shagunsodhani / **End-to-end optimization of goal-driven and visually grounded dialogue systems.md**

Created 5 years ago

Notes for paper "End-to-end optimization of goal-driven and visually grounded dialogue systems"

End-to-end optimization of goal-driven and visually grounded dialogue systems.md

End-to-end optimization of goal-driven and visually grounded dialogue systems

Introduction

- The paper introduces an architecture for end-to-end Reinforcement Learning (RL) optimization for task-oriented dialogue systems and its application to a multimodal task - grounding the dialogue into a visual context.

Encoder Decoder Models vs RL Models

- Encoder Decoder models do not account for the planning problems (which are inherent in the dialogue systems) and do not integrate seamlessly with external contexts or knowledge bases.
- RL models can handle the planning problem but require online learning and a predefined structure of the task.

Dataset

- Corpus of data collected for GuessWhat?! game
- The goal of the game is to locate an unknown object in a rich image scene by asking a series of questions.

- The authors use their previous work in GuessWhat?! paper to build a supervised agent and a neural training environment.
- The agent and the environment are used to train a Deep RL agent online which can solve the task.
- Link to summary of GuessWhat?! paper

Training Environment

- **Oracle**
    - Given an image and a question about the image, the **oracle** can reply in "yes" or "no" or "not applicable".
- **Questioner**
    - Given an image, and a list of previous question&answers (if applicable), the **questioner** generates a question with the aim of locating the object in the image.
- **Guesser**
    - Once the **questioner** is confident of having identified the image, the **oracle** presents a list of objects to the **guesser** to choose from.
- These components are based on the previous work by the authors where they develop nueral baselines for these components.

GuessWhat!? as a Markov Decision Process

- The state of the system is defined as the set of:
    - list of words/tokens generated so far in the current question
    - list of questions and answers so far
    - input image
- The action space comprises of set of all the words in the vocabulary (5K in this context).
- Transition Graph
    - if `<stop>` word is choosen as the action, the full dialogue is terminated.
    - if `<?>` word is choosen as the action, the current question is considered completed and the answer is sampled from the **oracle**.
    - if any other word is chosen as the action, it is used to update the state of the systema and the process continues.

- Reward is defined for every state-action pair.
- The dialogue is automatically terminated after J_{max} questions while the questions are automatically terminated after I_{max} words.
- The game can be modelled as an episodic RL scenario where **generator** can be trained using Policy Gradient methods.
- The paper defines the reward as 1 if the **guesser** could identify the correct object and 0 otherwise.
- Even though MDP assumption of "reward being a function of state and action" does not hold, policy gradient method employed by the paper (called as REINFORCE) is still applicable if the MDP is partially observable.

Training Process

- Train the **oracle**, the **questioner** and the **guesser** independently.
- Finetune the trained **questioner** using the proposed RL framework.

Results

- On test set, the baseline approach obtained 45% accuracy while the paper reports 53% accuracy.
- Beam search baseline (case of supervised encoder-decoder model) tends to repeat questions - an indication of poor generalization.
- Beam search basline also tends to generate longer sentences and incoherent sequence of questions (indicating towards the lack of planning component).
- RL trained model favours enumerating object categories and spatial information.
- RL trained model tends to generate shorter dialogues and uses a smaller set of vocabulary.
    - One player, called as the **oracle**, is randomly assigned an object in the given image.
- The second player, called as the **questioner**, tries to locate the object, given just the image.
- The **questioner** can ask a series of questions about the object and the **oracle** can reply in "yes" or "no" or "not applicable".

- Once the **questioner** is confident of having identified the image, the **oracle** presents a list of objects to the **questioner** to choose from.

- A small penalty is added, every time a question is asked, so as to encourage informative questions only.

GuessWhat?! Game

- One player, called as the **oracle**, is randomly assigned an object in the given image.
- The second player, called as the **questioner**, tries to locate the object, given just the image.
- The **questioner** can ask a series of questions about the object and the **oracle** can reply in "yes" or "no" or "not applicable".
- Once the **questioner** is confident of having identified the image, the **oracle** presents a list of objects to the **questioner** to choose from.
- A small penalty is added, every time a question is asked, so as to encourage informative questions only.

Dataset

- A filtered subset of images from MSCOCO is used as the image set.
- Two separate tasks create on Amazon Mechanical Turk (AMT) - for the role of **oracle** and **questioner**.
- Data was post processed -- both manually and using AMT -- to account for things like spelling mistakes and validation.
- Final dataset comprises of 150K thousand human game iterations with 800K question-answer pairs on 60K images.
- Dataset is available at https://guesswhat.ai/download

Interesting Observations

- an average number of questions, given the number of objects in the image, grows at a rate between logarithmic and linear possibly because:
  - **questioner** does not have access to the list of images while asking the question.
  - **questioner** might ask a few extra questions just to be sure.
- **questioner** uses abstract object properties like "human/object/furniture" early in the conversation and quickly switch over to spatial/visual aspects like "left/right or table/chair"/

- The paper also includes the analysis of how factors like size of the unknown object, its position, total number of objects etc affect the accuracy of humans (playing on AMT).

Model# End-to-end optimization of goal-driven and visually grounded dialogue systems

Introduction

- The paper introduces an architecture for end-to-end Reinforcement Learning (RL) optimization for task-oriented dialogue systems and its application to a multimodal task - grounding the dialogue into a visual context.

Encoder Decoder Models vs RL Models

- Encoder Decoder models do not account for the planning problems (which are inherent in the dialogue systems) and do not integrate seamlessly with external contexts or knowledge bases.
- RL models can handle the planning problem but require online learning and a predefined structure of the task.

Dataset

- Corpus of data collected for GuessWhat?! game
- The goal of the game is to locate an unknown object in a rich image scene by asking a series of questions.
- The authors use their previous work in GuessWhat?! paper to build a supervised agent and a neural training environment.
- The agent and the environment are used to train a Deep RL agent online which can solve the task.
- Link to summary of GuessWhat?! paper

Training Environment

- **Oracle**
    - Given an image and a question about the image, the **oracle** can reply in "yes" or "no" or "not applicable".
- **Questioner**

- Given an image, and a list of previous question&answers (if applicable), the **questioner** generates a question with the aim of locating the object in the image.

- **Guesser**
  - Once the **questioner** is confident of having identified the image, the **oracle** presents a list of objects to the **guesser** to choose from.

- These components are based on the previous work by the authors where they develop nueral baselines for these components.

GuessWhat!? as a Markov Decision Process

- The state of the system is defined as the set of:
  - list of words/tokens generated so far in the current question
  - list of questions and answers so far
  - input image

- The action space comprises of set of all the words in the vocabulary (5K in this context).

- Transition Graph
  - if `<stop>` word is choosen as the action, the full dialogue is terminated.
  - if `<?>` word is choosen as the action, the current question is considered completed and the answer is sampled from the **oracle**.
  - if any other word is chosen as the action, it is used to update the state of the systema and the process continues.

- Reward is defined for every state-action pair.

- The dialogue is automatically terminated after J_{max} questions while the questions are automatically terminated after I_{max} words.

- The game can be modelled as an episodic RL scenario where **generator** can be trained using Policy Gradient methods.

- The paper defines the reward as 1 if the **guesser** could identify the correct object and 0 otherwise.

- Even though MDP assumption of "reward being a function of state and action" does not hold, policy gradient method employed by the paper (called as REINFORCE) is still applicable if the MDP is partially observable.

Training Process

- Train the **oracle**, the **questioner** and the **guesser** independently.
- Finetune the trained **questioner** using the proposed RL framework.

Results

- On test set, the baseline approach obtained 45% accuracy while the paper reports 53% accuracy.
- Beam search baseline (case of supervised encoder-decoder model) tends to repeat questions - an indication of poor generalization.
- Beam search basline also tends to generate longer sentences and incoherent sequence of questions (indicating towards the lack of planning component).
- RL trained model favours enumerating object categories and spatial information.
- RL trained model tends to generate shorter dialogues and uses a smaller set of vocabulary.
    - One player, called as the **oracle**, is randomly assigned an object in the given image.
- The second player, called as the **questioner**, tries to locate the object, given just the image.
- The **questioner** can ask a series of questions about the object and the **oracle** can reply in "yes" or "no" or "not applicable".
- Once the **questioner** is confident of having identified the image, the **oracle** presents a list of objects to the **questioner** to choose from.
- A small penalty is added, every time a question is asked, so as to encourage informative questions only.

GuessWhat?! Game

- One player, called as the **oracle**, is randomly assigned an object in the given image.
- The second player, called as the **questioner**, tries to locate the object, given just the image.
- The **questioner** can ask a series of questions about the object and the **oracle** can reply in "yes" or "no" or "not applicable".

- Once the **questioner** is confident of having identified the image, the **oracle** presents a list of objects to the **questioner** to choose from.
- A small penalty is added, every time a question is asked, so as to encourage informative questions only.

Dataset

- A filtered subset of images from MSCOCO is used as the image set.
- Two separate tasks create on Amazon Mechanical Turk (AMT) - for the role of **oracle** and **questioner**.
- Data was post processed -- both manually and using AMT -- to account for things like spelling mistakes and validation.
- Final dataset comprises of 150K thousand human game iterations with 800K question-answer pairs on 60K images.
- Dataset is available at https://guesswhat.ai/download

Interesting Observations

- an average number of questions, given the number of objects in the image, grows at a rate between logarithmic and linear possibly because:
  - **questioner** does not have access to the list of images while asking the question.
  - **questioner** might ask a few extra questions just to be sure.
- **questioner** uses abstract object properties like "human/object/furniture" early in the conversation and quickly switch over to spatial/visual aspects like "left/right or table/chair"/
- The paper also includes the analysis of how factors like size of the unknown object, its position, total number of objects etc affect the accuracy of humans (playing on AMT).

Model

- Given an image, a set of segments objects (along with their category and pixel-wise segmentation mask) and the object to be identified.
- The **questioner** generates a series of questions to ask from the **oracle**

Oracle

- Modelled as a single hidden layer MLP.
- **Input**: Concatenate embeddings for the
  - image - obtained using FC8 features from VGG Net - fixed during training

  - cropped target object - obtained using FC8 features from VGG Net - fixed during training
  - spatial information about the target object (bounding box coordinates, normalised wrt coordinates of the centre) in form of a vector - fixed during training
  - category of target object - dense categorical embedding - trained
  - current question asked by the **questioner** - encoded by LSTM - trained
- **Output**: One of the three answers - "yes", "no", "not applicable"
- **Loss**: Negative log-likelihood of correct answer
- **Performance**: The best model achieves the test error of 21.5% and uses question embedding + category embedding + spatial information.

Questioner

- Question performs two sub tasks:
  - **Guesser**
    - **Input**: Concatenate embeddings for the
      - image - obtained using FC8 features from VGG Net - fixed during training
      - dialogue - the series of question-answer pair are embedded into fixed size vectors using an LSTM or HRED encoder
    - Objects are represented by:
      - concatenation of their category and spatial features
      - passing through an MLP (which shares parameters across objects)
    - Perform a dot product between input embedding and embedding for the objects, followed by softmax, to obtain the probability distribution of the objects.
    - **Performance**:
      - The best model achieves the test error of 38.7% and uses LSTM alone (no VGG features were used)
      - VGG features did not improve the performance, probably because the questions and the objects captured all the information already
      - Maybe using some different network for image features may help
  - **Generator**
    - HRED, conditioned over the VGG features (from the image) and questions asked so far (if any) is used to generate the natural language questions by maximising conditional log-likelihood.

    - The questions generated by the **generator** are answered by the **oracle** which is ground truth at train time and trained oracle at test time. The paper acknowledges this shortcoming.
    - 5 questions are generated before triggering the **guesser**
    - **Performance**: The best model achieves an error of 38.7% using human generated dialogues.
    - the performance is also deteriorated by the fact that **oracle** and **guesser** are not perfect.

Comments

- The paper has provided a very detailed analysis of the dataset and have experimented with various combinations to find the best embedding model.

- The sample questions show in the paper indicates that the **generator** model produces the same question many times - indicating poor generalisation.

- Given an image, a set of segments objects (along with their category and pixel-wise segmentation mask) and the object to be identified.

- The **questioner** generates a series of questions to ask from the **oracle**

Oracle

- Modelled as a single hidden layer MLP.
- **Input**: Concatenate embeddings for the
  - image - obtained using FC8 features from VGG Net - fixed during training
  - cropped target object - obtained using FC8 features from VGG Net - fixed during training
  - spatial information about the target object (bounding box coordinates, normalised wrt coordinates of the centre) in form of a vector - fixed during training
  - category of target object - dense categorical embedding - trained
  - current question asked by the **questioner** - encoded by LSTM - trained
- **Output**: One of the three answers - "yes", "no", "not applicable"
- **Loss**: Negative log-likelihood of correct answer
- **Performance**: The best model achieves the test error of 21.5% and uses question embedding + category embedding + spatial information.

Questioner

- Question performs two sub tasks:

- **Guesser**
    - **Input**: Concatenate embeddings for the
        - image - obtained using FC8 features from VGG Net - fixed during training
        - dialogue - the series of question-answer pair are embedded into fixed size vectors using an LSTM or HRED encoder
    - Objects are represented by:
        - concatenation of their category and spatial features
        - passing through an MLP (which shares parameters across objects)
    - Perform a dot product between input embedding and embedding for the objects, followed by softmax, to obtain the probability distribution of the objects.
    - **Performance**:
        - The best model achieves the test error of 38.7% and uses LSTM alone (no VGG features were used)
        - VGG features did not improve the performance, probably because the questions and the objects captured all the information already
        - Maybe using some different network for image features may help
- **Generator**
    - HRED, conditioned over the VGG features (from the image) and questions asked so far (if any) is used to generate the natural language questions by maximising conditional log-likelihood.
    - The questions generated by the **generator** are answered by the **oracle** which is ground truth at train time and trained oracle at test time. The paper acknowledges this shortcoming.
    - 5 questions are generated before triggering the **guesser**
    - **Performance**: The best model achieves an error of 38.7% using human generated dialogues.
    - the performance is also deteriorated by the fact that **oracle** and **guesser** are not perfect.

Comments

- The paper has provided a very detailed analysis of the dataset and have experimented with various combinations to find the best embedding model.
- The sample questions show in the paper indicates that the **generator** model produces the same question many times - indicating poor generalisation.