seq2graph: A Neural Approach to Scene Graph Generation from Natural Language

Frits van Paasschen and Isaac Kasevich

CS 224n
Winter 2019

Motivation & Definition

- **Scene graphs** capture image semantics
- Graphs have a variety of **powerful downstream applications** such as image retrieval and generation
- No **neural pipeline solution** for generating scene graphs from paragraph-level natural language

Problem Statement

- How can we build semantically rich scene graphs from a **paragraph-level description**?
- **Input:** A natural language description of a scene
- **Output:** G = {V, E, A}: a set of node multisets, relationship edges, and node attributes
- **Isomorphism**: many valid ways to represent G

Data

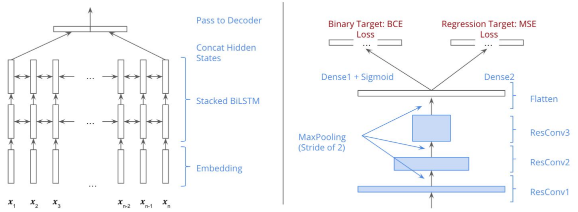
- **Visual Genome**: a collection of dense image annotations, including scene graphs and image descriptions
- **Preprocessing:** translated objects, relationships, and attributes to **synsets** (collection of synonyms)
- Training pairs: (paragraph: {objects, relationships})

- **Paragraph:** *"A time clock hangs on the wall in the center of the image. A silver object is sitting on top of it. On either side of the clock hangs grey time card holders. Each slot is number. Below the time clock, on a shelf is a white hard hat with a black and grey chin strap. Under the helmet is a file folder.*
- **Object SynSet Multiset:** *{time_clock.n.01, wall.n.01, clock.n.01, glass.n.01, numeral.n.01, shelf.n.01, support.n.01, strap.n.01, paper.n.01, point.n.09, prison_guard.n.01, lock.n.01, mailbox.n.01, pipe.n.01}*
- **Relationship SynSet Multiset:** *{(time_clock.n.01, along.r.01, wall.n.01), (clock.n.01, be.v.01, glass.n.01), (numeral.n.01, along.r.01, panel.n.01), (support.n.01, have.v.01, shelf.n.01), (time_clock.n.01, have.v.01, point.n.09), (prison_guard.n.01, be.v.01, time_clock.n.01), (support.n.01, have.v.01, shelf.n.01), (time_clock.n.01, be.v.01, wall.n.01), (point.n.09, be.v.01, time_clock.n.01), (time_clock.n.01, have.v.01, lock.n.01), (shelf.n.01, be.v.01, wall.n.01), (support.n.01, be.v.01, wall.n.01), (time_clock.n.01, have.v.01, lock.n.01), (time_clock.n.01, have.v.01, point.n.09), (mailbox.n.01, have.v.01, numeral.n.01), (clock.n.01, be.v.01, time_clock.n.01), (prison_guard.n.01, be.v.01, time_clock.n.01), (pipe.n.01, be.v.01, time_clock.n.01), (clock.n.01, be.v.01, wall.n.01), (numeral.n.01, along.r.01, wall.n.01), (support.n.01, be.v.01, wall.n.01)}*

Approaches

- Experimented with **various novel architectures/models**
- **One-shot Multiset Prediction:** avoid set isomorphism
- **Multitask Learning**: predict nodes and set cardinality
- **Dependency Parsing**: parse input into dependency graphs and **align** predicted nodes and edges
- **Neural Decoders**: use **sequence models** to build graph

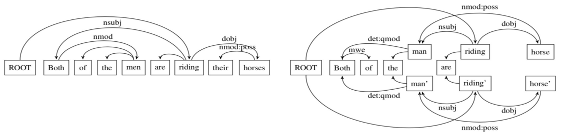
Multitask Architecture

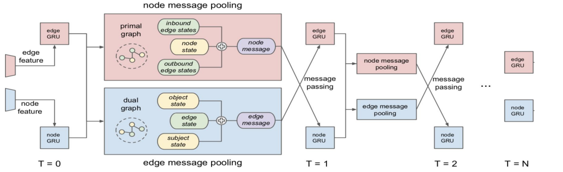


$$\ell(\hat{y}_b, y_b, \hat{y}_m, y_m) = \lambda \ell_{BCE}(\hat{y}_b, y_b) + (1-\lambda)\ell_{MSE}(\hat{y}_m, y_m)$$

$$= \frac{1}{|N|}\left[\lambda \sum_i \left(y_b^{[i]} \log(\hat{y}_b^{[i]}) + (1 - y_b^{[i]}) \log(1 - \hat{y}_b^{[i]})\right) + (1-\lambda)||y_m - \hat{y}_m||_2\right]$$

End-to-End Neural Pipeline





Experiments and Results

- Evaluated model performance using **F1 Score** on predicted edges/nodes and **multiset cardinality**

Input Paragraph: *This is an image of a sporting event. The woman is playing tennis. The woman is holding a tennis ball. The ball is light green. The woman is about to serve the ball The girl is holding a tennis racket. The girl is wearing a shirt. The shirt is white. The shirt has a design of a bird on it. The girl has on black shorts. The racket is orange and black.*
Predicted Object SynSets: *[shoe.n.01, line.n.01, ball.n.01, shirt.n.01, court.n.01, short_pants.n.01, leg.n.01, woman.n.01, racket.n.04]*

Sentence: *The other elephants are in the forest.*
Output: *Candidate Objects: [elephants, forest], Alignments: [elephants:elephant.n.01, forest:land.n.01], Relationships: [(elephant,along.v.01,forest),(forest,behind.v.01,elephants)]*

Encoder	Decoder/Predictor	F1 (%)	Accuracy (%)
2-Layer Bi-LSTM, $d = 128$	2-Layer Dense	27.8	24.62
3-Layer Bi-LSTM, $d = 256$	2-Layer Dense	34.1	32.21
2-Layer Bi-LSTM, $d = 128$	Multitask Conv (3x Maxpool)	29.6	21.11
2-Layer Bi-LSTM, $d = 128$	Multitask Conv (2x Maxpool)	31.3	21.84
3-Layer Bi-LSTM, $d = 256$	3-Layer Dense	33.4	31.04
3-Layer Bi-LSTM, $d = 256$	Multitask Conv (2x Maxpool)	34.3	33.17
Dependency Parse+Align	2 Layer Dense $d = 128$ (object)	–	0.059
Dep. Parse+Align+BiLSTM $d = 128$	2 Layer Dense $d = 128$ (edge)	–	0.061

Conclusions and Future Work

- Presented **novel architecture** for scene graph generation from natural language
- **Computational limitations** prevented full architecture test
- **Future Work:**
  - Completely implement proposed **GRU with message passing**
  - Explore algorithms for merging sentence-level scene graphs into **paragraph-level** scene graphs

$$\text{[node message]: } m_i = \sum_{j:i\to j} \sigma(\mathbf{w}_1^T[h_i; h_{i\to j}]) + \sum_{j:j\to i} \sigma(\mathbf{w}_2^T[h_i; h_{j\to i}])$$

$$\text{[edge message]: } m_{i\to j} = \sigma(\mathbf{v}_1^T[h_i; h_{i\to j}]) + \sigma(\mathbf{v}_2^T[h_i; h_{j\to i}])$$