CLEVR

Thinking:

1. It seems that we can use the graph to represent the positional relationship between different objects in the CLEVR dataset image. Take a look at Fig.2 in the CLEVR paper.
2. Next: read the paper about scene graph(Visual gnome) and the methods for attacking CLEVR.

Visual Gnome

1. How can we use the scene graph to facilitate the VQA task??
2. Relationship is important between different objects in VQA and different higher-level abstractions in QA between different sentences.
3. The visual representation is like knowledge graph in NLP tasks.

Graph-structured Representations for Visual Question Answering

Note: in VQA paper, we need to keep focus on three things: 1. How to deal with the image, 2. How to process the question, 3. How to produce the answer (reason).

Why:

1. CNN can capture the absolute position of each object, but not the relative location between them
2. LSTM cannot reflect the true complexity of language structure. This is because in VQA, the questions trained on cannot fully cover all the combinatorial diversity.

How:

1. Language: dependency parser: extract a graph from the question. Glove: embed each word to H=300 dim.
2. Image: scene graph, each node is the feature vector provided by the dataset, then like word embedding, projects them into H=300 dim. Each edge is the relative position (the difference between coordinates, if occluded in the depth direction) between two objects, thus, the scene graph is a fully-connected graph. **Note:** this is not a cheat. For real image, we can use the object detector to extract the corresponding coordinates and feature of each object in the image. (Q: how to get the affine transition parameter W and b for image feature??)
3. Reason: use the GNN(GRU) to get the representation for each node. Use the attention-like skill to compute a scalar weights(match two graph). Then use this weight and the word and image node representation to compute the probability of each possible answer.

Future work:

1. Note that the model can only generate the answer between pre-defined ones. Can it extends to answering by the natural language??
2. Can we extend the model to more complex dataset which requires more reasoning step, such like CILVR, Visual Genome??

3. This work only includes the positional relationship between objects, can we use this framework to solve more abstract relationship, such like “A boy is kicking a ball” (Visual Gnorme dataset)
4. How to use the unified framework GNN to unify the image and text representation, which we can solve VQA and QA together?
5. Can we train a end-to-end neural network from a image or text to graph directly??

GLoMo: Unsupervisedly Learned Relational Graphs as Transferable Representations

Why:

CNN and RNN only capture the high structural data, instead of more complex data such like graph. This article addresses: 1) feature vs graph deep transfer learning, 2) unsupervisedly learn a latent graph representation.

What:

This work is focusing on how to transfer the input data (image or context) to a graph using unsupervised learning method.

Future work:

1. Whether can we use it on more image tasks, such like image object detection? So that we can use them in VQA?

HOTPOTQA: A Dataset for Diverse, Explainable Multi-hop Question Answering

Why:

1. Existing QA datasets only focus on the one-paragraph reasoning, which is not complex and challenging enough for the machine to reason.
2. Existing methods for QA come out the answering without knowing about the reasoning process, which means they cannot find the supporting facts from provided context.

What:

1. Source: wikipedia
2. Question: questions about a and b, particularly, yes/no questions, and comparison questions.Note that answering comparison questions are a challenge for QA systems.
3. Reasoning type: there are 5 kinds of reasoning types, it seems that only depending on the entity reference cannot solve all of the reasoning types. If the QA system wants to solve all of these questions, it must have a higher level reasoning ability.
4. Supporting Fact.

Yin and Yang: Balancing and Answering Binary Visual Questions

Why:

Control the influence of language prior (so that the nn can answer the questions correctly without identifying the image) for supervision. That is, control the bias of the dataset, in order to implement the “real” reasoning.

Yin and Yang(yes/no questions, control the language prior(unbiased)) ----> Visual Genome (graph representation, extract more complex relationships besides the relative positions) -----> CILVR (unbiased, need to give the reason process, which means instead of giving the answer, the model need to also give the reason steps).

TrivalQA(supporting facts always in one paragraph) ----> HotpotQA (multi-paragraph reasoning).

Variational Reasoning for Question Answering with Knowledge Graph

Why:

1. Current best models don't train end-to-end model
2. Current models cannot do the multi-hop reasoning
3. How to locate topic entities in the KG
4. Fine-grained annotations(type of questions, the exact logic reasoning steps) are few

What:

Assumption: the KG is given. Each question only has one entity. (can we have more than one? Since it is the key for multi-hop reasoning??)

Algorithm: input: Graph G={V(G), E(G)} and questions
Output: the entity in the graph (E(G)). (can it return the relationship V(G)??)

Model(VRN):

Question → topic entity: $P_{\theta1}(y|q_{i})$

Topic entity → answer: $P_{\theta2}(a_{i}|y, q_{i})$

Note that y is a latent variable, it can be any entity in the graph, so it needs to be learnt.

Question → answer: \sum_{y} P_{\theta1}(y|q_{i})*P_{\theta2}(a_{i}|y, q_{i]).

How to get the first conditional probability:

Use the RNN or other NN (Let's say f_{ent})to embed the question into a vector, then use Softmax to get the conditional probability.

How to get the second conditional probabilities:

1. Embed the question using f_{qt}: capture all properties about the question, and the logical reasoning we need to perform.
2. Embed the reasoning subgraph using g: capture reasoning logics, reasoning paths, and reasoning rules.
3. Apply the softmax to the multiplication of two embeddings mentioned above to get the second conditional probabilities.

How to get the parameters \theta1 and \theta2? (don't quit understand it.)
Variance Inference + REINFORCE algorithm (since the value of y is discrete, which are not differentiable).

Experiment:
METAmovie dataset constructed by themself.

Personal Thinking:
What I take away is the methods they model the qa problem into two procedure: extract the topic entity in a question, then use the question and the entity to reason in the knowledge graph.

Graph-based Approach to the Question Answering Task Based on Entrance Exams
Why:
Tackle document understanding problem using graph.

What:
System Architecture:

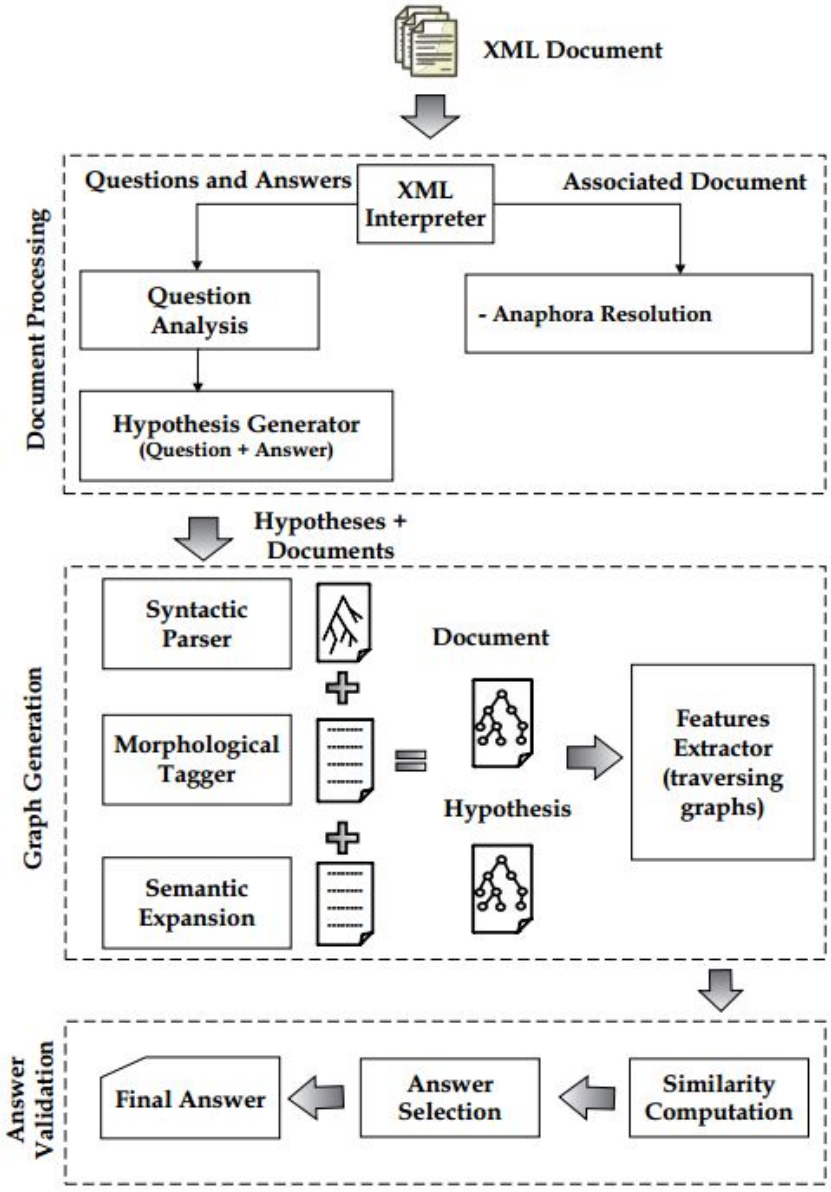

Fig. 1. Graph-based system architecture

1. Doc Preprocessing: XML interpreter sort the question and possible answers to the corresponding document. Question analysis: analyze the question type (who, what ...). Hypothesis Generator: substitute the question type words with the possible answers provided in the XML file. Finally, use Anaphora Resolution to find the reference relationship and substitute them.
2. Graph generation: through Syntactic parser, Morphological Tagger, Semantic Expansion, we get the graph of both document and the hypothesis. For feature

extraction, use Dijskra Algorithm to calculate the shortest distance between the root node to every rest nodes in hypothesis, while also use Dijskra Algorithm to compute the shortest distance between the **same pairs of nodes** in the document.

3. Answer Validation: use the cosine similarity to compute the similarity between the feature of the document and features of four hypothesis, select the highest similarity, then use that corresponding answer as the final answer for the question.

How:
Dataset:
CLEF QA Entrance Exams Task
Experiment:
In the experiment, they used 8 runs for answering questions in 12 documents. Each run extract different lexical, semantic features from the document and questions.
Metric:

$$c@1 = \frac{1}{n}(n_R + n_U \frac{n_R}{n}), \tag{2}$$

where:

n_R: number of correctly answered questions,
n_U: number of unanswered questions,
n: total number of questions.

Personal Thinking:
This paper although uses some techniques for graph, but all of them are just the traditional and hand-making features, so the accuracy of the final model is not very high. Also, their graph representation does not represent the abstract relationship other than the grammarly relations. The future work will focus on how to capture the more abstract relationship from the document, such like “Anne Saxon is the founder of a computer vision company.” which can represented by (a computer vision company, founded by, Anne Saxon).

However, the quite interesting part of this paper is that they preprocessed the question, and substituted the question type word with the word in the answer, which, I think, will help the model to have the attention to those words related to the answer and question when it processes the document. So that’s the question I ask that which one should we process first, document or question? If we process question, we can have it to guide finding the key word or topic entity in the document. If we process document first, we need to extract all the abstract relationship between all, which needs to present a more comprehensive graph representation.

KG^{2}: Learning to Reason Science Exam Questions with Contextual Knowledge Graph Embeddings

Why:

Currently, QA systems are limited to surface-level reasoning. IR and word co-occurrence method fails to solve the reasoning on ARC dataset.

What:

This paper used the similar method as the above one.

1. Generating Hypothesis: find the question type word in the question, then use the word or phrases in the answer to substitute it.
2. Searching Potential Supports: here, they used the generated hypothesis as a query to search the entire corpus, using a local search engine.
3. Construct Knowledge Graph: using Open IE to extract the relation triples (subject, predicate, object), then connect them using the words in both or all triples for supporting sentences tracked by the last step. Also, they build graphs for hypothesis.
4. Learning with Graph Embeddings: here, they belong this question to a graph ranking problem. For each knowledge graph, they used GNN to get every node's representation, then compute every graph cosine similarity for each (hypothesis, supporting facts) pair (Note that all the supporting facts are merged into one knowledge graph), then get the hypothesis, which has the max score for some supporting fact.

How:

Dataset:

AI2 Reasoning Challenge (ARC). Note the questions in this dataset are **multi-choice** questions!

Metric:

Correct: gain one point, if get k-tie correct (chooses multiple choices containing the correct one), gain 1/k point.

Personal Thinking:

First, for what we want to achieve, this paper method has two limitations: 1) it constructs a system which is composed of several designed separate "tools", which is like "RCNN". We want to end-to-end training instead of the separate parts. 2) the dataset they used is different from what we use in that their algorithm can see the potential answers to answer the question, so that they can construct the hypothesis based on them. In our case, we want the algorithm to answer the open-domain questions, so we cannot use this method. 3) In their method, they use the retrieval method, which has the shortcoming that maybe it cannot find the corresponding supports sentences which have sufficient information for that. So the answer can certainly not be able to answer correctly. 4) compared to "Variational Reasoning for QA based on KG", it seems like it does not present well the reasoning path, at least not embed the reasoning subgraph and path. It relies on the external system to get this.

However, they provided us a possibly useful tool to construct the graph from the sentences: OPEN IE. If we cannot solve the first problem, which is "extract the graph from the document",

then maybe we can use this tool to try to extract it, at least it is better than “only semantic” parser presented in the last paper, which only extracts the grammar and lexical relationship between words, does not contain any abstract relations.

Question Answering by Reasoning Across Documents with Graph Convolutional Networks

Why:

To solve the question answering problem of crossing Documents multi-hop reasoning.

What:

1. The representation of the supporting document (doing it offline)

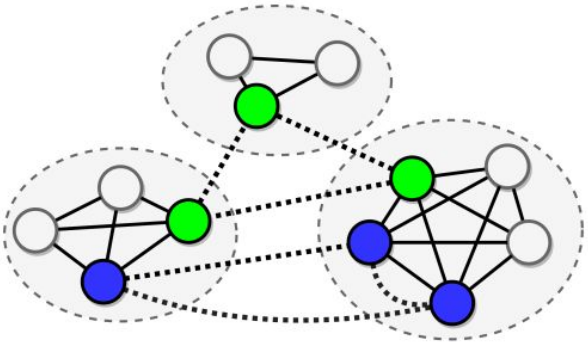

Figure 2: Supporting documents (dashed ellipses) organized as a graph where nodes are mentions of candidate entities. Nodes are connected by two simple relations: one indicating co-occurrence in the same document (solid edges), another connecting mentions of the same entity (dashed edges).

   Each dashed ellipse represents one of the supporting document. Each node represents for a mention for the candidate word appearing in some supporting document. Solid edges represents for the co-occurrence in the same document and the dashed edges represent for mentions of same entity either in the same document of different ones.
2. How to get the representation of the query words and the mentions in the supporting documents? Use the ELMo representation, each word in mentions is the concatenation of the forward and backward and the average-over-mention representation, while each word in the query is the output of the final layer of the bi-directional RNN and the representation for the whole query is the concatenation of those outputs.
3. How to reason in the graph? Use the Entity-GCN which is a gated relational graph convolutional neural network to aggregate the information through the edges and time to get the final representation for each node. Then use the softmax to get the probability for each candidate in the answer set, using the max of the affine transformation of all

mentions for each candidate and the query representation.(Note the all the mention words are the same as those in the candidate.)

How:

Dataset: WiKiHOP dataset

Personal Thought:

Pros:

1. The paper used GNN to solve the co-occurrence reasoning between different documents.
2. The paper gave us a new view about how to extract the node and the relationships between them from the supporting documents. It seems simpler since they don't need to represent every word in the document in the graph.
3. They get the multi-hop reasoning, which is via several time step information aggregation of GCN.

Cons:

1. This method seems like can only solve the problem of co-reference. For the broader problem, such like asking the relationship between two entities are impossible to answer by this method.
2. This method used the exact matching to construct the graph for supporting documents, so it cannot deal with the pronouns and different forms of the same word. We can see that from the performance of masked is better than that of unmasked.
3. As the paper itself said, we can explore more powerful tools, such like graph attention networks.
4. Why if they use the fully connected graph and gave different types of the edges will get the best performance. I think it is inefficient to construct the fully connected graph for any documents and even impossible if the edge represents for more abstract relationship other than the co-occurrence??
5. How to get the abstract relationship into the representation of the graph edge??
6. When their model answers the question, it must get the candidate answer, what if give it supporting documents, and the open question without any candidate??
7. The algorithm cannot give the reasoning and supporting sentence by itself as required by the HOTPOTQA dataset.

Modeling Text with Graph Convolutional Network for Cross-Modal Information Retrieval

This article is actually jointly train two parallel paths for extracting the features from the image and the text. Since our task does not include the image and retrieval, I only summarize how they use GCN to construct the model to get the text graph features.

How:

1. How to construct the graph from the corpus? For a set of text documents, they first get the a set of unique words in them, then use the pre-trained word2vector to represent each word in it. For the graph, they set 1 to the edges connecting to the node's k-neighbours, where k-neighbours is computed using the cosine similarity between word2vector embedding. For a specific document, it is a bag-of-word vector, and each

element for the node is the frequency of that word appearing in this document. So now, for some document, the property of the node is the frequency that word appearing in the document, and the property of the edge is the cosine similarity of word2vec embedding of different words.

2. How to get the graph representation of the input text document? Use the text GCN model, which is composed of two layers of graph convolutions, each follows by ReLU and a fully-connected layer as the last layer.(Of course this fully-connected layer is for getting the entire text feature, then used for getting the similarity of that of images. However, for our task, we don't need this?? Maybe we only need to get the graph representation for each node??)

Variational Knowledge Graph Reasoning

Why:

1. They are interested in reasoning and inference the relationship given two entities.
2. They frame the KG reasoning task as "path-finding" and "path-reasoning", the current state-of-the-art methods only focus on one of these tasks, which makes them less robust to the noise and adversarial samples.

What:

This paper treats the relation inference problem as classification problem.

Since in the log loss function, we get the latent variable (reasoning path), therefore, we used ELBO as the lower bound of it, then try to minimize it. ELBO includes 1. Likelihood, prior, and posterior.

1. Likelihood(Path Reasoner) P(r|L):

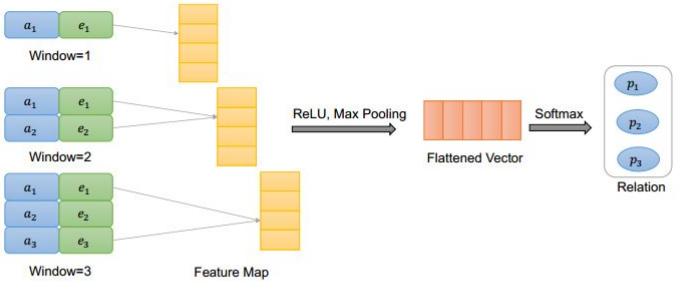

Figure 2: Overview of the CNN Path Reasoner.

They used the CNN to get the relation type from the embedding of intermediate entities and paths.

Exploring Graph-structured Passage Representation for Multi-hop Reading Comprehension with Graph Neural Networks

Why:

1. Multi-hop reasoning attracts less attentions nowadays

2. Previous work constructing the graph from text only relied on one kind of relationship(coreference) between entities.

What:

Generating Graph from the documents:

They divided three types of relationship between mentioned entities:

1. Same
2. Window
3. Coreference

Note in this work they solve the problem of pronouns, actually, they has the similar method as the previous one.

Integrating information from the graph:

Use the embedding got from the last step as the initial state of the graph, then use GRN or GCN to get the hidden states for each entity in each time step.

Getting the probability of being the answer:

Like in the baseline, they used the attention-like mechanism, use the embeddings for each time got from the last step, then concatenate them together combining the representation of the question to calculate the attention probability for each answer.

How:

Dataset: WiKiHop, ComplexWebQuestions

Apart from the models proposed in the paper, they also got two baseline models.

1. Local: concatenate relevant passages, then use BiLSTM to embed them from the original embedding of each word(like Glove) to capture the context. Then extract all the mention entities representation from it using hidden states of its start and end positions. Also, use the same method to embed the question. (Note here, for the passages and questions, LSTM only embed for the whole picture instead of a single word). FInally, they use attention to get the probability of a mentioned entity being the answer.
2. Coref LSTM: same as above, except that they use DAG LSTM instead of BiLSTM to embed mention entities.

Stanford CoreNLP: obtain coreference and NER annotations.

Glove: as the initial word embedding.

Personal Thinking:

1. This paper has the similar idea as **Question Answering by Reasoning Across Documents with Graph Convolutional Networks,** but it deals with the problem of pronouns.
2. This graph extracted from the text cannot represent more abstract relationships, the only relationship they represent is hand-crafted ones. Can we use the NN to learn which relationship between two entities, and whether there is a relationship between them(whether to connect them)?
3. It seems like GNN can only get the representation of every node. How to use it to get the supporting facts as required in HOTPOTQA?? Use Attention??

4. Since the shortcoming of the dataset they use, the answer must be an entity. So it cannot answer the question like comparison and yes/no questions which require more advanced reasoning ability.