Learning to Generate from Fact Representations

Rajarshi Bhowmik, Gerard de Melo

Rutgers University. New Brunswick, NJ, USA

gdm@demelo.org http://gerard.demelo.org

Goal

Spouse	Mirka Federer
Height	185 cm
Country of Citizenship	Switzerland
Date of Birth	8 August 1981
Place of Birth	Basel
Occupation	Tennis Player

Swiss Tennis Player born in Basel

Research Question: Recent machine learning algorithms are very successful at generating text outputs from text inputs. Can we generate text outputs from database facts instead?

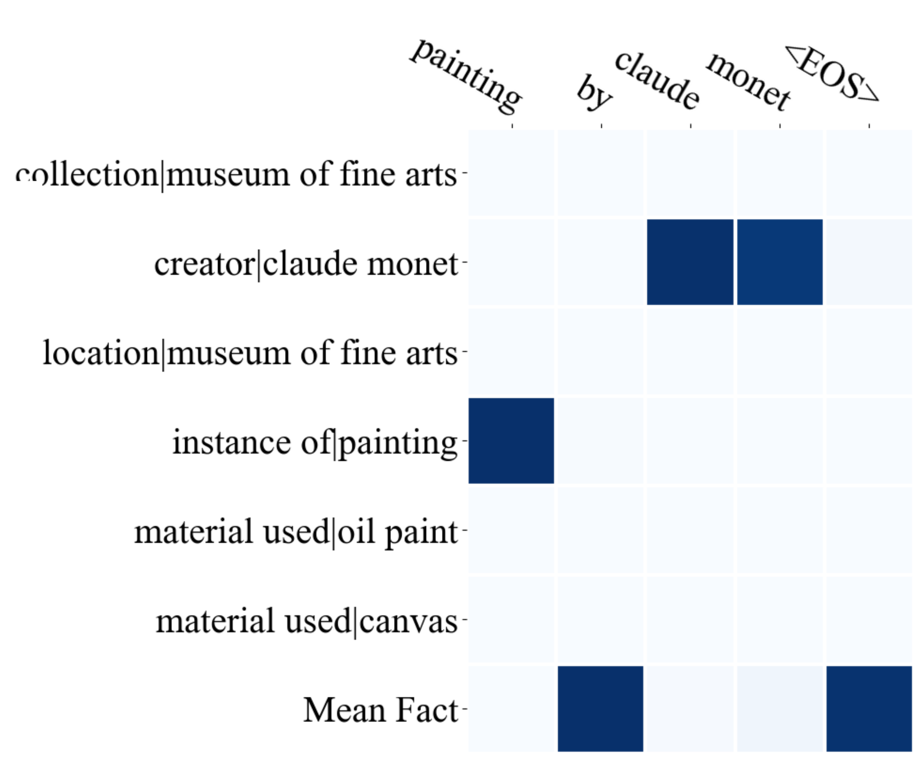
Key Challenge: Some output words (e.g. "Basel") may not have been observed in training data. Regular models cannot be applied on demand to generate new words that have never been seen before.

Solution: For on-demand generation of new words, we design a new deep neural architecture that can flexibly incorporate arbitrary items from the fact representations into the output.

Analysis

Visualization of Attention Component in model:

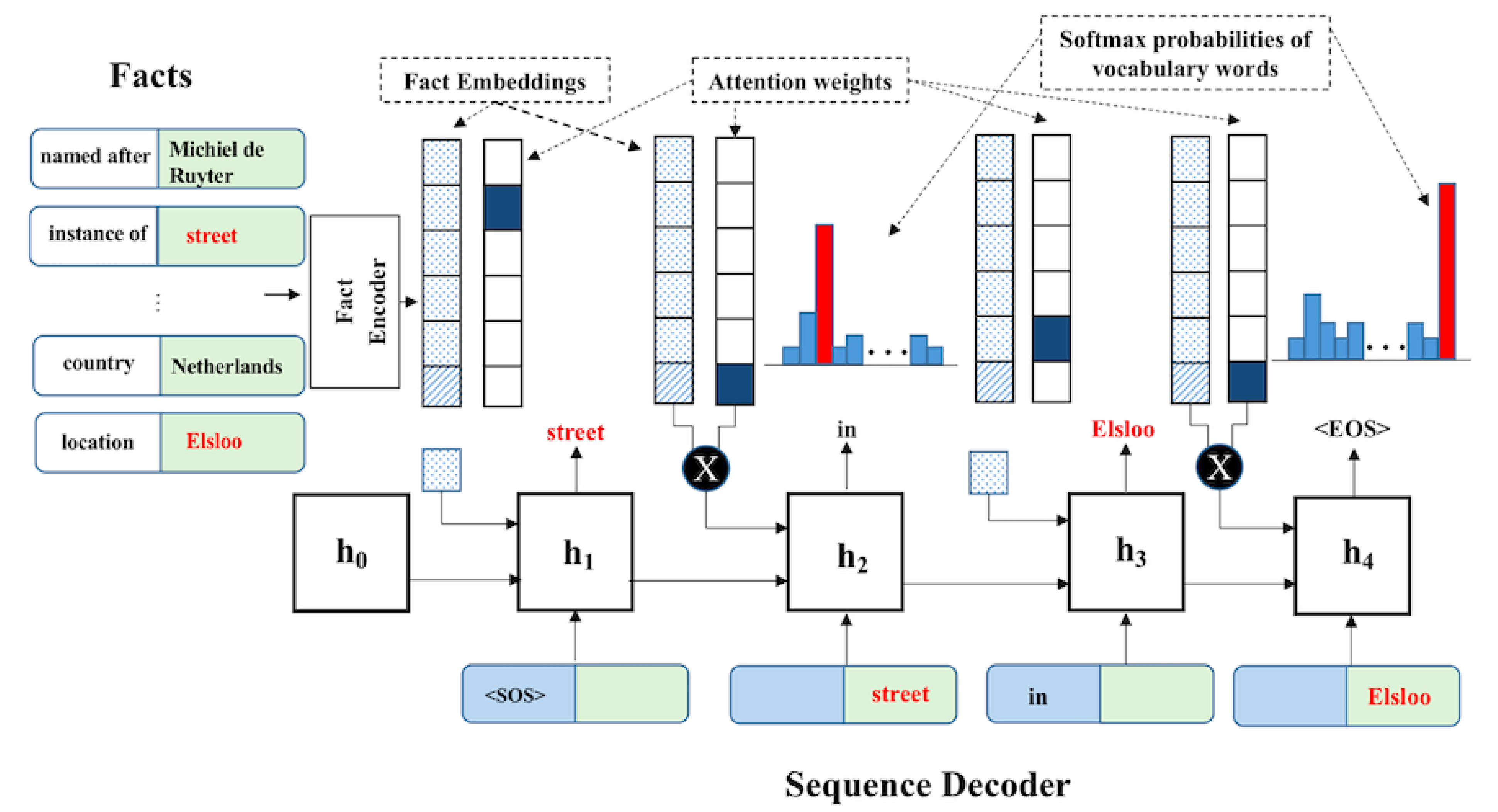
Our model pays attention to the right facts while generating each output word



Item	Instance of	Generated Description
Q11584386	Human	japanese tarento
Q2198428	Human	netherlands businessperson
Q3260917	Human	french military personnel
Q1494733	Painting	painting by august macke
Q16054316	Painting	painting by liselotte schramm-heckmann
Q15880468	Painting	painting by emile wauters
Q10288648	Book	book by izomar camargo guilherme
Q10270545	Book	novel by antonin kratochvil
Q10272202	Book	novel by jose louzeiro
Q1001786	Street	street in budapest
Q10552208	Street	street in orebro
Q10570752	Street	street in malmo municipality

Examples of generated descriptions for Wikidata entities with missing descriptions

Model



Fact Encoder

- **Input:** set of N facts $\{f_1, f_2, \ldots, f_N\}$
- **Output:** concatenation of Fact Embeddings $[\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N]$
- Learn Fact Embeddings using ***Word Embeddings*** + ***Positional Encoder***
- Positional Encoder: $\boldsymbol{f_i} = \sum_{j=1}^{J} \boldsymbol{l_j} \circ \boldsymbol{w^i_j}$
- Mean Fact: $\boldsymbol{f_{N+1}} = \frac{1}{N}\sum \boldsymbol{f_i}$

Sequence Decoder

- We use Gated Recurrent Unit (GRU) as a sequence decoder
- **Fact selection:** At each timestep *t*, the decoder chooses a fact from the set of *N + 1* facts

$$\mathbf{e}_i = \mathbf{W}_2 \tanh(\mathbf{W}_1[\mathbf{f}_i; \mathbf{h}_{t-1}]) \; \forall i \in \{1, N+1\}, \qquad f_t = \underset{i \in \{1,\ldots,N+1\}}{\arg\max} \; P(f = f_i \mid \mathbf{f}_i, \mathbf{h}_{t-1})$$

$$P(f = f_i \mid \mathbf{f}_i, \mathbf{h}_{t-1}) = \frac{\exp(\mathbf{e}_i)}{\sum_{i' \in \{1,N+1\}} \exp(\mathbf{e}_{i'})}, \qquad \mathbf{f}_t = \mathbf{f}_{f_t}$$

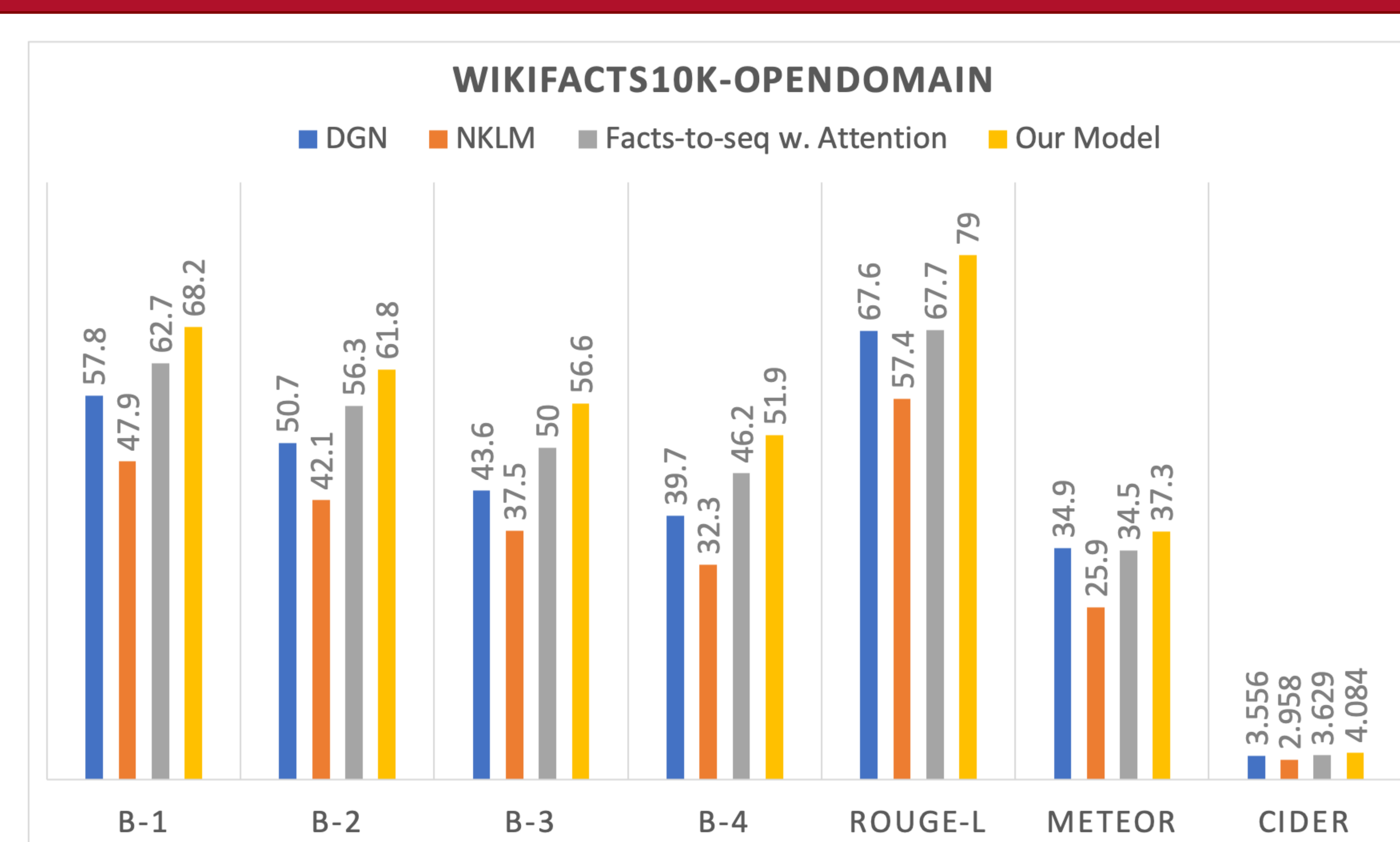
- **Generating vocabulary words:** If the selected fact is the Mean Fact, the model generates a non-factual vocabulary word

$$\mathbf{o}_t = \mathbf{W}_a \mathrm{ReLU}(\mathbf{W}_b[\mathbf{c}_t; \mathbf{h}_t]),$$
$$P(w \mid \mathbf{c}_t, \mathbf{h}_t) = \mathrm{Softmax}(\mathbf{o}_t)$$
$$w_t = \underset{w \in \mathcal{V}}{\arg\max} \; P(w \mid \mathbf{c}_t, \mathbf{h}_t)$$

- **Copying factual words:** If a fact other than the Mean Fact is selected, the decoder copies a word from the corresponding fact to the output sequence. The position of the word to copy is determined as follows:

$$\mathbf{r}_t = \mathbf{W}_c \mathrm{ReLU}(\mathbf{W}_d[\mathbf{f}_t; \mathbf{h}_t])$$
$$P(n \mid \mathbf{f}_t, \mathbf{h}_t) = \mathrm{Softmax}(\mathbf{r}_t)$$
$$n_t = \underset{n \in \{1,\ldots,|\mathcal{V}_{f_t}|\}}{\arg\max} \; P(n \mid \mathbf{f}_t, \mathbf{h}_t)$$

Results



RUTGERS

Code and Data

https://github.com/kingsaint/Wikidata-Descriptions