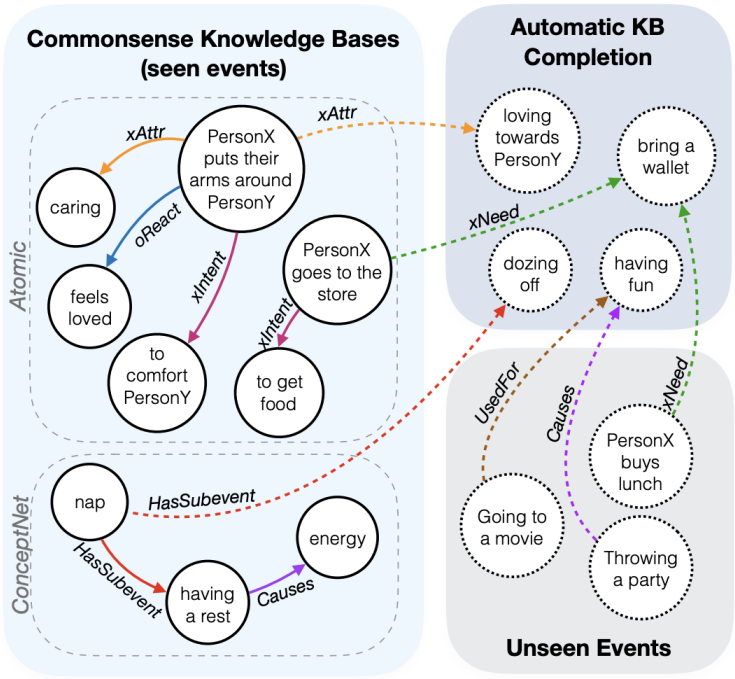


Figure 2: COMET generated novel nodes when trained on specific dataset like ConceptNet or ATOMIC, taken from [8]

edge retrieval techniques from existing full KBs. Sometimes Redundant and unnecessary information from the KB is filtered and fed into the model. It is not clear how these filtered triples upgrade the accuracy of the model. They may also fail to answer the question which requires combined knowledge from several triples rather than a specific one. Thus, the model might lack a generalized understanding and may only perform well when the specific triple used to answer the question is present. Again some approaches involve question to SPARQL query mapping [7], which gives rise to an even more significant cascading error effect in the final answer prediction. The thesis aims to conduct experiments and find the optimal way to incorporate external commonsense knowledge while answering a common sense-related question. The proposed approach to tackle the challenge involves generating comprehensible knowledge-based subgraphs (set of triples) using COMET based on visual attributes extracted from the image and relations extracted from the question. A pipeline architecture is proposed that prepares the subject and relation fed into the knowledge graph completion module that generates the subgraph. The final model is then built, it is trained on the image, question, and generated knowledge subgraph features to produce the answer. Two different attention-based final model architecture is proposed, a comparison between them is drawn, and the best one is used for the final analysis.

2 Related Works

2.1 Knowledge-based Visual Question Answering Models

2.1.1 Towards Knowledge-Augmented Visual Question Answering [6]

Visual Question Answering (VQA) continues to pose algorithmic difficulties while being effortless for humans. To answer a question concerning a given image, humans blend visual data with general and common sense information. Ziaeefard et al. [6] address the issue of adding general knowledge into VQA models while exploiting visual information in this study. They present a model that finds interaction of objects in a given image with entities in an external knowledge source. The model utilizes a graph-based technique that blends scene and knowledge graphs to build an adaptive graph representation of connected instances. The authors employ Graph Attention Networks to prioritise critical cases of knowledge that are most relevant to each question and use ConceptNet as a source of general knowledge, and evaluate the model's performance on the highly complex OK-VQA dataset [6].

This work [6] makes the following significant contributions:
1. A novel technique for incorporating contextual knowledge into VQA models, and unlike previous models, the authors omit the query construction stage. They do not rely on ground-truth facts, enabling them to incorporate any knowledge resources into the model [17]
2. They retrieve knowledge instances using sentence-level embeddings rather than word-level embeddings that encapsulate the semantic context of the questions and knowledge instances [17]
3. Concept graphs are developed using GAT [18] that operate on the neighbouring nodes to focus on crucial knowledge instances [17]
4. They use both Concept graphs and Scene graphs to embed the relationships between objects using [19] and [20].

2.1.2 ConceptBert:Concept-Aware Representation for Visual Question Answering [4]

Garderes et al. [4] explore a VQA model that consolidates both visual and text features in order to provide answers to image-based queries. Whereas VQA research focuses on questions that can be answered solely by analysis of the question and image, they introduce ConceptBert, a concept-aware algorithm for inquiries requiring common sense or basic factual knowledge gleaned from external structured content. Given a pair of an image and a natural language query, ConceptBert infers the proper response using visual elements from the image and a KG. They develop a multi-modal representation taking its inspiration from the popular BERT architecture and learn a joint Concept-Vision-Language embedding and encode common sense knowledge using ConceptNet KG [4]. Finally, they assess their methods using the Outside Knowledge-VQA (OK-VQA) [14] and VQA datasets.
Their approach is distinct from earlier methods in that it incorporates external information into the VQA problem using a BERT based concept and vision representation. As a result, the approach eliminates the requirement for extra knowledge annotations or search queries, significantly lowering computing costs [4].

In summary, [4] make the following contributions:
1. Incorporation of common sense knowledge into VQA models via a novel technique.
2. A concept-aware question representation and vision-language representation using VILBERT [21] is used.
3. The embedding comprising of Concept-Language and Vision-language are combined using Compact Trilinear Interaction [22].
However, explicit relations between entities are still not incorporated in this model.

2.1.3 Multi-Modal Answer Validation for Knowledge-Based VQA [10]

The problem of knowledge-based visual question answering entails responding to inquiries that require additional information in addition to the image's content. This type of knowledge often manifests itself in a variety of ways, including visual, textual, and commonsense. However, increasing the number of information sources raises the likelihood of recovering more irrelevant or noisy facts, making it more difficult to interpret the facts and choose the solution. To solve this issue, Wu et al. offer Multi-modal Response Validation with External Information (MAVEx), in which a group of promising answer choices is validated utilising answer-specific knowledge retrieval. This contrasts with present methodologies, which sift through a massive collection of frequently irrelevant facts in pursuit of the solution. Their technique seeks to determine which knowledge source should be trusted for each response candidate and how that source should be used to validate the candidate. The authors explore a multi-modal environment, relying on both textual and visual information resources, such as Google pictures, phrases from Wikipedia articles, and ConceptNet concepts. MAVEx produces new state-of-the-art outcomes in their studies with OK-VQA [14], a complex knowledge-based VQA dataset [10].

In summary, the distinction Wu et al. [10] provides in this work are:
1. They introduce a novel approach that utilises candidate answers to guide knowledge retrieval for open-domain VQA.
2. They employ multi-model knowledge extraction by examining visual and textual knowledge simultaneously.
3. They propose a consistency criterion for determining when to trust knowledge retrieved from each source.

2.1.4 Zero-shot Visual Question Answering using Knowledge Graph [11]

Chen et al. [11] observe that the existing methods rely heavily on pipeline systems comprised of many components for knowledge matching and extraction, feature learning, etc. However, such pipeline approaches suffer when a component performs poorly, resulting in error cascade and overall performance degradation – which can also be seen in the other related works presented here. Additionally, the bulk of existing techniques neglect the issue of answer bias – many answers may have never been observed during training (i.e., unseen replies) but may occur in a real-world application. To address these limitations, the authors offer a Zero-shot VQA algorithm that incorporates external knowledge via KG and a mask-based learning method, as well as new answer-based Zero-shot VQA splits for the F-VQA dataset. Experiments demonstrate that this strategy can surpass state-of-the-art

performance in Zero-shot VQA with unknown replies while augmenting existing end-to-end models significantly on the standard F-VQA challenge.

The following summarises the significant contributions in [11]:
1. The authors propose a robust ZS-VQA algorithm based on KGs that changes the answer prediction score via masking in two feature spaces based on the correlation between supporting entities/relations and fusion I-Q pairs.
2. They establish a novel ZS-VQA problem that demands external information and considers previously unseen solutions. As a result, they build an evaluation dataset for the ZS-F-VQA.
3. This ZS-VQA method is highly versatile because it is built on KG. It is capable of successfully addressing both traditional VQA tasks requiring external knowledge and ZS-VQA tasks and can be used to complement current end-to-end models.

2.1.5 Straight to the Facts: Learning Knowledge Base Retrieval for Factual Visual Question Answering [5]

The authors develop a strategy for knowledge retrieval that is based on learning. More precisely, their method discovers a parametric mapping between facts and question-image pairs and an embedding space. Then, they use the fact that is most aligned with the provided question-image pair to answer a question. This technique is capable of accurately answering both visual and fact-based queries. For example, given the image on the left and the question, "Which object in the image may be used to eat with?" the model can predict the correct response, "fork." Similarly, for the remaining two situations, the proposed approach is capable of predicting the correct answer [5].

This [5] has two significant advantages over previous work:
1. By avoiding explicit query generation, the authors avoid errors caused by synonyms, homographs, and incorrect prediction of visual concept type and answer type.
2. Their technique is easily extensible to any knowledge base, regardless of its structure or size. Additionally, they avoid the need for ad-hoc knowledge filtering. Instead, they learn to turn extracted visual concepts into a vector close to a relevant fact in the learnt embedding space. Finally, this system generates a score of facts deemed useful for the given question and image.

2.1.6 KRISP: Integrating Implicit and Symbolic Knowledge for Open-Domain Knowledge-Based VQA [12]

Marino et al. [12] observe that one of the most difficult sorts of questions in VQA is when the answer involves knowledge that is not included in the image. They investigate open-domain knowledge in this work, which refers to the situation in which the knowledge required to answer a question is not provided/annotated, neither during training nor during testing. They make use of two distinct categories of knowledge representations and reasoning. First, implicit information can be effectively learnt using transformer-based models from unsupervised language pretraining and supervised training data. The second type of information is explicit, symbolic knowledge that is encoded in knowledge bases. This technique combines the two approaches by utilising the obvious reasoning capability of transformer models for response prediction and fusing symbolic representations from KG without sacrificing their explicit semantics in the process. The authors combine multiple sources of knowledge in

order to encompass the breadth of knowledge required to address knowledge-based problems and demonstrate that the technique, KRISP (Knowledge Reasoning with Implicit and Symbolic Representations), outperforms the state-of-the-art significantly on OK-VQA, the largest dataset available for open-domain knowledge-based VQA [12].

The following are the distinctive contributions of this work [12]:
1. They introduce KRISP (Knowledge Reasoning with Implicit and Symbolic Representations), a unique paradigm that integrates explicit reasoning in a knowledge graph with implicit reasoning in a multi-modal transformer.
2. This model establishes a new state-of-the-art on the difficult Open Knowledge VQA dataset (OK-VQA), surpassing past work by a wide margin.
3. The authors extensively ablate their model in order to assess alternative ways for knowledge fusion.
4. They also examine how their model reasoned directly about facts and responded to queries by predicting answers from its knowledge network.

After careful study of previous works, a few un attended areas of concern can be summarised as follows: 1. There is no mention as to why a particular KB like ConceptNet[1] or Atomic [2]or Webchild[3] etc is used. Since all these knowledge bases have non intersecting set of facts. All the works integrate various combinations of KB without any justification. 2. The filtering technique is data dependent, and may not produce good result when the ground truth fact is not present in the KB 3. In some works it is assumed that a question necessarily has a fact complement in the KB and not otherwise. The above zones are tackled in the proposed architecture of this thesis.

2.2 Knowledge-based VQA datasets

2.2.1 FVQA [7]

Only questions involving external (non-visual) information are included in the FVQA dataset. It was developed with the goal of including more annotations to facilitate the supervised training of procedures employing knowledge bases. In contrast to the majority of VQA datasets, which contain simply question/answer pairs, FVQA provides a supporting fact for each question/answer pair. These facts are denoted by the triple (arg1, rel, arg2). Consider the following question/answer pair: "Why are these folks wearing yellow jackets?" For the sake of safety". It will incorporate the substantiating fact (wearing bright clothes, aids, safety). This dataset was created by extracting a significant number of such facts (triples) on visual concepts from the knowledge bases DBpedia [16], ConceptNet [1], and Webchild [3]. Annotators were required to select an image and a visual element within it, and then one of the pre-extracted supporting information connected to the visual concept. Finally, they were instructed to formulate a question/answer that directly addressed the supporting fact they had chosen. The collection contains 193,005 potential supporting facts for 4,608 queries associated with 580 visual concepts. [7].

2.2.2 KB-VQA [13]

The MS COCO [23] dataset contains 700 validation images, which were chosen for use in KB-VQA because of the wealth of contextual information and the variety of item classes included in them. In total, they cover approximately 150 object classes and 100 scene classes, with an average of 6 to 7 objects per class. There are 23 templates, eg:- Name: IsThereAny, Template: Is there any <concept>?; Name: WhatIs, Template: What is the <obj>? etc. Five human volunteers (questioners) were given the task of developing three to five question/answer pairings for each of the 700 photographs, by instantiating the 23 templates. In the templates various fields like <obj> or <concept> correspond to entities in existing knowledge base, DBpedia. The dataset contains 2,402 questions related to700 images [13] .

2.2.3 OK-VQA [14]

OK-VQA, which is an abbreviation for Open Knowledge VQA, is concerned with knowledge that is not associated with a single knowledge base. The photos in this dataset were selected at random from the COCO dataset. When compared to other datasets, these photos have a label of visual complexity, which makes them particularly suitable for labelling knowledge-based inquiries. In the first round of labelling, MTurk workers were asked to submit a question in response to an image that they were shown online. Users were given instructions on how to create questions that would deceive a "smart robot". The students were also given specific instructions that the question should be relevant to the picture's content. Also instructed not to ask questions such as "what is in an image?" or "how many of anything> are there?" and to explain if the query should necessitate the use of external information. Afterwards, during the second round of labelling, five separate MTurk employees were requested to label each question-image combination with an appropriate answer, and in this fashion, about 14,055 open knowledge questions were generated from 14,031 images [14].
A comparative description of different VQA datasets is shown in the Table 1

Dataset	**No. of questions**	**No.of Images**	**Goal**	**Answer Type**
KB-VQA [13]	2402	700	reasoning with given KB	open
FVQA% [7]	5826	2190	reasoning with given KB	open
OK-VQA [14]	14055	14031	reasoning with open KB	open

Table 1: Knowledge Based VQA dataset comparison [14]

2.3 Automatic Knowledge Base Construction

2.3.1 COMET : Commonsense Transformers for Automatic Knowledge Graph Construction [8]

The goal of the COMET model is to generate an adaptation framework for the building of common sense knowledge bases, i.e., the model is capable of providing basic logical inference from a given set of tuples containing things and their causal relationships. During training, these tuples supply COMET with the necessary KB structure and relations, and COMET learns to augment the seed knowledge graph with novel nodes and edges [8].

While COMET is independent of the language model with which it is initialised, its performance is affected by the weights associated with it, demonstrating that transfer learning from an established language model has an effect on the outcome when compared to randomly initialised weights. In the given work [8] initialisation is performed using weights from a transformer language model architecture described by Radford et al. (GPT)[24]. It encodes input text by combining many transformer blocks with multi-headed scaled dot product attention and fully connected layers [8].

The work's contributions can be summarised as follows. To begin with, a generative approach to knowledge base development is utilised to develop a model capable of creating new nodes and identifying edges between existing nodes by producing phrases that coherently fulfil an existing seed phrase and relation type. Second, a framework is designed for learning to produce common sense knowledge tuples utilising a large-scale transformer language model [8].

2.3.2 Commonsense Knowledge Base Completion with Structural and Semantic Context [15]

The purpose of this study is to provide a novel KB completion model that, by leveraging the structural and semantic context of nodes, can overcome a significant barrier to existing KB completion approaches, which require densely connected graphs across a relatively small set of nodes [15].

Two critical concepts are examined in detail. First, it utilizes graph convolutional networks and automatic graph densification, to learns from the local graph structure. And second, to transfer learning from pre-trained language models to knowledge graphs in order to create a more accurate and enriched contextual representation of information. This work demonstrates the efficacy of language model representations in improving link prediction performance, as well as the benefits of learning from local graph structure while training on subgraphs for computational efficiency. The graph embedding is generated using both GCN [25] and BERT [26]. Therefore, progressive masking for fusion is done to reduce dependency on BERT embeddings [15].

Finally, Convolutional models are used since it is known for having good completion rates for KBs. This study does the same thing through the use of a convolutional decoder. They score a tuple using the convolutional decoder CONVTRANSE [27] (subject, relation, object).

This model is based on ConvE but incorporates TransE's translational characteristic. They train the model using a binary cross entropy loss after computing scores for all candidate tuples. All target nodes that exist in the training set are considered positive instances, while all target nodes that do not exist are considered negative instances [15].

3 Background

This section consists of a brief introduction to topics which is required to comprehend the model architecture described later on. The knowledge data has a specific data structure, which is represented and embedded in a way different from natural language data. Also, a quick introduction to Transformers and the subsequent emergence of BERT [26] is discussed, since the idea from these architectures is used to develop the final model of the thesis. Some existing architectures from which the idea of the proposed architecture is drawn is discussed at the end of this section.

3.1 Knowledge Representation

Knowledge is a useful term for assessing an individual's comprehension of a subject. Representing knowledge is thought to be an essential strategy for encoding knowledge in any intelligent system. The primary goal of an AI system is to create programs that supply information to a computer so that it can interact with humans and solve problems in a variety of sectors[28]. Logical Representation, Semantic Network Representation, and Frame Representation are the four most common types of knowledge representation. This thesis focuses on the use of Semantic Networks to represent knowledge. Knowledge is expressed in SN by graphical networks. This network consists of nodes that represent entities and arcs that describe their relationships. Semantic networks may classify objects in a variety of ways and link them together. Semantic networks are simple to comprehend and expand upon[28].

In Figure 3, representation knowledge in the form of a graph is shown. Each subject node is connected with another object node by a directed link or relation[28].

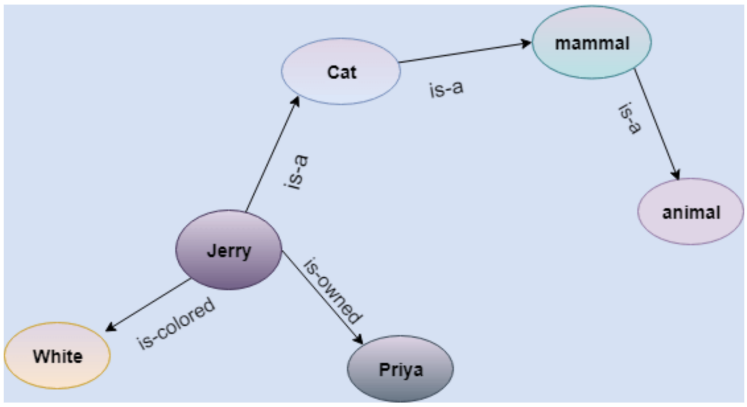


Figure 3: Semantic Network, taken from [28]

3.1.1 Drawbacks in Semantic representation

Because one needs to traverse the entire network tree to answer some questions, SN takes longer to compute at runtime. In the worst-case scenario, users might discover that the answer does not exist in this network after traversing the entire tree. This is because SN attempt to simulate human memory to store the information, but in reality, it is not feasible to build such a vast network. This is the reason why KB built are finite in nature; this is a noteworthy limitation of generic KG integration using KB. Furthermore, SN does not have any standard definition for the link names. As a result, these networks are not consistent and depend on their creator [28].

3.1.2 Advantages of Semantic network

Semantic networks are a natural way to represent information. They are used as a way of conveying meaning. These networks are straightforward to comprehend [28].

3.1.3 RDF Triple

RDF Triple is a tangible expression that defines a way in which one can represent a relationship between objects. A triple has three parts: subject, predicate, and object [29]. A graph can be viewed as a set of triples. In Figure 3, "$Jerry\ is-colored\ white$" is a Triple. Here the subject node is "Jerry". The directed edge is towards *white*, so this is the object node. The link is the relation between them.

3.1.4 RDF Framework

RDF is a standard for representing data on the internet [30]. In RDF, every piece of information is represented as a triple. Classes, properties, and vocabularies are used to express RDF data. These data are stored in databases known as RDF stores. A vocabulary is a data model that includes classes, characteristics, and relationships for describing data and information. A class is a construct that represents things in the physical and/or digital worlds, such as a person, an organization, or notions like "racism" or "well-being". A link is a relation between two classes, for example, between a document and the organisation that published it, or the link between a map and the geographic region that depicts it. In RDF, relationships are marked as object type properties. A property is a characteristic of a class in a certain dimension, such as an organisation's legal name. [31].

3.1.5 SPARQL

SPARQL is the standard language for querying graph data expressed as RDF triples [32]. Different Types of SPARQL queries are:

• SELECT

Return a table of all X, Y, etc. satisfying a given condition [32]

• CONSTRUCT

Return all X, Y, etc. satisfying given condition and insert them into a given template in order to generate RDF statements, thereby developing a new graph. [32]

• DESCRIBE

Find all statements in the dataset that provide information about given resource(s) ... (identified by name or description) [32]
In [7] LSTM is used to generate SPARQL query from a given question and image features, the generated SPARQL query is further used to retrieve knowledge from existing KBs for answering the question.

A sample SPARQL query with outputs are shown in Figure 4 [32].

Sample data

```
comp:A rov:haslegalName "Niké" .
comp:A org:hasRegisteredSite site:1234 .

Comp:B rov:haslegalName "BARCO" .

site:1234 locn:fullAddress "Dahliastraat 24, 2160 Wommelgem .
```

Query

```
PREFIX comp: < http://example/org/org/>
PREFIX org: < http://www.w3.org/TR/vocab-regorg/ >
PREFIX site: <http://example.org/site/>
PREFIX rov: <http://www.w3.org/TR/vocab-regorg/>

SELECT ?name

WHERE
 { ?x org:hasRegisteredSite site:1234 .
   ?x rov:haslegalName ?name  .}
```

Result

name
"Niké"

Figure 4: Sample data and SPARQL query with output, taken from [32]

3.1.6 Knowledge Graph

A knowledge graph, also known as an Semantic Network (SN), is a network of real-world items (i.e. objects, events, situations, or concepts) that depicts the link between them. The term "knowledge graph" comes from the fact that this information is frequently kept in a graph database and represented as a graph structure [33].

There have been several efforts to build graph-structured commonsense representations in [34] and [35]. Some prominent publicly knowledge graphs includes ConceptNet [1], ATOMIC [2], WebChild [3] etc. The term knowledge graph and knowledge base have been interchangeably used in some literature. In Figure 5, the "Fact" column represents the number of triples/facts in the knowledge graph or knowledge base. The 'Examples' column gives some examples of extracted facts in (subject, relation, object format) [7].

KB	Relationship	#Facts	Examples
DBpedia	Category	35152	(Wii, Category, VideoGameConsole)
ConceptNet	RelatedTo	79789	(Horse, RelatedTo, Zebra), (Wine, RelatedTo, Goblet), (Surfing, RelatedTo, Ocean)
	AtLocation	13683	(Bikini, AtLocation, Beach), (Tap, AtLocation, Bathroom)
	IsA	6011	(Broccoli, IsA, GreenVegetable)
	CapableOf	5837	(Monitor, CapableOf, DisplayImages)
	UsedFor	5363	(Lighthouse, UsedFor, SignalingDanger)
	Desires	3358	(Dog, Desires, PlayFrisbee), (Bee, Desires, Flower)
	HasProperty	2813	(Wedding, HasProperty, Romantic)
	HasA	1665	(Giraffe, HasA, LongTongue), (Cat, HasA, Claw)
	PartOf	762	(RAM, PartOf, Computer), (Tail, PartOf, Zebra)
	ReceivesAction	344	(Books, ReceivesAction, bought at a bookshop)
	CreatedBy	96	(Bread, CreatedBy, Flour), (Cheese, CreatedBy, Milk)
WebChild	Smaller, Better, Slower, Bigger, Taller, ...	38576	(Motorcycle, Smaller, Car), (Apple, Better, VitaminPill), (Train, Slower, Plane), (Watermelon, Bigger, Orange), (Giraffe, Taller, Rhino), (Skating, Faster, Walking)

Figure 5: Examples in different knowledge bases, taken from [7]

Some noteworthy KGs that are being used by the researchers working in this domain are described below:

WebChild [3]: The work in [3] considered a type of commonsense knowledge that has largely gone overlooked by the majority of existing knowledge bases, including comparison relations such as Faster, Bigger, and Heavier. [3] extracts this type of content automatically from the Web.

ConceptNet [1]: ConceptNet is a publicly available semantic network that tries to assist computers in interpreting the meanings of human-spoken words. ConceptNet evolved from the MIT Media Lab's 1999 crowdsourced project Open Mind Common Sense. It has grown to contain information since then from various other crowdsourcing sources, expert-created resources, and purpose-driven games. CN-100K contains general commonsense facts about the world [36].

DBpedia [16]: By crowdsourcing, the structured information in DBpedia is taken from Wikipedia. The project collects information from 111 distinct Wikipedia language versions. Over 400 million facts provide information about 3.7 million objects in the most extensive DBpedia knowledge base, which is derived from the English edition of Wikipedia [7].

ATOMIC [2]: The ATOMIC KG is a set of social commonsense information about everyday happenings. The entities' effects, needs, intents, and attributes are represented in the dataset. Nodes have a considerably longer average phrase length (4.40 words) than CN-100K [36]. A source entity and connection may have several targets or may have no target at all [37].

3.2 Knowledge Graph Embeddings

A directed heterogeneous graph with domain-specific relation types and nodes is a typical description for KG. KGs allow us to encapsulate information in a human-readable format that can be analyzed and inferred automatically. KGs are a popular way to represent many sorts of data in the form of various types of entities connected by various types of relationships called triples [38].

Low-dimensional approximations of the entities and relations in a KG are known as knowledge graph embeddings (KGEs). They provide a generalizable context about the whole

KG from which relationships can be inferred [38].

The knowledge graph embeddings are generated in such a way that they satisfy certain properties, such as adhering to a specific KGE model. In the low-dimensional embedding space, these KGE models define different score functions that assess the distance between two entities relative to their relation type. The KGE models are trained using these scoring functions such that entities linked by relations are close together, while entities not linked are far apart [38].

Many prominent KGE models specify different scoring functions to learn entity and relation embeddings, such as RotatE, TransR, TransE, DistMult, RESCAL, and, ComplEx. [39].

3.3 Graph Convolution Network

Knowledge graph embedding methods learn low-dimensional embeddings of entities and relations, which can be utilized for subsequent machine learning tasks, including link prediction and entity matching. Various graph convolutional network algorithms that use various sources of input to learn the characteristics of entities and relations have been developed [40].

Convolution in GCN is analogous to Convolutional Neural Networks' convolution layers. It is the process of multiplying the input neurons by a series of weights known as filters or kernels. CNN may learn information from surrounding cells with the help of the filters, which operate as a sliding window across the entire image. The same filter will be utilized throughout the image within the same layer, which is known as weight sharing [41]. When using CNN to identify images of cats as well as no cats, for example, the same filter will be utilized in the same layer to detect the cat's body parts, as shown in Figure 6.

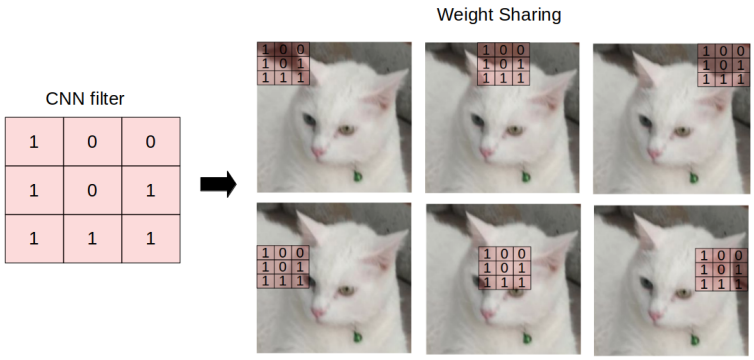


Figure 6: The same weight is applied throughout the image for CNN, taken from [41]

Similar actions are performed by GCNs, in which the model learns the features by analyzing surrounding nodes. The main distinction between CNNs and GCNs is that CNNs are designed to work with normal (Euclidean) organized data, whereas GCNs are a more generalized variant of CNNs in which the number of nodes connections varies and the nodes are not in any particular order.

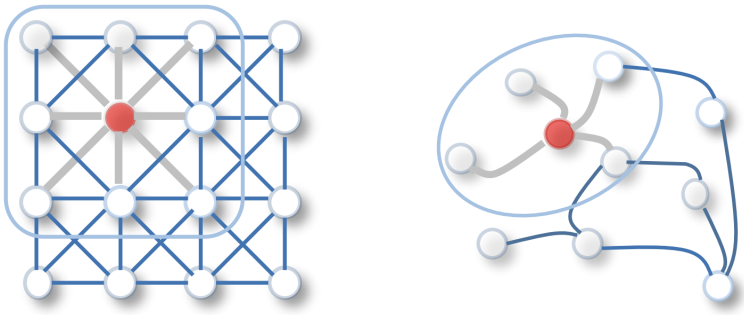


Figure 7: 2D Convolutional Neural Networks (left) and Graph Convolutional Networks (right), taken from [41]

In order to get a fair overview of GCN, which is required to understand the background of knowledge graph embeddings used in the thesis, the workings of vanilla neural network is revised as follows. This network is modified further to develop the equations for GCN. The forward propagation equation in neural network could be described as [41]:

$$H^{[i+1]} = \sigma(W^{[i]}H^{[i]} + b^{[i]})$$

where $H^{[i+1]}$ is feature represented at layer i+1, σ is activateion function, and $H^{[i]}$ is feature representation at layer i, $W^{[i]}$ represent weights at layer i, and $b^{[i]}$ represent bias at layer i.

For the first layer i.e. i = 0, the above equation can re-written as [41]:

$$H^{[1]} = \sigma(W^{T[0]}H^{1} + b^{[0]})$$

where input feature (X) is the representation at layer 0. The forward pass equation for Graph Convolutional Networks differs because it includes the adjacency matrix as an additional element as shown below [41]:

$$H^{[i+1]} = \sigma(W^{[i]}H^{[i]}A^{*})$$

where the normalized version of A is A^{*}

In GCN, Eigen-decomposition [42] aids in the understanding of graph structure and thus the classification of graph nodes. This is related to the basic notion of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), in which dimensionality is reduced and clustering is performed using Eigen-decomposition [41].

The Adjacency Matrix (A) in the forward propagation equation, as well as the node attributes (or so-called input features), are taken into consideration in this approach. In the forward propagation equation, A is a matrix that represents the edges or connections between the nodes. When A is included in the forward pass equation, the model is able to learn feature representations based on node connectivity. The bias b is removed to keep the calculations simple [41].

3.4 Transformer

Transformer is initially introduced in the publication 'Attention Is All You Need' [43]. It employs the attention-mechanism, as the term suggests. Transformer, just like LSTM,

is an architecture for converting one sequence into another using two parts (Encoder and Decoder). However, it differs from the previously described/existing sequence-to-sequence models in that it does not involve Recurrent Networks like LSTM [44]. Up to this point, one of the finest approaches to preserve the temporal dependencies in sequences was to use recurrent networks. However, the authors of the article demonstrated that a design based solely on attention processes, rather than RNN, can enhance efficiency [44].

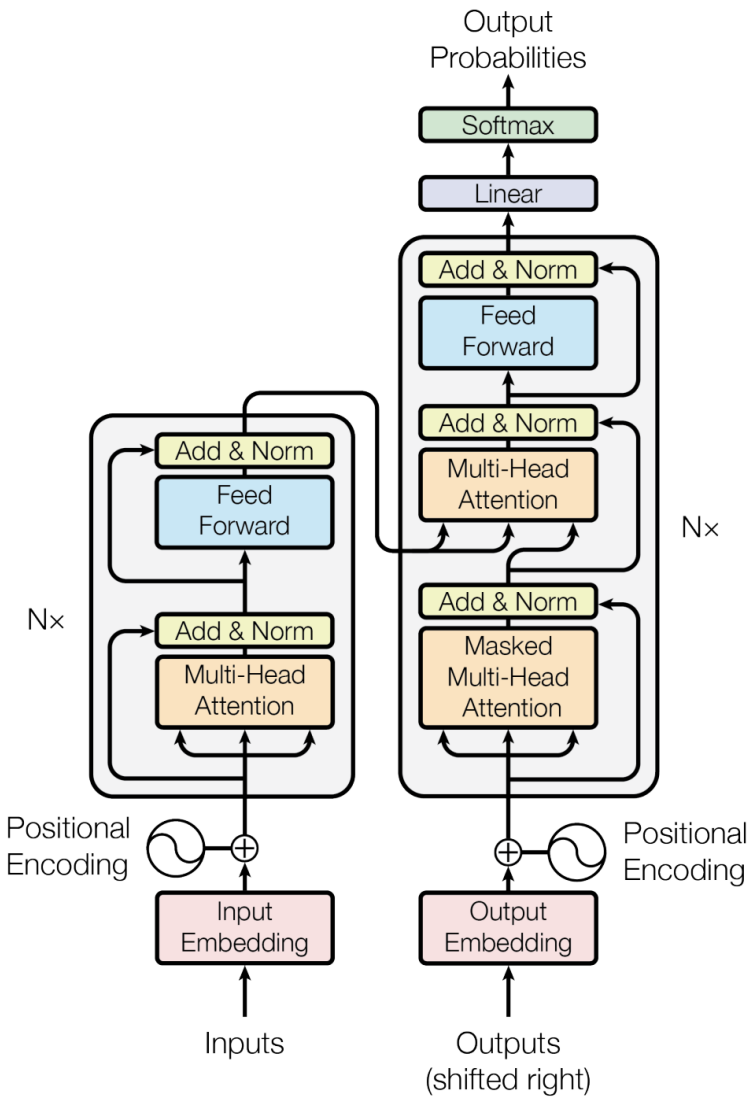


Figure 8: The transformer model architecture, taken from [43]

In Figure 8, the Encoder is shown on the left and the Decoder is on the right. Encoder and Decoder are both made up of modules that can be layered on top of each other several times, referred to as Nx. Multi-Head Attention and Feed Forward levels are frequently layers in these modules. First, the inputs and outputs (target sentences) are placed in an n-dimensional space. Positional encoding of a word is done to capture the global context. Because there are no recurrent networks that can memorize how sequences are fed into a model, every word/part in the sequence must be given a relative position, as a sequence is determined

by the order of its elements. These coordinates are added to each word's embedding into a n-dimensional vector [44]. The multi head attention mechanism is shown in Figure 9.

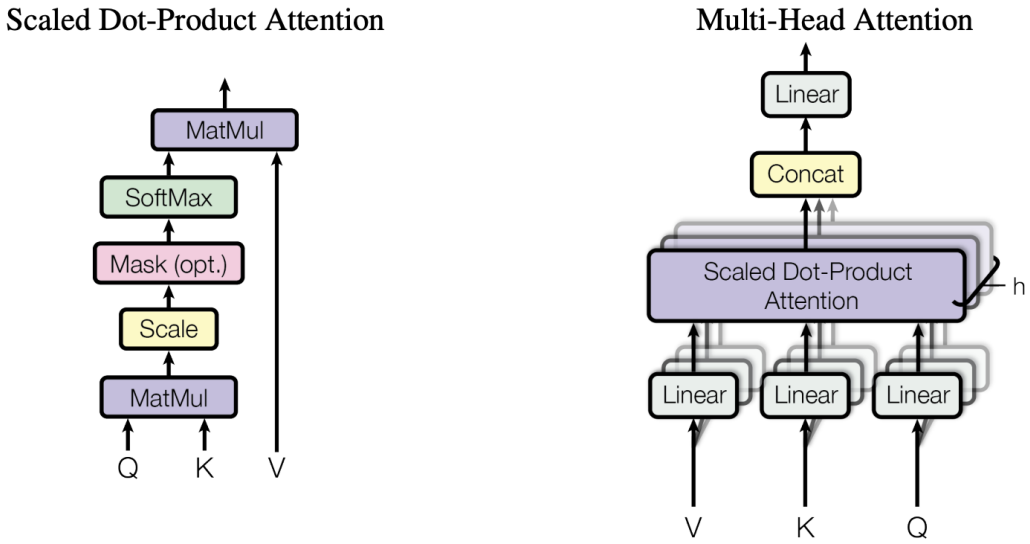


Figure 9: Scaled dot product attention (left) and Multi head attention (right), taken from [43]

The attention mechanism in the left module (with Q, K, V inputs) can be described by the following equation [43]:

$$Attention(Q, K, V) = softmax(QK^T/\sqrt{d_k})V$$

Q is a matrix comprising the query which represents each word in the sentence, K all the keys which represent all the words in the input text, and V all the values which represent all the words in the input text. V would be the same word pattern as Q for the encoder and decoder, multi-head attention modules. V, on the other hand, differs from the sequence represented by Q in the attention module, which takes into consideration the encoder and decoder sequences[44]. To put it another way, the values in V are multiplied and added with certain attention-weights a, where the weights are specified as follows [43]:

$$a = softmax(QK^T/\sqrt{d_k})$$

The weights a are determined by the impact of each word in the sequence(Q) on the other words in the sequence(K). The SoftMax function is also applied to the weights a, resulting in a distribution between 0 and 1. After that, the weights are applied to all of the words in the sequence that are introduced in V [44]. Figure 9 on the right shows the way this attention-mechanism is parallelized. The attention mechanism is performed numerous times with linear projections of Q, K, and V. This enables the system to benefit from a variety of Q, K, and V representations. These linear representations are calculated by multiplying Q, K, and V by the weight matrices W learned during the training [44]. Those matrices Q, K, and V differ depending on whether the attention modules are in the entire encoder input sequence or a piece of the decoder input sequence is the focus of attention. A multi-head attention module

connects the encoder and decoder, ensuring that the encoder input sequence is taken into account alongside the decoder input sequence up to a particular point. The multi-attention heads are followed by a pointwise feed-forward layer [44]. A pointwise feed-forward layer follows the multi-attention heads. This small feed-forward network has same parameters at each position, and every element from the sequence may be characterized as an independent, identical linear transformation [43]. The vectors Q, K, V are defined in the sense that which feature will be treated as Q, and which ones as K and V depend on a given task.

3.5 BERT

BERT [26] is bidirectionally trained stacked encoders of a Transformer [43]. Rather than anticipating the next word in a sequence, BERT employs a technique known as Masked LM (MLM), which involves masking words in a sentence at random and then attempting to predict them. To anticipate the masked word, the model looks in both directions and uses the entire context of the sentence, considering both the left and right surroundings. Furthermore, unlike earlier language models, it considers both the preceding and next tokens at the same time. The previous and next tokens were not really considered simultaneously in the existing combined left-to-right and right-to-left LSTM-based models [45]. BERT is based on the Transformer architecture, which learns contextual links among words in a text by using attention mechanism. A basic Transformer comprises of a text input encoder and a task prediction decoder. BERT, on the other hand requires the encoder part because its objective is to construct a language representation model. A sequence of tokens is fed into the BERT encoder, which is subsequently transformed into vectors and processed by the neural network. However, before BERT could begin processing, the input must be processed [45].

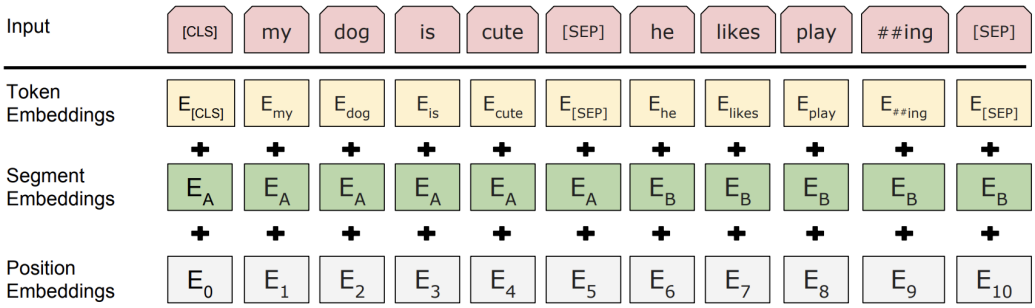


Figure 10: BERT Input Embedding, taken from [26]

All the steps for generating the input as shown in figure 10 are as follows:
Token embeddings: To mark the beginning of the first sentence, [CLS] token is added before the actual starting token and a [SEP] token is inserted at the end of every sentence [26].
Segment embeddings: Each token has a flag indicating whether it is Sentence A or Sentence B. This enables the encoder to make meaningful distinctions between sentences [26].
Positional embeddings: Each token is given a positional embedding to reflect its position inside the sentence [26].
BERT training is done on the following to tasks:

1. Masked LM (MLM):
This is a simple masking task where 15% of the words in the input are masked with a [MASK] token randomly, then the entire sequence is run through the BERT attention based encoder and just the masked words are predicted based on the context provided by the other non-masked words in the sequence. This basic masking strategy, however, has a flaw: the model only tries to predict the correct tokens when the [MASK] token is present in the input, but it is expected the model should try to predict the proper tokens regardless of the token present in the input. To address this issue, 80% of the tokens chosen for masking are actually substituted with the token [MASK]. In 10% of the cases, tokens are replaced at random with other tokens. In the remaining 10% cases, tokens are unattended. During training, the BERT loss function only considers masked token predictions and ignores non-masked token predictions. This results in a model that converges much more slowly than bidirectional models [45].
2. Next Sentence Prediction (NSP):
The BERT training procedure also leverages next sentence prediction to study the relationship between two sentences. For applications like question answering, a pre-trained model with this depth of understanding is useful. During training, the model is given pairs of sentences as input and is taught to anticipate if the second sentence is the same as the next sentence in the original text [45]. BERT uses a special [SEP] token to separate sentences. During training, two input sentences are fed at the same time, with the second sentence appearing 50% of the time after the first. In the other 50% of the time, a random sentence from the entire data. BERT must then predict whether the second sentence is random or not, assuming that the random sentence will be unrelated to the previous sentence [45]. Figure 11 shows an illustration of the same.

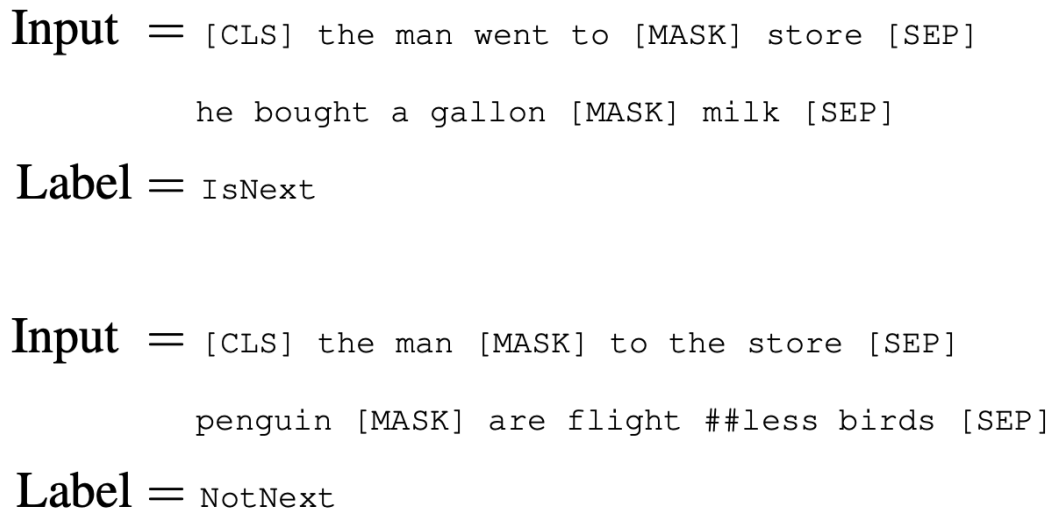


Figure 11: Next Sentence Prediction Task, taken from [45]

Both Masked LM and Next Sentence Prediction are used to train the model. The goal is to reduce the combined loss function of the two techniques [45]. There are different pre-trained BERT versions of different dimensions available, which can be fine tuned for a given task.

3.6 Comprehensive Study of SOTA Models

This section provides a more in-depth analysis of existing models that are closely related to the proposed model.

3.6.1 ConceptBert: Concept-Aware Representation for Visual Question Answering [4]

In Figure 12, the ConceptBert Model is depicted. This model decreases computing costs by eliminating the need for additional knowledge annotations or search queries. The model's input consists of three modules: an image representation, a question representation, and a knowledge graph representation module, all of which are explained below [4].

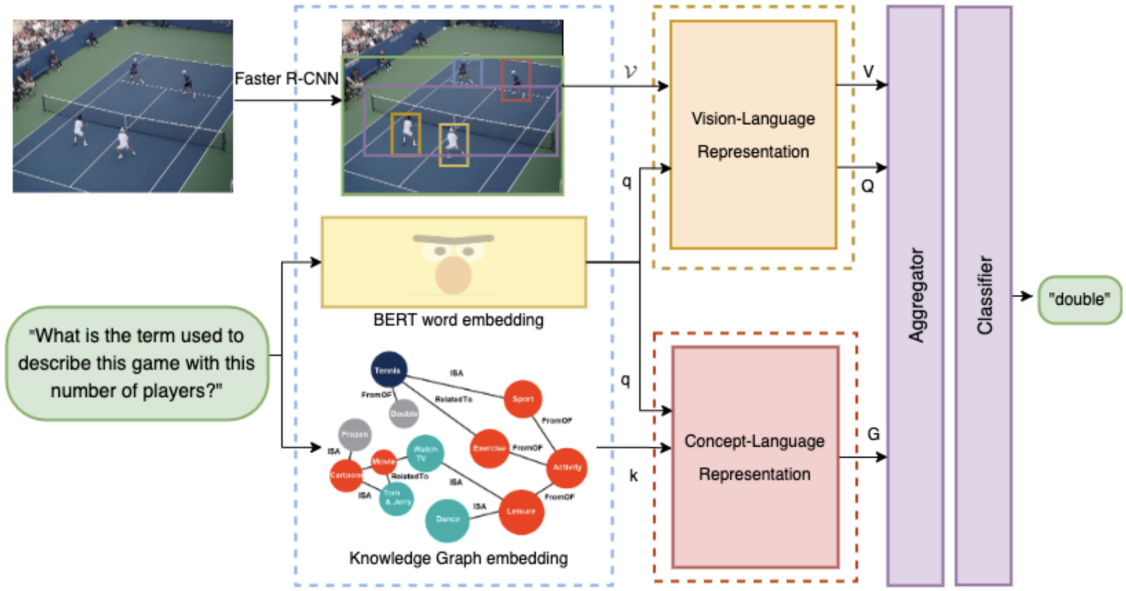


Figure 12: Model Architecture for ConceptBert, taken from [4]

Image representation:
A set of objects $V = \{vi|i = 1, ..., nv\}$ is extracted using pre-trained Faster R-CNN [46] per image, where each object vi is accompanied by a visual feature vector $vi \in R^{d_v}$ and bounding-box coordinates $bi \in R^{d_b}$ [4].

Question representation:
Given a question consisting of n_T tokens, BERT embeddings [47] is used to generate question representation $q \in R^{n_T \times d_q}$. BERT works with discrete token sequences made up of vocabulary words and special tokens, such as SEP, CLS, and MASK. Each token's representation is made up of a token-specific learnt embedding as well as position and segment encodings. Position denotes the token's position in the sequence, whereas segment denotes the token's sentence index if several sentences exist [4].

Knowledge graph representation:
They rely on the entire ConceptNet for common sense knowledge. ConceptNet is a multilingual knowledge base that represents the common sense relationships between words and phrases that people use. There are almost 21 million edges and over 8 million nodes in this

graph. They concentrate their efforts on the English vocabulary, which has over 1.5 million nodes. To skip the query construction stage and take full advantage of the KG's huge scale, they exploit ConceptNet embedding done in [15]. The KG is represented as $k \in R^{n_T \times d_k}$. To incorporate information from a node's local neighborhood, this method employs Graph Convolutional Networks [5] [4].

Vision-Language representation:

The module that produces vision-attended language characteristics V and language-attended visual features Q is shown in Figure 13. VILBERT [21] was the inspiration for this module. It's built on two parallel BERT-style streams that work on image regions and text segments. To enable information interchange across image and text modalities, each stream is made up of a series of transformer blocks and co-attentional transformer layers [4].

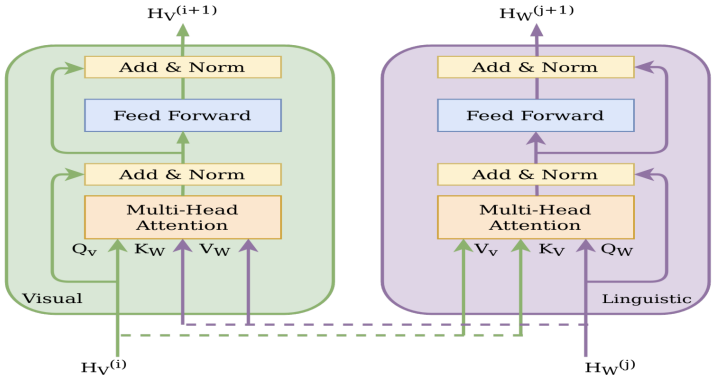


Figure 13: Vision Language Module, taken from [4]

Concept-Language representation:

The concept-language generation module is depicted in Figure 14. They accomplish this by employing an attention transformer layer, which is a multi-layer bidirectional Transformer that makes use of the encoder component of the original Transformer [43]. The model is able to combine common sense knowledge into the query and enhance its perception using the information included in the knowledge graph [4].

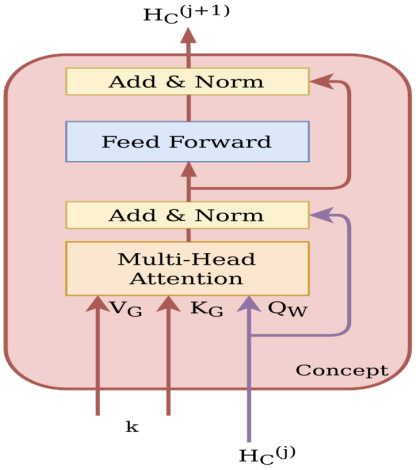


Figure 14: Concept language Module, taken from [4]

Concept-Vision-Language embedding and Final Classifier:
To produce a meaningful result, the aggregator must recognize high-level connections between the three streams without erasing the lower-level interactions retrieved in the preceding steps. They create the aggregator by applying the Compact Trilinear Interaction [22] to question, knowledge, and image features and generating a vector to represent the three features together. Finally, the answer is predicted using a classifier [4].

3.6.2 Zero-shot Visual Question Answering using Knowledge Graph [11]

Zero-shot VQA model is outlined in Figure 15. Chen et al. [11] propose a zero-shot VQA algorithm using knowledge graphs and a mask-based learning mechanism for getting better external knowledge representation. It presents new answer-based zero-shot VQA splits for the F-VQA dataset. Experiments show that this method can perform in Zero-shot VQA with unseen answers [11]. Firstly, three distinct feature mapping spaces is learnt independently: semantic space for relations, object space for supporting entities, and knowledge space for answers. Each of them is utilized to align the image question pair's (I-Q pair) joint embedding with the relevant target. By combining all of the selected supporting entities and relations, masks are determined according to a mapping table that contains all triplets in a fact KG and serves as a guide for the unseen response prediction alignment process [11].

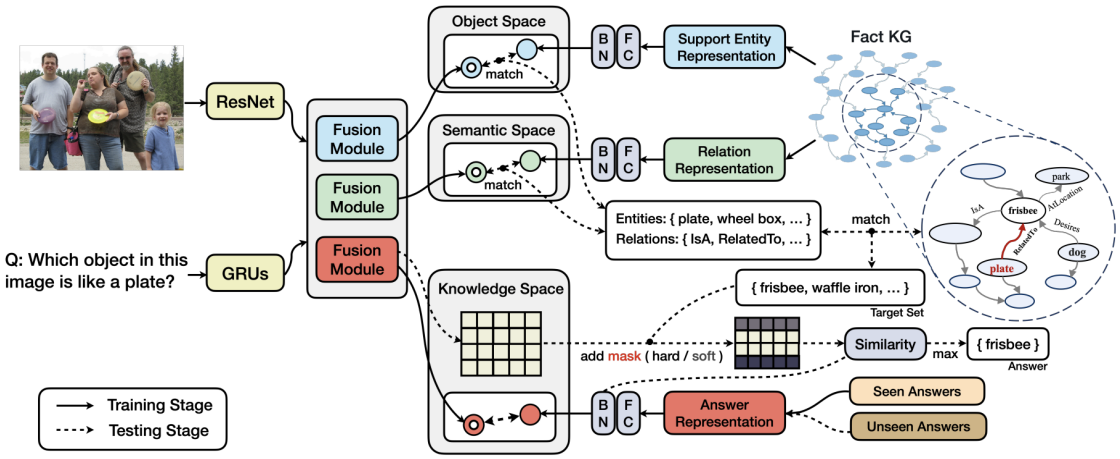


Figure 15: Zero-shot VQA model, taken from [11]

ZS-VQA Architecture Overview:
This model converts VQA from a classification task into a mapping task utilizing semantics embedding features as answer representation. After parameter learning, the joint embedding distribution of the question and the image can partly get close to the embedding of the answer, which is denoted as the knowledge space about answers. Besides, the authors define two other feature spaces independent of the knowledge space: semantic space representing relations and object space representing support entities (a.k.a. entities on KG). The semantic space projects (i, q) joint features into relation (r) as per the semantic information in triplets,

while object space establishes the relevant connection between (image(i), question(q)) pair, and a support entity. Together, they play the role of answer-guidance [11].

Establishment of Multiple Feature Spaces:
The authors then establish a connection between an answer and its corresponding (image(i), question(q)) pair by projecting them into a common feature space and get close to each other. Between the (q, i) pair a fusion feature extractor, $F_\theta(i, q)$, is leveraged to combine multimodal information, and $G_\phi(a)$ is defined as the representation of the answer, a. Semantic space works on the language information within (i, q) pair, which guides triplet-relation r's projection in the KG. The object space acts as a feature space of the support entity classifier, which observes images and texts simultaneously for salient features. Specifically, the alignment of (i, q) joint embedding and the support entity embedding circumvent the direct learning of complex knowledge and at the same time acts on the subsequent answer mask process along with the prediction relations, r, obtained from the semantic space [11].

Answer Mask via Knowledge:
Answer Masking is done to improve the machine's understanding of the text. A novel VQA answer masking strategy is proposed here. With the learned semantic and object space embedding, a disjoint fusion embedding in two independent feature spaces is obtained. These are respectively taken as the basis for subsequent entity and relation matching. A candidate set is generated for relation and supporting objects using the vector similarity approach. Finally, a target set is collected, which contributes to the masking strategy on all answers. A score is given, which generates a hard mask or soft mask on the target and hence controls the candidate target set [11].

3.6.3 Straight to the Facts: Learning Knowledge Base Retrieval for Factual Visual Question Answering [5]

A VQA system is depicted in Figure 16. The graphic shows the pipelined architecture used to generate SOTA outcomes on the FVQA dataset. Given an image and a question about the image, they present a method for obtaining a joint Image-Question Embedding by applying a CNN to the image, an LSTM [48] to the question, and an MLP to combine the two modalities. They employ another LSTM to predict the fact relation type from the question in order to filter relevant facts from the Knowledge Base (KB). GloVe embeddings are used to encode the recovered structured facts. The recovered facts are ordered using the dot product of their embedding vectors (of image, visual concept and question), and the fact. The fact getting the highest rank is returned to answer the question. Subsequent sections will discuss the components individually [5].

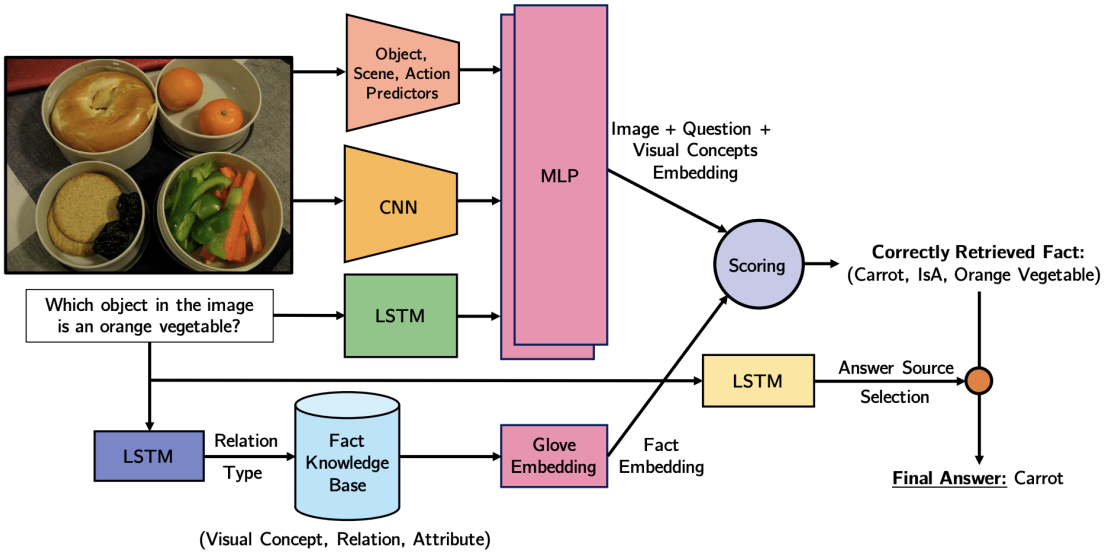


Figure 16: Factual VQA Model, taken from [5]

Scoring Facts:
Figure 16 illustrates the approach to score the facts in the knowledge base, represented as $S(g^F(fi), g^{NN}(x, Q))$. The score is calculated in the following steps: (1) computing of a fact representation $g^F(fi)$; (2) computing of an image-question representation $g^{NN}(x, Q)$ with question, image and visual concept embeddings; (3) combining the fact and image question representation to obtain the ultimate score S [5].

Predicting the relation:
The predicted relation could be represented as $\hat{s} = h^r_{w_1}(Q)$, derived from the question Q. To sum it up, they embed and then encode the words in the question Q one by one, and then use a typical multinomial classification to linearly change the LSTM's [48] final hidden representation to predict $\hat{r}$, from |R| possibilities alternatives. The relation prediction parameters w_1 is trained separately from the score function [5].

Predicting the answer source:
The predicted answer source could be represented as $\hat{s} = h^s_{w_2}(Q)$, again derived from a given question Q. They use an LSTM [48] network for embedding and encoding the words of the question before passing it through a linear transformation layer to predict $\hat{s} \in \{Image, KnowledgeBase\}$. Similar to relation prediction, they train the parameters w_2 independently [5].

Learning:
The parameters for score function w, relation w_1 prediction, and answer source prediction w_2 are trained separately. They train w_1 using a dataset $D1 = \{(Q, r)\}$ that contains pairs of questions and the relation that was used to derive the response. To train w_1, they used the $D2 = \{(Q, s)\}$ dataset, which contains pairs of questions and their associated answer sources. They use a time interval t = $\{1, ..., T\}$ to train the scoring function's parameters. They gradually increase the difficulty of the dataset D(t) by mining hard negatives at each time step. To be more precise, for each question Q and image x, D(0) comprises the 'groundtruth' fact f^* in addition to 99 randomly picked 'non-groundtruth' facts. After training the scoring function on D(0), they use it to forecast facts for image-question combinations and construct

a new dataset D(1) that now comprises, in addition to the groundtruth fact, another 99 non-groundtruth facts assigned a high score by the score function [5].

3.6.4 COMET : Commonsense Transformers for Automatic Knowledge Graph Construction [8]

The purpose of the COMET model is to produce an adaptation framework for the construction of knowledge bases with common sense, i.e. the model is capable of producing basic logical inference from a given set of tuples consisting of entities and the causal relationships between any two entities. To put it simply, the model is expected to complete a triple by producing object given a subject and relation. During training these tuples provide COMET with the KB structure and relations that must be learned, and COMET learns to add novel nodes and edges to the seed knowledge graph [8].

While COMET is independent of the language model with which it is initialized,the transformer language model architecture introduced in GPT [24] to encode input text is used here. This architecture utilizes multiple transformer blocks of multi-headed scaled dot product attention and fully connected layers [8].

The architecture of COMET is shown in Figure 17. The key, value, and query all travel via a head-specific projection in the multi-headed attention module depicted in (a) before a scaled dot-product attention is generated between them. The heads' outputs are concatenated and projected further [8].

The outputs of all prior layer blocks from earlier time steps are input to the multi-headed attention inside the transformer block illustrated in (b), with the preceding block for the current time step as the query [8].

Each token, along with all previous tokens, is send an input to a first-layer block, as shown in (c). Dotted lines denote inputs to all prior blocks in the previous layer and outputs to all subsequent blocks in the next layer. [8].

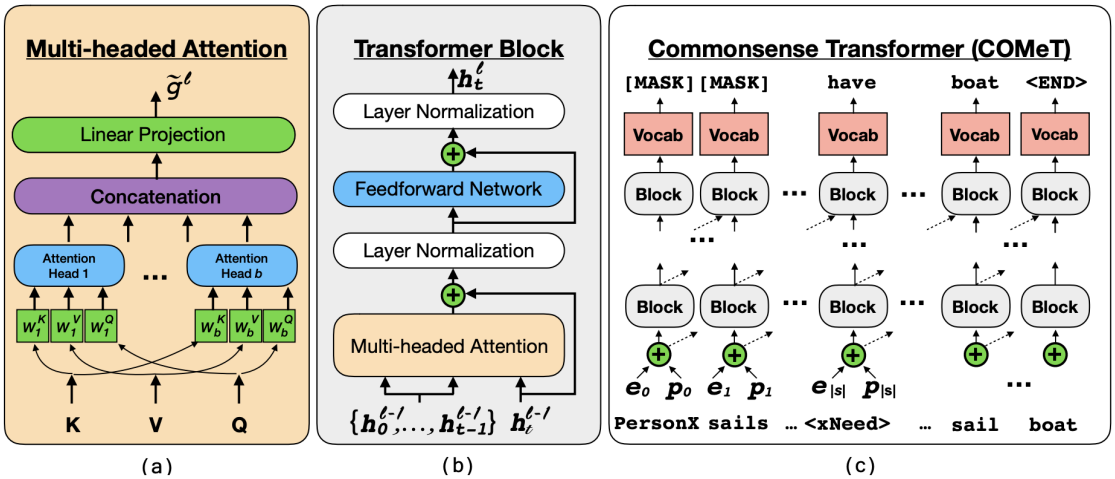


Figure 17: COMET Architecture, taken from [8]

4 Approach

The recent approaches that are used to integrate common sense knowledge are reviewed. In [7] the authors tried to formulate a SPARQL query using LSTM [48], visual concepts and answer source extracted from the image and question, then finally retrieve data from a RDF store containing all the knowledge triples. In [5] a model is used to score triples which are extracted from original knowledge base based on relation type predicted by another model. In [11] a non classifier based approach is used and it successfully tackles the drawback of pipeline architecture adopted in the previous two mentioned approaches.

However, none of the above approaches consider a situation where the ground truth knowledge may not necessarily be present in the original knowledge base. In this case, the model will not able to answer the question since the essential knowledge will never be short-listed from the knowledge base since it is not present there. Also, in a more generalised real world setting, a certain overview is vital to answer a question that could be possibly mapped to multiple triples in a knowledge base instead of one. This means, not a single but multiple triples may be required to decipher the answer of a question. In this case the question to ground truth triple mapping approach is narrow.

Keeping in mind the recent advances in automatic knowledge graph construction research in Malaviya et al. [15] and Bosselut et al. [8], a plausible approach to mitigate the drawbacks encountered in previous approaches is to integrate novel triples via automatic knowledge graph construction model COMET [8]. A state of the art architecture is proposed which uses visual concepts derived from the image and relations predicted from the question to trigger COMET. This in turn delivers diverse, novel knowledge of high quality that boosts the knowledge supply. To summarize, the approach considers using triples produced by COMET [8] instead of solely filtering triples from a existing knowledge base or using the entire finite knowledge base like Gardère et al. [4]. In the end of the pipeline, two attention based models is proposed which can be used interchangeably to predict the final answer. Several modules are required to have a fully functional architecture including relation predictor model, visual concept predictor, scene classifier, action classifier, object classifier to determine the visual concepts, COMET [8] module to produce complete triples and the final module which produce the output when provided with image, question and knowledge features.Different experiments related to embedding techniques adopted for the knowledge triples is conducted, the integrity and stability of the pipeline is evaluated by addition and removal of different modules. After the final evaluation, the best model obtained so far is compared with other SOTA approach.

5 Problem Formulation

Let $q \in Q$ be a question, which can be answered by looking at an image $I \in I$ and a knowledge graph G where G represents a knowledge graph constructed using set of triples generated by an automatic knowledge graph construction model guided by the visual concepts from the image. The goal is to predict a meaningful answer $a \in A$. Let Θ be the parameters of the model p that needs to be trained [4]. Therefore, the predicted answer $\hat{a}$ of the model is [4]:

$$\hat{a} = arg\ max(p_\Theta(a|I, q, G))for\ a \in A$$

In order to retrieve the correct answer, a joint representation $z \in R^{d_z}$ of q, I, and G is learned such that [4]:

$$a^* = \hat{a} = arg\ max(p_\Theta(a|z))for\ a \in A$$

where a^* is the ground-truth answer. d_z is the dimension of the joint embedded space of z [4].

6 Methodology

6.1 Dataset and Metrics

FVQA (Fact-based VQA) [7], a knowledge-based VQA dataset is used in this thesis to support the hypothesis. FVQA is mostly comprised of questions that require external data to be answered. A conventional VQA dataset is expanded, which contains image-question-answer triples, through additional image-question-answer-supporting fact tuples. The supporting-fact is expressed structurally as a triple, such as <Chair,HasA,fourlegs>. FVQA focus on collecting visual questions which need to be answered with the assistance of supporting-facts. To this end, a specialized system is designed, in which the procedure of procuring the dataset is conducted in the following steps [7]:

1) Picking Concept:
Using FasterRCNN [46], 326 separate object classes are extracted from the image, 221 distinct scene classes are obtained using VGG-16 model trained on MIT Places 205-class dataset [49], and 24 distinct human or animal action classes are obtained using [50]. These visual concepts are also linked to a number of other external knowledge sets. An image and a set of visual concepts are presented to annotators (object, scene, and action). They must select one of the visual concepts associated with this image [7].

2) Choosing Fact:
When a visual concept is chosen, the facts that with it are displayed in the form of phrases, with the two entities, subject and object of the triple highlighted. The fact (apple, IsA, piece of fruit) is, for example, represented as 'Apple is piece of fruit.' Annotators should choose a correct fact on their own [7].

3) Create Question and Answer pair:
The Annotators must pose a question that requires information from both the image and the selected fact. Thus, the chosen fact becomes a supporting fact for the posed question. The response is constrained by the two entities contained in the supporting-fact. In other words, the answer can be derived from either the visual concept embedded in the questioned image or the KB [7].

6.1.1 Dataset Selection Criteria

The target of the proposed approach is to integrate common sense knowledge into the VQA system to boost common sense understanding of the system. From a universal perspective, an automatic knowledge base completion model COMET generates relevant knowledge

triples given an image and question. In this setup, ideally, a dataset like OK-VQA [14] with open knowledge requirement should be used to testify the approach; however, OK-VQA deals with general knowledge questions and does not ideally fall into the common sense set. So, FVQA[7] is used here, which deals in and out with common sense knowledge. Although this dataset requires a given Knowledge Base, precisely a particular triple from a knowledge base to answer a question at hand, nevertheless it would be interesting to find out if COMET generated knowledge triples that may not contain the ground truth target triple required to answer the question be able to produce comparable results. To sum up, this dataset would be challenging enough to deliver good results as far as the proposed approach is concerned and therefore rightly validate its effectiveness.

6.1.2 Evaluation Metrics

The performance is measured using accuracy [51]; accuracy@1, accuracy@3, accuracy@10 are observed in each experiment to judge the all-around performance of model. Accuracy@k indicates that the correct answer ranks in the top-k predicted answer sorted by probability. The same metric is used in all the papers which is used for comparison later.

6.2 Modality Representation

6.2.1 Knowledge Graph embedding

BERT Embeddings of Graph Nodes: Transfer learning from language to knowledge graphs has been found to be beneficial for the building of commonsense knowledge graphs [8]. To complete the translation from language to knowledge graphs, BERT is finetuned [26] using the masked language modeling loss. The semantic node representations are derived depending on the text phrase. This enables BERT to be customized to a KG's unique writing style. The set of distinct phrases used to represent nodes in the KG is employed as the input for finetuning. The input to the model is represented as [CLS] + $\overline{e^i}$ + [SEP], where $\overline{e^i}$ is the phrase of a node. The [CLS] token from the last layer of the BERT model is used as node representations. The node embedding matrix obtained from the BERT model as $T \in R^{|N| \times M}$, where M is the dimension of BERT embeddings [15].

Graph Convolution Network (GCN) embedding [25]: are efficient at capturing information about a node's local neighborhood. A graph convolutional encoder takes a graph G as input and embeds each node in a D-dimensional space, $h_i \in R^D$ for each nodes $e_i \in N$. For a given triple, there are two nodes connected by a relation between them. So if there are N nodes $N/2$ triples are used to build the knowledge graph [15]. The GCN encoder works by transmitting messages from a node to its surrounding nodes, typically weighted by the edge's relation type [15]. This procedure is performed in layers, integrating information located numerous hops distant from a node. The representation of the final layer is used as the node's graph embedding. A variant of the GCN that allows to parameterize the edge's relation type and account for the relevance of a node's neighbor during aggregation is chosen. For a graph G with R relation type and a GCN with L layers, the equation for representing

the embeddings of a node e_i in $(l+1)^{th}$ layer [15] is:

$$h_i^{l+1} = tanh(\sum_{r \in R} \sum_{j \in J_i} \alpha_r \beta_{ij}^l W^l h_j^l + W_0^l h_i^l)$$

where J_i represents the surrounding nodes for a given node e_i in the graph, and W^l is a projection matrix for layer l. The initial node representation h_i^0 is calculated using an embedding layer. The second term in the above equation represents the self-connection for the node, and is used to propagate information from one layer to the next [15]. α_r is the weight for the relation type of the edge and β_i^l is a vector denoting the relative importance of each of e_i's neighbors [15]:

$$\beta_i^l = softmax(\hat{\beta}_i^l)$$

where each element of $\hat{\beta}_i^l$ is computed as [15],

$$\hat{\beta}_{ij}^l = h_i^l h_j^l$$

h_i^l and h_j^l are the representations of node e_i and its neighbor e_j . The final output of the GCN is a node embedding matrix $H \in R^{|N| \times D}$, where D is the dimension of GCN embeddings [15].The entire graph embeddings module is shown in Figure 18

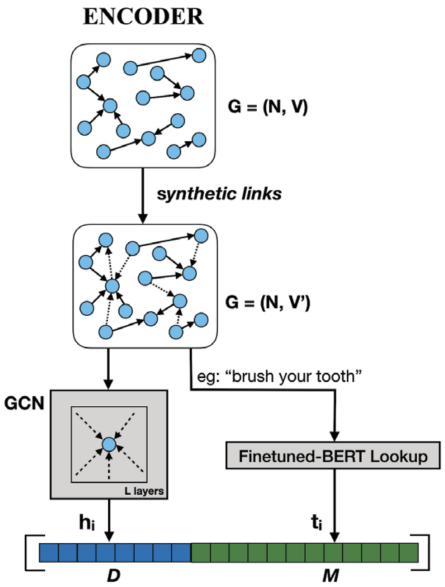


Figure 18: GCN plus BERT embeddings, taken from [15]

Glove Embeddings: GloVe [52] is a word vector representation approach.Word vectors translate words to a fixed dimension vector space, where homogeneous words cluster together and contrasting words diverge. The advantage of GloVe is that, unlike Word2vec [53], GloVe does not only depend on local contextual information of words, but include global information [54].
GloVe is a weighted least-squares model that is essentially a log-bilinear model. The model's central concept is based on the notion that ratios of word-word co-occurrence probabilities

are capable of conveying some type of meaning. As one might assume, ice occurs more frequently in conjunction with solids than with gases, whereas steam occurs more frequently in conjunction with gases than with solids. Both terms usually appear in conjunction with their shared property water, while both terms occur infrequently in conjunction with the unrelated term fashion.Noise from non-discriminatory words such as water and fashion cancels out only in the ratio of probability.For example, water and fashion cancel out, such that large values (significantly greater than 1) correlate well with ice-specific properties, while small values (significantly less than 1) correlate well with steam-specific properties [52].
In this work, glove.840B.300d.txt [52] is used with 2.2M vocab, 840B tokens, cased, 300 dimension vectors. This means for each token, $D = 300$ dimensional vector is generated. For each data point, K triples are selected from COMET which might be helpful to answer the question. Recursively, for every triple, tokens are extracted from the subject and object. Finally, all triples are processed to have an equal length of |N| tokens.The output of each knowledge embedding using glove is $G \in R^{K \times |N| \times D}$ [52].

6.2.2 Image Feature Embedding

To get image embeddings, visual features is extracted from the layer 4 output of ResNet-152 (14×14×2048 tensor) [55] pre-trained on ImageNet [56]. The Deep Residual Network is quite similar to conventional networks in that it employs convolution, pooling, activation, and fully linked layers arranged on top of one another. The only construction made to the simple network to make it a residual network is the identity connection between the layers. It has surpassed the top5 accuracy of AlexNet [57], VGGNet [58] and Inception [59], at the cost of computational complexity.

6.2.3 Question Embedding

Given a question consisting of N tokens, a representation $Q \in R^{|N| \times D}$ is generated for the Transformer based Model, discussed in section 6.4.7. Each token is represented by a combination of a token-specific learned embedding and position and segment encodings. Position denotes the token's index in the sequence, whereas segment denotes the token's sentence index if several sentences exist.
Again, for Stacked attention based network, discussed in 6.4.8, glove embeddings are used to represent the questions. For a question consisting of N tokens, a $D = 300$ dimensional vector is generated for each token. This means output for each question embedding is $Q \in R^{|N| \times D}$

6.3 Architecture

The proposed architecture is shown in Figure 19. To understand the Architecture in depth, it is necessary to revise the problem again. Given a question and image, the model should be able to answer it using some external common sense knowledge which is in the form of some set of triples. This triple has three parts, a subject, an object and a connection between them namely relation.
At first the question is fed into a relation prediction BERT Classifier [60] which predicts the relation of the triples that could probably be useful in answering the question.
If one follows the figure 19, the question reads "Which object in the image has four legs?",

the relation of triples which could possibly answer it, would be $HasA$. In this case we need a triple which could probably be like $---(fill\ in\ with\ the\ object\ in\ the\ image)\ HasA\ four\ legs$. So the first step is to find this "relation".

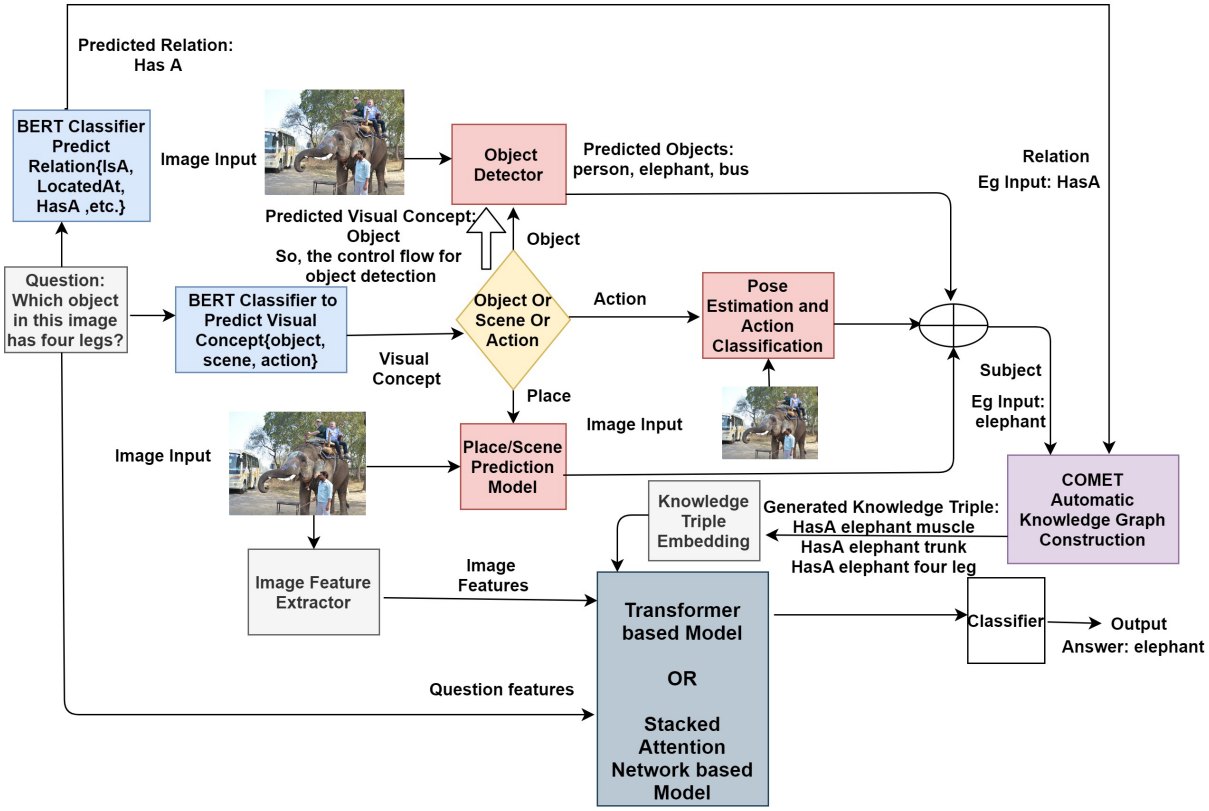


Figure 19: Proposed Architecture

From the perspective of the visual concept embedded in the image, the question could possibly be related to any object present in the image or the scene depicted or any action going on inside the image. This leads to the requirement of another BERT Classifier, which uses the question to classify the visual concept of the image, i.e *object*, *action* or *scene*. Accordingly based on the prediction, program flow follows the respective object or action or scene classifier model. In case of the sample question, "Which object in this image has four legs?" as shown in figure 19, the required visual concept to answer it is the *object*. Proceeding towards the object detector, *elephant* is predicted. This is used to fill in the missing subject position which needs to be fed into the COMET.
The predicted object or scene or action forms the subject of the knowledge triple, it is fed into the COMET [8] along with the predicted relation to generate complete, novel, diverse triples which is expected to produce the knowledge required to answer the question. Again, from the example, "*elephant HasA*" is fed into the COMET which produces triples like "*elephant HasA trunk*", "*elephant HasA four legs*", "*elephant HasA muscle*". Top 15 triples generated by COMET are further processed for generating the embeddings. Finally, the question features, image features and knowledge triples embeddings are fed into either the Transformer based model or Stacked attention based model. One or more of the generated knowledge triples could possibly be used for reasoning by the final transformer based model or stacked attention based model. These models tries to learn how to decipher the answer from

the given features through training. Both these models have produced stable and comparable results with other state of the art approaches. The details of these two models is described in the upcoming section 6.4. All the different modules discussed in the architecture are trained separately.

6.4 Module Description

6.4.1 Relation Prediction model

In this module, a pretrained BERT text classifier is used to predict the relation from a given question. There are 11 possible relation classes which can be predicted. These are present in conceptnet[1] KB and shortlisted by Wang et al. in FVQA [7]. COMET will not be able to recognize any relation other than the ones supported by the conceptnet KB since it is pre-trained on Conceptnet. The relation classes are, $RelatedTo$, $AtLocation$, IsA, $CapableOf$, $UsedFor$, $Desires$, $HasProperty$, $HasA$, $PartOf$, $ReceivesAction$, $CreatedBy$. BERT pre-training fine tuning and its workings are described in section 3.6. The "bert-base-uncased" variant of BERT is employed in this instance, as it is the smaller model trained on lower-cased English text (with 12-layer, 768-hidden, 12-heads, 110M parameters).The model was Pretrained on English language using a masked language modeling (MLM) objective. This module fine tunes the pre-trained model for a text classification task. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model weights are used to initialize models for variety of tasks which is a classification task in this case. All parameters are fine-tuned according to the given task. [CLS] is a special symbol that is appended to the beginning of each input example, and [SEP] is a special separator token (for example, separating questions/answers).The Figure 20 below provides a precise overview of the same. This module selects "all" relations in case the model gives equal probability to more than one class.

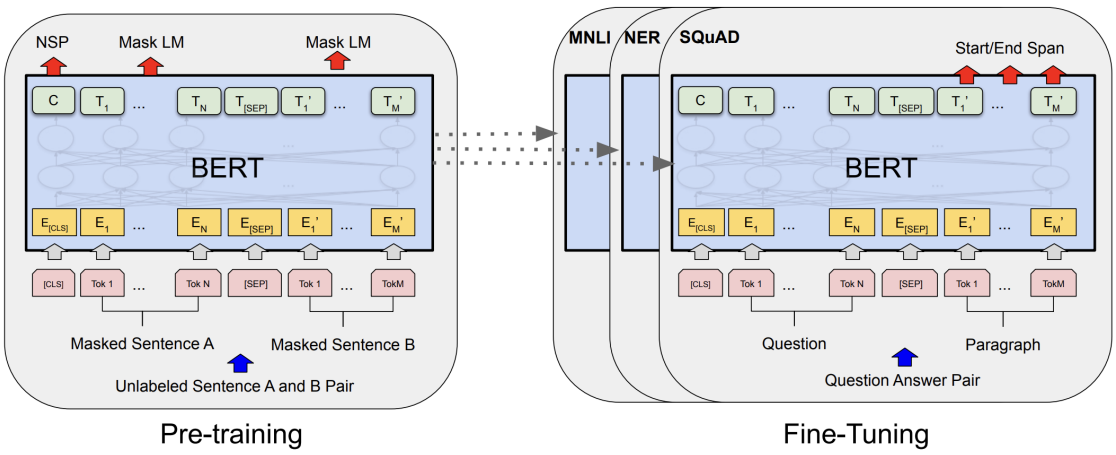


Figure 20: Pre-training and fine tuning of BERT, taken from [26]

6.4.2 Visual Concept prediction model

Similar to the previous module, this is also a classification task. A BERT classifier is fine tuned again to deligate the Classification task by modifying the last layer. Again the

bert-base-uncased variant of BERT is used with the modification of last layer to make it work as a classification model. Again this is a three class classification task. The generated output could either belong to *object* or *scene* or *action* class.

6.4.3 Object detection

If the predicted visual concept in the previous step is "object" then the program control flows into this module. In this module two state of the art object detectors are used to predict the objects present in the image. FVQA [7] dataset is built using MS COCO [23] and Imagenet [56] images. To detect objects from these images, Mask R-CNN [61] using [62] with 80 classes trained on MS COCO dataset and Darknet-53[63] with 1000-classes trained on Imagenet is used. For both the models, a threshold value of 0.7 is selected and any object predicted with a probability of more than 0.7 is short listed for the next step. For the given FVQA [7] dataset, object based questions i.e. questions which requires the knowledge of object present in the image to be answered correctly, comprises 90.66% of the total data points. Hence the output of these models is vital for the robust performance of this entire architecture.

6.4.4 Scene Prediction

If the predicted visual concept in the previous step is "scene" then the program control flows into this module. Say, for example the question is: "Which place in this image is used for cooking?", then possibly the scene shown in the image is that of a "kitchen" which should be predicted by this model. A VGG16_places365 [64] based model found in [65] pre-trained on Places365-Standard dataset [64] is used to predict scenes from the images. For the given FVQA [7] dataset, scene based questions i.e. questions which requires the knowledge of the scene in the image, to be answered correctly, comprises 8.77% of the total data points.

6.4.5 Action Prediction

Finally, If the predicted visual concept in the previous step is "action" then the program control flows into this module. Say, for example the question is: "Which action takes longer than the action shown in this image?", then probably the action shown in the image is needed to generate the suitable knowledge required to answer the question. This model generates a pose from a given image[66] considering the scene around the person, Inception-v3 architecture pretrained on the Stanford 40 dataset (9523 images) [67], which incorporates 40 different day-to-day activites is used for action prediction in this module. For the given FVQA [7] dataset, action based questions i.e. questions which requires the knowledge of action going on in the image to be answered correctly, comprises $< 1\%$ of the total data points.

6.4.6 COMET triple prediction and Knowledge Embedding

This module is responsible for generating the knowledge triple based on the inputs from the previous modules. The Relation Prediction BERT classifier predicts a relation from the question, either one of the three (object or scene or action) visual concept predictor module gets activated and predicts the respective object or scene or action from a given image. This constitutes namely the subject of the triple. There can be multiple objects predicted based on

the selected threshold, similarly there can be multiple scenes and actions based on the varying degree of precision for each prediction. Again, the relation predictor can predict more than one relation based on the given question (since all questions are not from Conceptnet and the model only handles relations from conceptnet). So for each relation-object/action/scene pair top 5 triples are generated by COMET. Now that, the subject and relation is at hand, it is time for COMET [8] to get into action. This automatic knowledge construction network generate complete triples based on given subject and relation. The Figure 19 shows sample output of COMET. The output of this is fed into an GCN plus BERT embedding module described in section 6.2.1 following [15]. The output of GCN and BERT is fed into the next model. For further experiments the triples are also embedded using glove embeddings as described in section 6.2.1.

6.4.7 Transformer based knowledge-Image-Question-Answer Model

Image features extracted using ResNet-152 [55] is represented with embedding $I \in R^{d_h \times d_b \times d_b}$. Knowledge features generated using GCN-BERT [15] duo is represented as $K \in R^{|N_k| \times |d_k + d_g|}$ where d_k is the dimension of GCN embedding per node discussed in 6.2.1, d_g is dimension of BERT embeddings per node discussed in 6.2.1, N_k is number of nodes (subject and object of triples). Question is represented as $Q \in R^{|N_t|}$ with $|N_t|$ tokens for each question. Now each of these embedding is fed into the model either as Query Q, Key K or Value V to produce different attention guided representation using [43]:

$$Attention(Q, K, V) = softmax(QK^T/\sqrt{d_k})V$$

Tokenized Questions, Q_W,K_W and V_W is fed into A(left) and B(right) as shown Figure 21. It is further embedded in each of the transformers using token-specific learned embedding and encodings for position and segment. Question features is further represented as $Q \in R^{|N_t| \times d_q}$, where d_q is the dimension of the token embedding as shown in Figure 10.

There are three transformer representation (this model does not include the masked multi-head attention in the decoder) in the model described in Figure 21. The left most transformer A generates Image conditioned question attention. The right one, B generates Knowledge conditioned question attention. Finally both these representation is fed into C for a fused Image, Question and Knowledge embedding. All the sub units A, B and C are cloned N_x times. In the original paper [43], this value is 6. However, N_x is modified to 3 for this task.

Encoder and Decoder of all the transformer calculates the Multi-Head Attention as described in 3.4. Q, K, V represent the query, key and value of the attention unit. In Figure 21 the suffix "W" represent that respective input is question embedding, "V" represent that respective input is Image embedding, and "G" represent that respective input is knowledge embedding, "VW" is image guided attention layer and "GW" is knowledge guided attention layer.

The final fused Image, question and knowledge representation (output of C) is passed through a linear and softmax layer for producing the ultimate output.

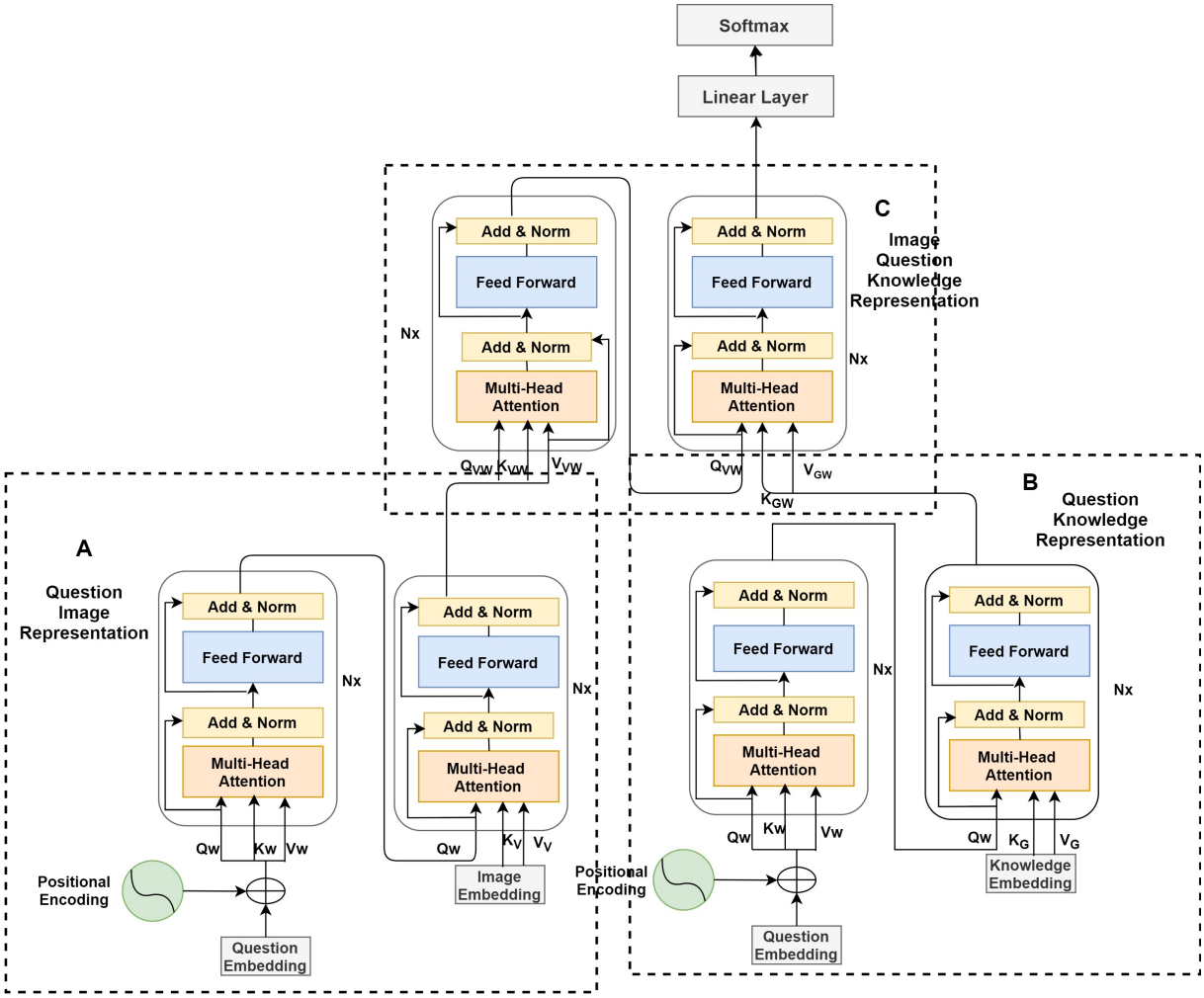


Figure 21: Transformer based Model

6.4.8 Stacked Attention Network based Model

Stacked Attention Network based model [9] can alternatively be replaced with the previous transformer based model for better results and stability. The features inserted in this section is same as described in the previous transformer based model in 6.4.7. The question embedding is however done using glove. A 300 dimensional vector is generated for every token in a question. The pretrained glove embeddings used here, is mentioned in section 6.2.1. In addition to the knowledge triple being embedded using the method described in 6.2.1, they were also embedded using glove, as discussed in 6.2.1. Both embedding methods are compared and the results were better when glove was used to generate the knowledge features (details in result section 8). As it is clear from Figure 22, the top half of the model generates a question guided image attention, the bottom half generates the question guided facts/knowledge attention. Finally the outputs of the top and bottom layer are concatenated and fed into the linear layer for classification.

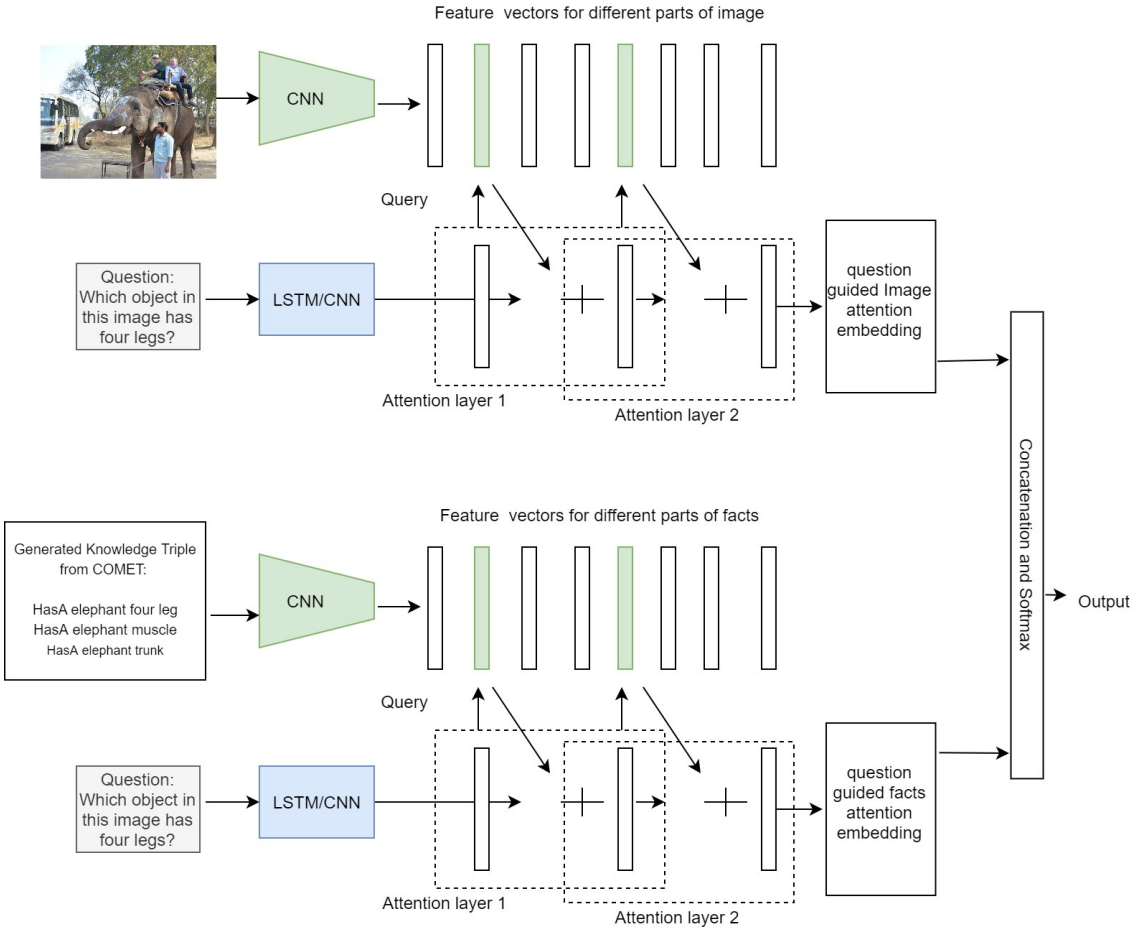


Figure 22: Stacked Attention Network based Model

7 Training Details

7.1 Relation Prediction model

The output from the last layer of "bert-base-uncased" is passed through a Dropout layer and then finally through a linear layer of size 11 which is equal to the number of classes for prediction. Adam Optimizer [68] with CrossEntropyLoss [69] was used for training. The details of all the hyper parameters used for training the Relation Prediction model described in section 6.4.1 is shown in Table 2

Hyper parameters	**Value**
Optimizer	Adam Optimizer
Learning Rate	2e-5
Loss	CrossEntropyLoss
Batch size	16
No. of Epochs	10

Table 2: Training details for Relation Prediction BERT Classifier

7.2 Visual Concept prediction model

The final output from the "bert-base-uncased" is passed through a Dropout layer and then finally through a linear layer of size 3 which is equal to the number of classes for prediction. Adam Optimizer [68] with CrossEntropyLoss [69] was used for training.The details of all the hyper parameters used for training the Visual Concept prediction model described in section 6.4.2 is shown in Table 3.

Hyper parameters	**Value**
Optimizer	Adam Optimizer
Learning Rate	2e-5
Loss	CrossEntropyLoss
Batch size	16
No. of Epochs	10

Table 3: Training details for Visual Concept Prediction BERT Classifier

7.3 Transformer based Model

The details of all the hyper parameters used for training the Transformer based model described in section 6.4.7 is shown in Table 4. A gradual fall in the learning rate of each parameter group by 0.5**(1 / 10000) is maintained in every epoch. Adam Optimizer [68] with negative log likelihood loss [70] was used for training. The number of encoder and decoder layer stacked in each transformer is 3. Since, the number of data points was less for training, in order to reduce the overfitting, the depth of the model was reduced to 3 as contrary to 6 as suggested in the original paper [43]

Hyper parameters	**Value**
Optimizer	Adam Optimizer
Learning Rate	1.5e-5(initial)
Loss	negative log likelihood loss
Batch size	32
No. of Epochs	50

Table 4: Training details for Transformer based Model

7.4 Stacked Attention Network based Model

The details of all the hyper parameters used for training the Stacked Attention Network based Model described in section 6.4.8 is shown in Table 5. Training is accomplished through the use of dropout and batch normalization. For the first seven epochs, a controlled learning rate warm-up is applied ($2.5 \times (\text{epoch} + 1) \times 5 \times 10{-}4$), and then it is reduced by 0.7 for every three epochs from epoch 14 to epoch 47, with no change in the value for the remaining epochs [11]. Meanwhile, the loss temperature τ is set to 0.01 and early stopping is employed in conjunction with patience set to 30 [11]. This means that the model will halt the training after waiting for 30 epochs when no further improvement is noticed in the performance metric

for the validation set data. Adam Optimizer [68] with negative log likelihood loss [70] was used for training.

Hyper parameters	**Value**
Optimizer	Adam Optimizer
Learning Rate	gradual learning rate
Loss	negative log likelihood loss
Batch size	128
No. of Epochs	800
patience	30

Table 5: Training details for Stacked Attention based Model

8 Results

8.1 Relation Prediction model

Table 6 shows the results of the relation prediction model from Section 6.4.1. As evident from the table the BERT classifier proposed in this method has surpassed the previous LSTM [71] based methods.

Method	**Accuracy**
FVQA[7]	64.94%
Straight to the Facts[5]	75.4%
BERT Classifier used in this pipeline	**76.35%**

Table 6: Performance of Relation Prediction Model

8.2 Visual Concept prediction model

Table 7 shows the results of the visual concept prediction model from Section 6.4.2. This module is particularly built to direct the program control flow to either an object, scene or action predictor. This addition is vital as it decides the flow and hence controls the subject (which can either be predicted object or scene or action from the image) being fed into COMET which in turn will impact the production of relevant triples. Accuracy, Precision, Recall are important metrics to judge the credibility of the model [72]. The ratio of correctly anticipated observations to total observations is a useful performance statistic called accuracy. The ratio of accurately predicted positive observations to all expected positive observations is the precision. The recall ratio is calculated by dividing the number of correctly predicted positive observations by the total number of observations in the class. Precision and Recall are averaged to get the F1 Score. As a result, both false positives and negatives are accounted for in this score [72].

Method	Accuracy	Average F1 Score	Average Precision	Average Recall
BERT Classifier used in this pipeline	**97.60%**	**97.47%**	**97.59%**	**97.60%**

Table 7: Performance of Visual Concept Prediction Model

8.3 COMET triple prediction

Table 8 shows performance of COMET [8] described in Section 6.4.6. The results show how precise COMET is in predicting the ground truth triple required to answer the question. As expected, the performance is much poorer in comparison to other state of the art approaches since comet is not entitled to generate the triple required to answer the question. On the other hand, other approaches directly filter triples from original knowledge base which contains the ground truth triple. COMET is a knowledge completion model, which produce complete triples (subject, relation, object), given a subject and relation.

Method	Accuracy
FVQA [7]	41.12%
Straight to the Facts [5]	**64.50%**
COMET [8] fact precision	20.92%

Table 8: Correct fact prediction precision

8.4 Transformer based Model

The accuracy of the Transformer based model from Section 6.4.7 for different modalities is shown in Table 9. All the three features, i.e. image, question and knowledge together outperforms the other two possible modality combinations.

To this end, some more experiments are conducted with varying knowledge generation technique. Since the data points related to scene and action is 8.77% and $< 1\%$ respectively, in order reduce the cascading error effect of pretrained scene and action prediction models, these are dropped from the pipeline for the time being and only object prediction model is used to detect objects present in the image. In this set up, visual concept prediction model, scene and action classifiers are removed from the pipeline. Only the objects generated from the object detectors are fed into COMET for further knowledge generation. This knowledge is referred as Knowledge_object in Table 10. The knowledge obtained from the regular flow described in the proposed architecture 6.3 (without any module elimination), is referred to as Knowledge_object_scene_action. Both these knowledge combination are used for training. This is done to test the consistency of the modules and how well it contributes towards upholding the stability of the pipeline based architecture.

Again to account for the error propagated through the pipeline due to wrongly predicted relation from the question, ground truth relation is fed into the COMET for both object and object/scene/action path set up. The produced knowledge by making the above changes is referred to as Knowledge_object_gt and Knowledge_object_scene_action_gt where gt

refers to the ground truth relation being fed to COMET instead of the output from the relation classifier. Finally all these combinations are compared to the output generated by the state of the art model [5] over FVQA dataset (marked in bold). The predicted scores from various combination of the knowledge in addition to the image and the question features are shown in Table 10. The best results obtained by the proposed approach is also marked in bold and knowledge obtained using the ground truth relation is not considered for comparison. It is also evident from the results that Knowledge_object setting produce the highest accuracy as far as the proposed method is considered.

Inspite of having a low ground truth fact precision of 20.92% compared to 64.50% as mentioned in Table 8, the accuracy obtained from the proposed approach, 54.69% is not too far from 62.20% in [5].It is also seen that knowledge generated with and without ground truth relation, does not have considerable impact on the accuracy.

Modality	**Accuracy@1**	**Accuracy@3**	**Accuracy@10**
Image + Question	42.12%	58.20%	68.00%
Knowledge + Question	44.60%	58.60%	68.80%
Image + Knowledge + Question	**54.69%**	**70.08%**	**78.79%**

Table 9: Performance of Transformer based model on different modalities

Different knowledge generation approaches	**Accuracy@1**	**Accuracy@3**	**Accuracy@10**
Image + Knowledge_object + Question(proposed approach)	**54.69%**	**70.08%**	**78.79%**
Image + Knowledge_object_scene_action + Question	54.49%	67.54%	78.24%
Image + Knowledge_object_gt + Question	55.55%	69.71%	79.72%
Image + Knowledge_object_scene_action_gt + Question	56.41%	69.80%	77.66%
Straight to the Facts-Question + Visual Concepts[5]	**62.20%**	**75.60%**	-

Table 10: Performance of Transformer based model on different knowledge combinations

8.5 Stacked Attention based Model

The accuracy of the Stacked attention based model from Section 6.4.8 for different modalities is shown in Table 11. All the three features, i.e. image, question and knowledge cumulatively outperforms the other two modality combinations i.e Image + Question and knowledge + Question.

Another experiment with diferent embeddings is performed. Knowledge + question modality is used for this experiment. The knowledge is embedded using glove described in Table 6.2.1 for one experiment and for the other setting it is embedded using GCN plus BERT as described in 6.2.1. As shown in Table 12 glove embedding perform much better than GCN plus BERT embeddings with inherent neighbouring node information in it.

Similar to the previous set of experiments conducted on the Transformer based model, different knowledge combinations is generated here as well. The results for different knowledge combinations is shown in Table 13. Knowledge_object_scene_action refers to knowledge generated when the usual flow is followed. Knowledge_object refers to the knowledge generated when scene and action classifier is not used and all images are diverted to the object detector for identifying objects inside the image. Knowledge_object_gt and Knowledge_object_scene_action_gt refers to the scenario where only ground truth relations is used to generate knowledge triples.

Finally all these combinations are compared to the output generated by the state of the art model [5] over FVQA dataset (marked in bold). The best results obtained by the proposed approach is also marked in bold and knowledge obtained using the ground truth relation is not considered for comparison.

The outputs are comparable with the state of the art approach. Image + Knowledge_object _scene_action + Question combination works best so far, for the proposed Architecture described in Figure 19 in combination with the Stacked attention based Model demonstrated in Figure 22. The @1 accuracy is 61.92 which is very close to 62.20 obtained in [5]

Modality	**Accuracy@1**	**Accuracy@3**	**Accuracy@10**
Image + Question	48.20%	67.31%	78.12%
Knowledge + Question	56.20%	72.80%	82.60%
Image + Knowledge + Question	**61.92%**	**78.20%**	**86.00%**

Table 11: Performance of Stacked Attention Network based model on different modalities

Knowledge Embedding	**Accuracy@1**	**Accuracy@3**	**Accuracy@10**
GCN + BERT	51.40%	67.40%	82.00%
Glove	**56.20%**	**72.80%**	**82.60%**

Table 12: Performance fluctuations over various types of knowledge embedding

Different knowledge generation approaches	**Accuracy@1**	**Accuracy@3**	**Accuracy@10**
Image + Knowledge_object + Question	59.80%	75.20%	85.40%
Image + Knowledge_object _scene_action + Question	**61.92%**	**78.20%**	**86.00%**
Image + Knowledge_object_gt + Question	63.60%	76.80%	84.60%
Image + Knowledge_object_scene_action _gt + Question	65.20%	77.00%	85.40%
Straight to the Facts-Question + Visual Concepts[5]	**62.20%**	**75.60%**	-

Table 13: Performance of Stacked Attention Network based model on different knowledge combinations

9 Analysis

9.1 Analysis of Relation Prediction and Visual Concept Prediction Model

The various experiments conducted gives a fair idea of how well the Relation prediction and Visual Concept Prediction model fits in the proposed architecture.
The relation prediction model has 76.35% accuracy. In order to estimate the error overhead generated by this model, knowledge triples are generated with the ground truth relation. As per the results shown in Table 13, the knowledge combination referred to as, Knowledge_object_scene_action_gt with ground truth relation gives 65.20% accuracy and knowledge combination referred to as Knowledge_object_scene_action produces 61.92% accuracy. A difference in 3.28% accuracy is caused due to the faulty relation predicted by the relation prediction model. The train-validation loss curve for the relation prediction model is shown in Figure 23

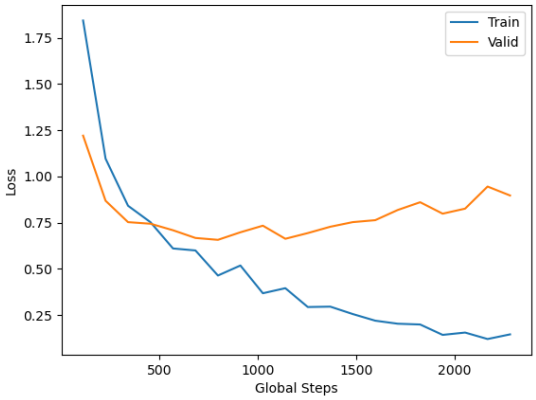


Figure 23: Train-Validation loss curve for Relation Prediction model

The Visual Concept prediction model has 97.60% accuracy. In order to estimate the error overhead generated by this model, knowledge triples are generated by keeping only the object detector workflow. The entire decision making part of whether to use a scene, action or object predictor shown in Figure 19 is eliminated. The entire data traffic is directed to the object prediction model. From Table 13, the combination Knowledge_object_scene_action with the decision making process (which detector needs to be selected) gives 61.92% accuracy and knowledge combination referred by Knowledge_object (without additional decision making overhead) produces 59.80% accuracy. This means that the visual concept prediction model is vital and should be used to predict the required the visual concept i.e. object, action or scene. This in turn shall produce different subject like scene or action or object and hence effect the knowledge triple production. So, for a question that requires the knowledge of a scene in an image, there is no utility for the objects predicted for that image. By simply passing all image into a object detector and feeding this in COMET will not enrich the triple set generated by COMET. There is an even lower possibility of having the required ground truth triple in the triple set produced by COMET. The train-validation loss curve for the visual concept prediction model is shown in 24

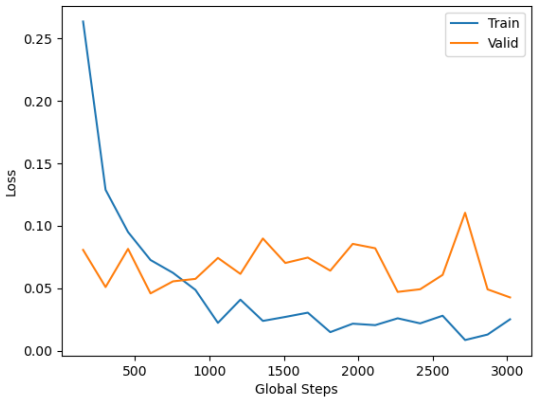


Figure 24: Train-Validation loss curve for Visual Concept Prediction model

9.2 Analysis of COMET Generated knowledge graph Analysis

The triples generated by COMET based on the visual concepts predicted and the corresponding relation from the Relation prediction model, precisely contains the ground truth knowledge triple required to answer the question in only 20.92% cases. This is the most significant area where there is a possibility of improvement.It would be interesting to see if the ground truth triple mapping accuracy is improved, will the model perform better. However, how can COMET produce the required exact triple to answer a question? We are not dealing with a static knowledge base here, which can be manipulated to extract the required knowledge. We are dealing with a Model that constructs new nodes based on a given node and relation. The nodes that are generated are result of graph completion and natural language generation modules. It is therefore understood that comet may not necessarily produce the required knowledge to answer a question. In spite of this, the final transformer based model or stacked attention based model is developing a common sense understanding with which it is producing comparable results.
Also this model could perform well on a dataset which is build without keeping any ground truth fact in mind, i.e. there may not necessarily be any particular fact that is used to answer a question for given an image, rather it requires a more of general understanding of objects or scenes or actions to answer a question.

9.3 Analysis of Transformer based model and Stacked attention based model

The transformer based model delivers 54.69% accuracy@1, 70.08% accuracy@3 and 78.79% accuracy@10 respectively. The Accuracy@3 (the correct answer is present in one of the top 3 classes predicted by the model) obtained for the Transformer based model, is very close to the Accuracy@3 obtained in [5] which is 75.60%.

However due to very few training samples [7], the model overfits. The problem of overfitting is also reported in other SOTA models like [11] over FVQA8 dataset. The loss, accuracy curve for training and validation set while training the transformer based model is shown in Figure 25

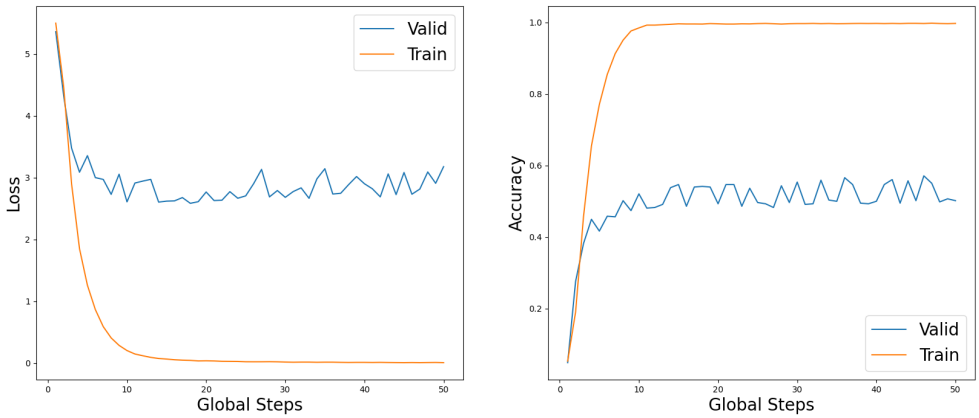


Figure 25: Loss-Accuracy@1 curve for Transformer based Model

The stacked attention network based model delivers, 61.92% accuracy@1 and 78.20% accuracy@3 and 86.00% accuracy@10 respectively. In this case, accuracy@3 for the proposed architecture surpasses the 75.60% accuracy@3 mark obtained in [5].Although the accuracy have improved significantly compared to the Transformer based model, the problem of over fitting persists. This problem was addressed with early stopping and reducing the model complexity but the scarcity of data is a major set back in this regard. The model over fitting was also dealt with by removing image features (feature reduction) as the visual concepts were already embedded in knowledge. Nevertheless, the overfitting did not go away instead the accuracy reduced consequentially. The loss, accuracy curve for training and validation set while training the Stacked attention network based model is shown in Figure 26.

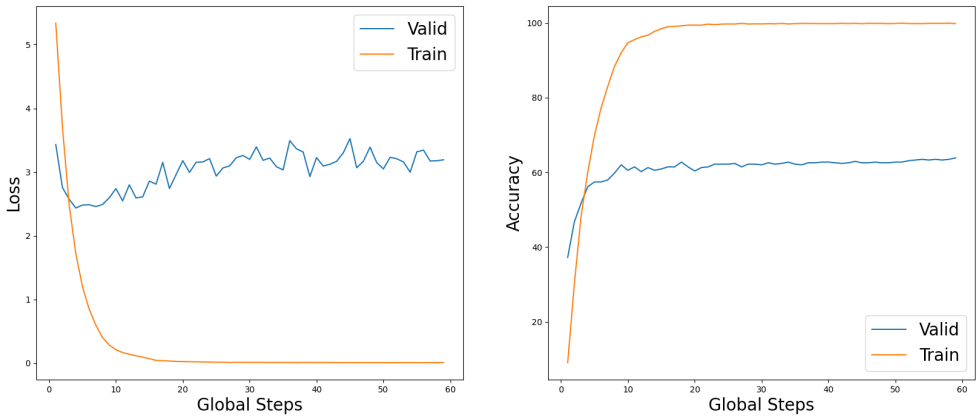


Figure 26: Loss-Accuracy@1 curve for Stacked Attention based Model

9.4 Correct output generated by the proposed Architecture

With respect to the proposed architecture in Section 6.3, the Stacked attention based model performs better than the transformer based model. Hence, the following analysis is

made on the data processed by the pipeline in combination with the Stacked attention based Model. The correct answers predicted by the model could broadly be classified in three different categories. The first category of correctly predicted answers fall under the scenario where there is no faulty prediction in the entire pipleline. As reported in Figure 27, the relation prediction, visual concept prediction generates the correct output. The triples in this section is written in "*relation subject object*" format. The ground truth triple required to answer the question i.e.

$$IsA \quad banjo \quad musical - instrument\ (left)$$

and

$$IsA \quad apple \quad fruit\ (right)$$

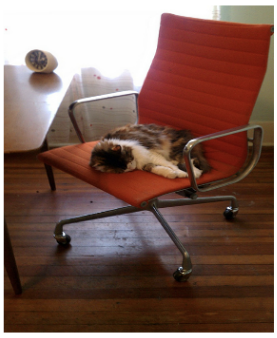
is correctly generated by COMET. The predicted output is also correct.

	
Question: Which object in this image is a musical instrument?	**Question: Which round object in this image is a piece of fruit?**
Correct Visual Concept: Object **Predicted Visual Concept: Object** **Detected Object: banjo**	**Correct Visual Concept: Object** **Predicted Visual Concept: Object** **Detected Object: person, apple**
Correct Relation: IsA **Predicted Relation: IsA**	**Correct Relation: IsA** **Predicted Relation: IsA**
Correct Triple: **IsA banjo musical instrument** **COMET generated triples:** **IsA banjo musical instrument** **IsA banjo instrument** **IsA banjo string instrument** **IsA banjo woodwind** **IsA banjo music instrument**	**Correct Triple:** **IsA apple piece of fruit** **COMET generated triples:** **IsA person hold cigarette** **IsA person person** **IsA person play frisbee** **IsA person ski down mountain slope** **IsA person human be** **IsA apple fruit** **IsA apple domesticate plant** **IsA apple food** **IsA apple apple** **IsA apple good food**
Correct Answer: Banjo **Predicted Answer: Banjo**	**Correct Answer: apple** **Predicted Answer: apple**

Figure 27: Correct Output with no fault in pipeline

The second category of correctly predicted answers fall under the scenario where COMET may not generate the exact ground truth triple but still the model predicts the correct answer. The model is able to decipher the correct answer from closely related triples generated by COMET. As reported in Figure 28, the relation prediction, visual concept prediction generates the correct output. However ground truth triple required to answer the question (see left of Figure 28) is not predicted by COMET. However it was able to predict the correct

answer based on an understanding of the generated triples which are closely related to the ground truth triple. Towards right side of the figure 28, zucchini and cucumber are predicted as objects, the knowledge that it is vegetable and belongs to the food category is learned by the model from the generated triples.

<table>
<tr><td>  </td><td>  </td></tr>
<tr><td>Question: Which object in this image has a back you can lean against?</td><td>Question: Which object in this image belongs to the category Foods?</td></tr>
<tr><td>Correct Visual Concept: Object
Predicted Visual Concept: Object

Detected Object: dining table, chair</td><td>Correct Visual Concept: Object
Predicted Visual Concept: Object

Detected Object: zucchini, cucumber</td></tr>
<tr><td>Correct Relation: HasA
Predicted Relation: HasA</td><td>Correct Relation: BelongTo
Predicted Relation: all relations used (since data is not from conceptnet)</td></tr>
<tr><td>Correct Triple:
HasA chair back you can lean against

COMET generated triples:

HasA dining table chair in it
HasA dining table table
HasA dining table two leg
HasA dining table leg
HasA dining table four leg and two side
HasA chair leg
HasA chair back and front
HasA chair two legs
HasA chair four leg
HasA chair seat</td><td>Correct Triple:
BelongTo vegetable food

COMET generated triples:

IsA zucchini vegetable
IsA zucchini fruit
IsA zucchini herb
IsA zucchini squash
IsA zucchini grain
IsA cucumber plant
IsA cucumber vegetable
IsA cucumber fruit
IsA cucumber flower
IsA cucumber cucumber</td></tr>
<tr><td>Correct Answer: chair
Predicted Answer: chair</td><td>Correct Answer: vegetable
Predicted Answer: vegetable</td></tr>
</table>

Figure 28: Correct Output with generalised understanding by the model

The third category of correctly predicted answers fall under the scenario where the model tries to learn from the image features. As reported in Figure 29, neither axe (left) nor keyboard (right) is one of the detected objects. However the answer is predicted correctly. A plausible explanation could be that the model is relying more on image features for the prediction.

	
Question: Which object in this image are dangerous?	**Question: Which object in this image is used for type letter?**
Correct Visual Concept: Object **Predicted Visual Concept: Object** **Detected Object: hammer, hatchet**	**Correct Visual Concept: Object** **Predicted Visual Concept: Object** **Detected Object: mouse**
Correct Relation: HasProperty **Predicted Relation: HasProperty**	**Correct Relation: UsedFor** **Predicted Relation: UsedFor**
Correct Triple: **HasProperty axe dangerous** **COMET generated triples:** **HasProperty hammer heavy** **HasProperty hammer hard or soft** **HasProperty hammer sharp** **HasProperty hammer dangerous to person** **HasProperty hammer very heavy** **HasProperty hatchet dangerous** **HasProperty hatchet sharp** **HasProperty hatchet useful when chop firewood** **HasProperty hatchet use for carve wood** **HasProperty hatchet heavy**	**Correct Triple:** **UsedFor keyboard type letter** **COMET generated triples:** **UsedFor mouse write** **UsedFor mouse play** **UsedFor mouse make software** **UsedFor mouse do crossword puzzle** **UsedFor mouse program**
Correct Answer: axe **Predicted Answer: axe**	**Correct Answer: keyboard** **Predicted Answer: keyboard**

Figure 29: Correct Output with no exact matches for object detection

9.5 Incorrect output generated by the proposed Architecture

The wrong answers predicted by the model could broadly be distinguished in three categories. The first category of wrongly predicted answers fall under the scenario where there is fault in the pipeline of the architecture. As reported in Figure 30, the object detector is not able to predict the object on which the question is asked. Hence the complete flow is erroneous as well as the predicted final answer.

<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Question: Which object in this image belongs to the category Sports originating in England?</td>
<td>Question: Which object in the background is used for mass transportation?</td>
</tr>
<tr>
<td>Correct Visual Concept: Object
Predicted Visual Concept: Object

Detected Object: rugby ball</td>
<td>Correct Visual Concept: Object
Predicted Visual Concept: Object

Detected Object: bicycle, person</td>
</tr>
<tr>
<td>Correct Relation: BelongTo
Predicted Relation: AtLocation</td>
<td>Correct Relation: UsedFor
Predicted Relation: UsedFor</td>
</tr>
<tr>
<td>Correct Triple:
BelongTo soccer ball sports originating in england

COMET generated triples:

AtLocation rugby ball sport good store
AtLocation rugby ball soccer field
AtLocation rugby ball football stadium
AtLocation rugby ball gym
AtLocation rugby ball school</td>
<td>Correct Triple:
UsedFor bus mass transportaion

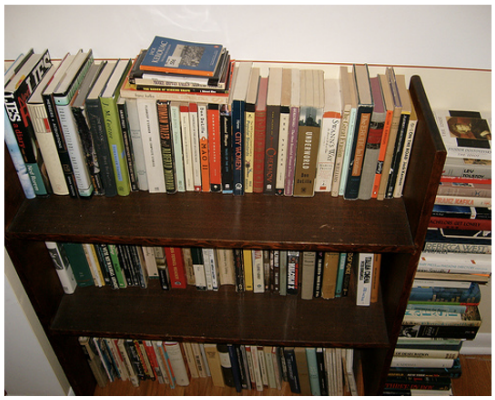
COMET generated triples:

UsedFor bicycle travel
UsedFor bicycle transport thing
UsedFor bicycle go from place to place
UsedFor bicycle ride in
UsedFor bicycle transportation
UsedFor person play with on playground
UsedFor person talk to
UsedFor person help
UsedFor person love
UsedFor person work for you</td>
</tr>
<tr>
<td>Correct Answer: soccer ball
Predicted Answer: rugby ball</td>
<td>Correct Answer: bus
Predicted Answer: bicycle</td>
</tr>
</table>

Figure 30: Wrongly predicted answers due to Faulty pipeline

The second category of wrongly predicted answers fall under the scenario where COMET fails to produce correct or closely related triples. As reported in Figure 31, the object detector predicts bookcase (Figure 31 left) which is close enough to bookshelf but COMET is not able to produce the the precise location. Similarly, COMET produces "cabinet", "refrigerator" and "cupboard" as possible locations where a bottle can be found but not "table" (Figure 31

right). The correctness of COMET generated triples are high but the relevance with respect to the question is low.



Question: Where does the object in the image can be found in?

Correct Visual Concept: Object
Predicted Visual Concept: Object

Detected Object: bookcase

Correct Relation: AtLocation
Predicted Relation: AtLocation

Correct Triple:
AtLocation bookshelf furniture store

COMET generated triples:

AtLocation	bookcase	library
AtLocation	bookcase	room
AtLocation	bookcase	house
AtLocation	bookcase	home
AtLocation	bookcase	den

Correct Answer: furniture store
Predicted Answer: book

Question: Which object in this image can be found in table?

Correct Visual Concept: Object
Predicted Visual Concept: Object

Detected Object: cup, bottle, tv

Correct Relation: AtLocation
Predicted Relation: AtLocation

Correct Triple:
AtLocation bottle table

COMET generated triples:

AtLocation	cup	kitchen
AtLocation	cup	cupboard
AtLocation	tv	house
AtLocation	tv	in room
AtLocation	tv	home
AtLocation	tv	apartment
AtLocation	tv	bedroom
AtLocation	bottle	cabinet
AtLocation	bottle	refrigerator
AtLocation	bottle	cupboard

Correct Answer: bottle
Predicted Answer: house

Figure 31: Wrongly predicted answers due to inaccurate COMET triple prediction

The third category of wrongly predicted answers fall under the scenario where the dataset lacks proper annotations. As reported in Figure 32, the correct answer and predicted answer (Figure 32 left) "cook food" and "cooking" respectively are same. There is only one correct

answer per question in the dataset but there can be multiple correct answers for a question. Similarly "baseball bat" and "baseball" are both correct (Figure 32 right) answers for the given question.

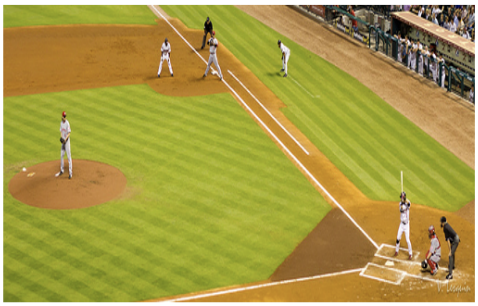
	
Question: What is the place in this image used for?	**Question: What object can be used to hit person?**
Correct Visual Concept: Place **Predicted Visual Concept: Place** **Detected Place: kitchen, artists_loft**	**Correct Visual Concept: Object** **Predicted Visual Concept: Object** **Detected Object: person, baseball bat**
Correct Relation: UsedFor **Predicted Relation: UsedFor**	**Correct Relation: UsedFor** **Predicted Relation: UsedFor**
Correct Triple: **UsedFor kitchen cook food** **COMET generated triples:** **UsedFor kitchen cook** **UsedFor kitchen eat dinner** **UsedFor kitchen prepare food** **UsedFor kitchen serve food** **UsedFor kitchen make food** **UsedFor artists_loft art** **UsedFor artists_loft artist to display artwork** **UsedFor artists_loft paint picture** **UsedFor artists_loft live in** **UsedFor artists_loft display art**	**Correct Triple:** **UsedFor baseball hit person** **COMET generated triples:** **UsedFor person play chess with** **UsedFor person talk to** **UsedFor person help** **UsedFor person love** **UsedFor person work** **UsedFor baseball bat hit** **UsedFor baseball bat smash something** **UsedFor baseball bat throw** **UsedFor baseball bat injure person** **UsedFor baseball bat bat**
Correct Answer: cook food **Predicted Answer: cooking**	**Correct Answer: baseball** **Predicted Answer: baseball bat**

Figure 32: Some close misses

10 Comparative Study

10.1 Straight to the Facts: Learning Knowledge Base Retrieval for Factual Visual Question Answering [5]

This work is described in Section 3.6.3, The authors have modified the raw questions so that the relation prediction module can perform better. In the proposed approach however, no such manipulation is done. In spite of this, the Relation Prediction model in the proposed approach delivers 76.5% accuracy compared to [5] which produce 75.4% accuracy.

The accuracy achieved by Narasimhan et al. [5] is 62.20%. On the other hand the SAN [9] based proposed approach delivers 61.92% accuracy. After reviewing all the the parts of the approach described in [5], the only module where this proposed approach lags behind is the ground truth query mapping module. For the proposed approach in this thesis (with Stacked attention based model in the end), when ground truth triples are supplied along with COMET triples, the accuracy increases to 82.34%. Hence there is a direct correlation between the presence of ground truth triple in knowledge set with the accuracy. Inspite of the fact that ground truth triple prediction model in the proposed approach produce 20.92% accuracy compared to 64.50% reported in [5], the final accuracy of 61.92% and 62.20% for the proposed model and [5] respectively are close.

10.2 Zero-shot Visual Question Answering using Knowledge Graph [11]

This work is described in Section 3.6.2. In this method, the authors focus on removing the pipeline from the architecture and replace the classifier layer with zero shot learning. However, the accuracy acheived on FVQA dataset is 58.27%, which has not surpassed the 61.92% accuracy mark. In this paper the authors have trained a fact space with the ground truth facts avalaible in the training data, then they have mapped a fact and relation to a question. They have run a for loop in the end to check if the fact-relation pair maps to some answer in the KB.In this entire approach, one very crucial assumption is the availability of ground truth facts for training. In order to compare the approach by Chen et al. [11], the fact space is retrained with the COMET generated facts. This new fact fusion model generated is used for final joint testing described in [11]. The performance of the Zero-shot approach when trained with ground truth fact and COMET generated fact on FVQA dataset is shown in Table 14

Method	**Accuracy@1**	**Accuracy@3**	**Accuracy@10**
ZS-F-VQA-non classifier based approach on FVQA dataset[5]	58.27%	75.2%	86.40%
ZS-F-VQA-non classifier based approach trained on COMET generated fact space	42.69%	62.31%	78.46%

Table 14: Performance of ZS-F-VQA-non classifier based model over standard F-VQA datasets

10.3 Final Comparison of the best Model so far with other SOTA Approaches

The best performance in each category is highlighted in the performance comparison table 15 below.

Method	**Accuracy@1**	**Accuracy@3**	**Accuracy@10**
LSTM-Question+Image+Pre-VQA [7]	24.98%	40.40%	-
Hie-Question+Image+Pre-VQA [7]	43.14%	59.44%	-
FVQA[7]	56.91%	64.65%	-
Ensemble[7]	58.76%	-	-
Straight to the Facts-Question + Image [5]	26.68%	30.27%	-
Straight to the Facts-Question + Image + Visual Concepts [5]	60.30%	73.10%	-
Straight to the Facts- Question + Visual Concepts[5]	**62.20%**	75.60%	-
ZS-F-VQA-non classifier based approach on FVQA dataset[11]	58.27%	75.2%	**86.40%**
Proposed Best Model (see 8.5)	61.92%	**78.20%**	86.00%

Table 15: Performance of different models over standard F-VQA datasets

11 Future Work

11.1 Extend COMET to check the correctness of a given triple

Until now, knowledge construction models were used to generate triples based on given subject and relation. It is niether able to predict a set of entities or subject given a relation and object nor able to validate the correctness or degree of correctness for a given triple.
Let us consider an image with an aeroplane and two helicopters, the question says "Which object in this image is used for carrying goods?". The relation predictor predicts the correct realation: $UsedFor$, similarly the visual concept model predicts *object* and then the object detector predicts *aeroplane* and, *helicopter*. COMET Input is

$$aeroplane \quad UsedFor$$

and it completes the triples by producing

$$aeroplane \quad UsedFor \quad Carrying\ Passengers$$

$$aeroplane \quad UsedFor \quad travelling$$

For the second object predicted by the object detector, COMET takes

$$helicopter \quad UsedFor$$

as input, COMET delivers

$$helicopter \quad UsedFor \quad flying$$

$$helicopter \quad UsedFor \quad going\ from\ one\ place\ to\ another$$

as output. However

$$aeroplane \quad UsedFor \quad Carrying\ goods$$

is not generated by COMET, hence the required information to answer the question is not produced. Infact for a general case this required triple may or may not be produced by a automatic knowledge construction graph [8].
A possible future work can henceforth be proposed, where a knowledge completion graph could be used to rate the accuracy, novelty, diversity and correctness of a triple. Say, if the question predicate, *carrying goods* is learned by certain model. Then two possible triples can be formulated from the given information,

$$aeroplane \quad UsedFor \quad carrying\ goods$$

and

$$helicopter \quad UsedFor \quad carrying\ goods$$

In this case if the knowledge construction graph could deliver the probability of correctness of these triples then the triple mapping would be generic(can be applied to any data) and at the same time, specific to the given question-image pair.
Probably then all the drawbacks of static knowledge base related to ranking, and the constant confusion regarding which knowledge base to use, can be mitigated. The automatic knowledge

graph construction model learns and generates triples based on the KB it is trained on. If such a model could be trained on multiple KBs then it's ability to predict the integrity of a given triple would be high, then the system would flourish with exact (check correctness for triple specific for a given question-answer pair) as well as abundant knowledge(can work on any given question-answer pair, not KB specific)
Extend COMET to accept complete triple as input and check if its a plausible knowledge and vouch for its correctness or provide a degree of its correctness. It would then be interesting to integrate the same in a VQA setup.

11.2 Modify Proposed Architecture with additional triple scoring module

From the detailed analysis of proposed architecture and results, it is evident that output from knowledge completion models generates triples that may not be the exact ones required to answer the questions as opposed to different methods with very high query mapping accuracy. This limitation results from COMET's inability to understand which triple is necessary to answer the question for a given image. It uses a top-k [8] algorithm to generate top triples with the highest precision given a specific entity and relation. The value of k in top-k can be increased to include more options, i.e., if top-5 is requested, COMET generates five triples for a given subject and relation. Thus, there is a possibility that by increasing k, COMET might be able to generate a triple that can answer the question, which will result in a high dimensional knowledge feature space.
A solution to this can be selecting n-random triples out of the top-k, giving access to a broader selection of facts with a trade-off for a less relevant one. However, a more reasonable solution is to score the triples generated by COMET based on the joint embedding of image elements, such as objects, actions, or scenes and, the question. Triples can also be selected from the original knowledge base using a scoring technique so that only relevant knowledge is used to train the model. The scores given to the triples could be trained such that the highest score is given to the triple closest to the ground truth triple using the method described in [5].

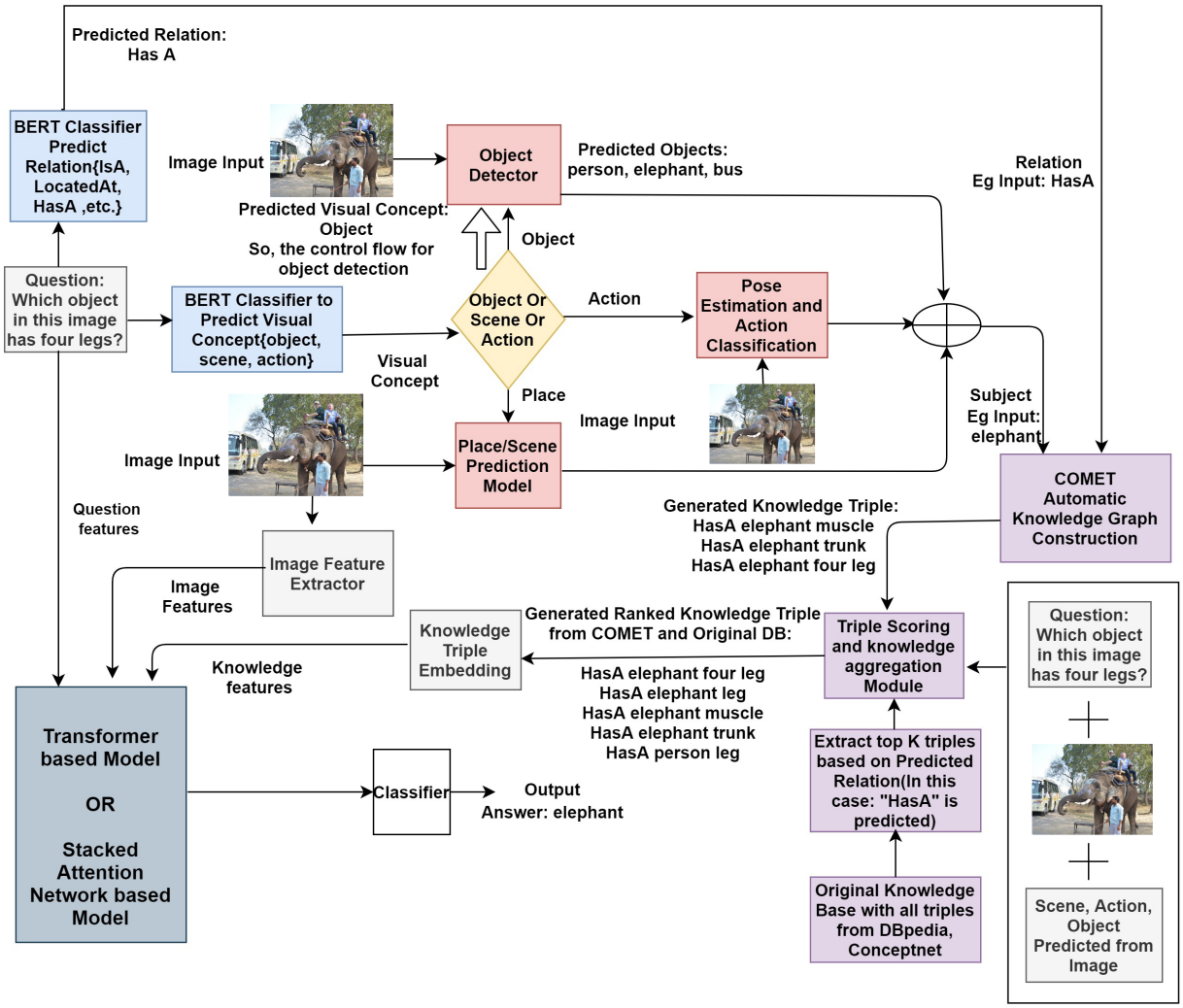


Figure 33: Modified Architecture.

11.3 Build new knowledge-based VQA dataset

After studying all the datasets in this domain, one very important finding is that all state of the art approaches encounter over fitting of models due to scarcity of data as mentioned in [11]. To deal with this issue, a new dataset can be build with more number of questions. One should consider building the dataset, without keeping in KB in mind i.e. the question should not be framed in a way such that the answer has to be present in KB.

12 Conclusion

Integrating Common sense knowledge into VQA System is an unexplored problem in Natural language Processing and computer vision domain. The primary goal of the thesis was to integrate knowledge into the VQA system. From the experiments conducted and the findings gathered, it can be concluded that COMET is not able to generate the exact triple required to answer the question consistently. Nevertheless, It is able to generate knowledge which helps the model get an idea about common sense knowledge related to the visual concepts in the image.

The performance of the proposed model is at par with the performances of other SOTA approaches. The architecture has been proved to be robust under various settings (adding or removing modules from the pipeline does not hamper the accuracy by larger margin). It is also evident that using an automatic knowledge base construction module is the futuristic approach for integrating knowledge and improving the models' understanding. Integrating the entire KB or using filtering methods hinders the model from learning about facts associated with the given question-image pair.

The detailed analysis of the proposed model's failure scenarios demands for a knowledge graph completion module that predicts the exactness of a generated knowledge triple, such that knowledge integration becomes hassle-free. Also, the actual verification of this approach is suitable on a dataset that is built without keeping any KB fact in mind, i.e., there may not necessarily be any particular fact used to answer a question given an image. Instead, it requires more of common sense to answer a question.

This means that the research question that prompted building a more generic knowledge based VQA system is answered with optimal utilization of a trained automatic knowledge graph construction module.

List of Figures

List of Tables

References

[1] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211—-226, 2004.

[2] Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for ifthen reasoning. *AAI*, 2019.

[3] N. Tandon, G. de Melo, F. Suchanek, and G. Weikum. Webchild: Harvesting and organizing commonsense knowledge from the web. *International Conference on Web Search and Data Mining. ACM*, 2014.

[4] François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. ConceptBert: Concept-aware representation for visual question answering. pages 489–498, November 2020.

[5] Medhini Narasimhan and Alexander G. Schwing. Straight to the facts: Learning knowledge base retrieval for factual visual question answering. 2018.

[6] Maryam Ziaeefard and Freddy Lecue. Towards knowledge-augmented visual question answering. pages 1863–1873, December 2020.

[7] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. Fvqa: Fact-based visual question answering. 2017.

[8] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. 2019.

[9] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. 2016.

[10] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and R. Mottaghi. Multi-modal answer validation for knowledge-based vqa. *ArXiv*, abs/2103.12248, 2021.

[11] Zhuo Chen, Jiaoyan Chen, Yuxia Geng, Jeff Z. Pan, Zonggang Yuan, and Huajun Chen. Zero-shot visual question answering using knowledge graph. 2021.

[12] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. 2020.

[13] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. Explicit knowledge-based reasoning for visual question answering. 2015.

[14] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. 2019.

[15] Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. Commonsense knowledge base completion with structural and semantic context. 2019.

[16] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web, Springer*, 4(14):722—-735, 2007.

[17] Sumithra Bhakthavatsalam, Kyle Richardson, Niket Tandon, and Peter Clark. Do dogs have whiskers? a new knowledge base of haspart relations. 2020.

[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. 2018.

[19] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. 2019.

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[21] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. 2019.

[22] Tuong Do, Thanh-Toan Do, Huy Tran, Erman Tjiputra, and Quang D. Tran. Compact trilinear interaction for visual question answering. 2019.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. 2015.

[24] A. Radford. Improving language understanding by generative pre-training. 2018.

[25] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[27] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. 2018.

[28] Techniques of knowledge representation - the coding bus. `https://thecodingbus.info/techniques-of-knowledge-representation/`.

[29] data structures - what's a rdf triple? - stack overflow. `https://stackoverflow.com/questions/273218/whats-a-rdf-triple/`.

[30] Applying w3c semantic web standards to lims. `https://www.thefreelibrary.com/Applying+W3C+Semantic+Web+standards+to+LIMS%3A+emerging+applications...-a0139966085`.

[31] Data stewards and data management principles. `https://courses.cs.ut.ee/LTAT.02.014/2021_spring/uploads/Main/L04_21_03_09_FAIR_I.pdf`.

[32] Presentation metadata - europa. https://data.europa.eu/sites/default/files/d2.1.2_training_module_1.3_introduction_to_rdf_sparql_en_edp.pdf.

[33] What is a knowledge graph? | ibm. https://www.ibm.com/cloud/learn/knowledge-graph.

[34] Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. *Proceedings of the National Conference on Artificial Intelligence*, 2, 07 2014.

[35] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3027–3035, Jul. 2019.

[36] Conceptnet. https://conceptnet.io/.

[37] Exploiting structural and semantic context for commonsense knowledge base completion. https://deepai.org/publication/exploiting-structural-and-semantic-context-for-commonsense-knowledge-base-completion.

[38] Introduction to knowledge graph embedding — dglke. https://dglke.dgl.ai/doc/kg.html.

[39] Summary of translate model for knowledge graph embedding. https://towardsdatascience.com/summary-of-translate-model-for-knowledge-graph-embedding-29042be64273.

[40] Nasrullah Sheikh, Xiao Qin, Berthold Reinwald, Christoph Miksovic, Thomas Gschwind, and Paolo Scotton. Knowledge graph embedding using graph convolutional networks with relation-aware attention. 2021.

[41] Understanding graph convolutional networks for node classification. https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b.

[42] H. Abdi. The eigen-decomposition : Eigenvalues and eigenvectors. 2006.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[44] What is a transformer? — inside machine learning - dzone ai. https://dzone.com/articles/what-is-a-transformer-inside-machine-learning.

[45] Bert explained: A complete guide with theory and tutorial. https://medium.com/@samia.khalid/bert-explained-a-complete-guide-with-theory-and-tutorial-3ac9ebc8fa7c.

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[48] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[49] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, 1, 05 2015.

[50] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. What value do explicit high level concepts have in vision to language problems? 2016.

[51] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering. 2016.

[52] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. `https://nlp.stanford.edu/projects/glove/`.

[53] Word2vec. `http://vectors.nlpl.eu/repository/`.

[54] Thushan Ganegedara. Intuitive guide to understanding glove embeddings. 2019. `https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010`.

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2015.

[56] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. pages 248–255, 2009.

[57] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997, 2014.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[59] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. 2015.

[60] Raymond Cheng. Bert text classification using pytorch. `https://towardsdatascience.com/bert-text-classification-using-pytorch-723dfb8b6b5b`.

[61] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. 2018.

[62] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. *GitHub repository*, 2017.

[63] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. 2018.

[64] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.

[65] Grigorios Kalliatakis. Keras-vgg16-places365. 2017.

[66] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[67] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. USA, 2011. IEEE Computer Society.

[68] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2017.

[69] A gentle introduction to cross-entropy for machine learning. `https://machinelearningmastery.com/cross-entropy-for-machine-learning/`.

[70] Donglai Zhu, Hengshuai Yao, Bei Jiang, and Peng Yu. Negative log likelihood ratio loss for deep neural network classification. 2018.

[71] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[72] Accuracy precision recall f1 score: Interpretation of performance measures. `https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/`.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Druck-Exemplaren überein.

15.09.2021 Adrika Mukherjee

Datum und Unterschrift:

Declaration

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references that the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted hard copies.

15.09.2021 Adrika Mukherjee

Date and Signature: