Unsupervised Text Generation from Structured Data

Martin Schmitt  
Center for Information and Language Processing (CIS)  
LMU Munich  
martin@cis.lmu.de

Hinrich Schütze  
Center for Information and Language Processing (CIS)  
LMU Munich

Abstract

This work presents a joint solution to two challenging tasks: text generation from data and open information extraction. We propose to model both tasks as sequence-to-sequence translation problems and thus construct a joint neural model for both. Our experiments on knowledge graphs from Visual Genome, i.e., structured image analyses, shows promising results compared to strong baselines. Building on recent work on unsupervised machine translation, we report the first results – to the best of our knowledge – on fully unsupervised text generation from structured data.

1 Introduction

Knowledge graphs (KGs) are a general-purpose method for storing information (knowledge) in a structured, machine-accessible way and as such are an important component of the semantic web (Van Harmelen et al., 2008). So it is no wonder that they were also adapted in other domains such as image analysis (Lu et al., 2016; Krishna et al., 2016). Very often such an analysis (also called scene graph) is sufficient to make an image more accessible to both human and machine.

A KG is human-interpretable in principle, but non-experts may have difficulty making sense of raw triples. Thus, there is a need for methods such as an automatic natural language summarization that support non-experts working with KGs. This paper addresses both directions of conversion between KGs and text to both help humans better understand graphs and machines better understand texts (i.e., information extraction).

Our contributions are in particular:

1. We propose a joint model for data to text generation and open information extraction.
2. We obtain first results in a fully unsupervised setting for the above tasks.
3. We are the first to attempt image caption generation from a structured analysis of an image without using the actual image.

2 Task Formalization

2.1 The Task

Visual Genome (Krishna et al., 2016) is a large collection of images with associated scene graphs, where a scene graph annotates visual objects with properties and visual relations between them. Each scene graph is organized into smaller so-called region graphs, representing a subpart of a

Gold a girl holding a purple tennis racket
Graph2text racket is purple and racket is for tennis and girl holding racket
Text2Graph (girl, holding, tennis); (tennis, attr, purple)

Figure 1: Example scene graph and output generated by the rule-based systems.

For tennis racket purple attribute girl holding attribute for tennis purple attribute

for tennis racket
attribute
purple
attribute
girl
holding

1. We propose a joint model for data to text generation and open information extraction.
2. We obtain first results in a fully unsupervised setting for the above tasks.

more complex larger picture that is interesting on its own. Each region graph is associated with a textual region description. The region graphs were not specifically designed to resemble the region descriptions although they should describe the same region in an image. This semantic correspondence makes it an interesting problem for data to text generation, while it is still challenging because text and graph are not simple translations of each other.

Generating the description from the corresponding region graph can be seen as an instance of concept-to-text generation whereas the opposite direction can be called open information extraction (Niklaus et al., 2018).

## 2.2 The Data

The graphs from Visual Genome consist of objects (as nodes $V$), relations (as directed edges between these nodes $E \subseteq V \times V$) and attributes (of one node). In order to fit with standard definitions in graph theory, we formalize attributes as nodes as well and link them to their respective nodes with edges bearing the special label `attribute`. The other edges and nodes are labeled as well. Edge labels are called predicates, node labels are called object names.

Knowledge graphs (KGs) are often considered sets of facts rather than edges with labels. A fact is simply a triple consisting of an edge’s start node label (the subject), the edge label, and the end node label (the object). We call the sequence of facts of a KG the serialization of a KG. We serialize a KG by separating the elements of a fact triple with special `SEP` symbols and separating serialized facts with end-of-fact symbols (`EOF`).

The serialization of a KG defines an order for its facts although in reality the triples in a graph are unordered. However, working with serializations permits us to use the same sequence models for text and KGs, which offers a lot of advantages – e.g., we can embed both in the same semantic space using a sequence encoder.

### 2.3 Notation

In the remainder of the paper, we denote the space of textual region descriptions and region graphs by $T$ and $G$. The set of available supervised examples $(x, y) \in G \times T$ is called $S \subset G \times T$. $P_{a \rightarrow b}$ will stand for translation models expecting an $a$ as input and producing a $b$ as output where $g$ (graph) and $t$ (text) are possible values for $a$ and $b$.

## 3 Models

### 3.1 Rule-based systems

We propose two simple rule-based systems that rely on the similarities between KG representations and actual sentences. These two systems will serve as baselines in our experiments.

**Graph2text.** From a serialized KG, we simply remove separator symbols and replace end-of-fact symbols by the word `and`. The special predicate `attribute` is translated as `is`. See Fig. 1 for an example.

**Text2graph.** After preprocessing a text using NLTK’s default POS tagger (Loper and Bird, 2002) and removing stop words, we apply two simple heuristics to identify facts in a given text: (1) Each verb is translated to a predicate (only `is` is changed to `attribute`). Subject and object are identified as the content words directly before and after the verb. (2) All adjectives `a` form an attribute, i.e., build triples of the form $(X,\texttt{attribute},a)$. $X$ is filled with the first noun after `a`.

### 3.2 Neural translation systems

For our neural translation systems, we employ the standard encoder-decoder architecture with attention (Bahdanau et al., 2014) augmented with a copy mechanism (Gu et al., 2016). Allowing the model to directly copy from the source to the target side has been shown to be beneficial in data to text generation (Wiseman et al., 2017; Puduppully et al., 2019). We use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) as encoder and a unidirectional one as decoder. Following (Britz et al., 2017), we furthermore apply dropout (Hinton et al., 2012) at the input of both encoder and decoder cells.

In the following, we describe the changes we made to this standard architecture and our design principles to adapt the neural translation architecture to the task of text generation from structured data.

**Same system for both directions.** Following Lample et al. (2018b), we train our system for both directions text ↔ graph, sharing encoder as well as decoder. To tell the decoder which output language should be produced (text or graph), we initialize the cell state of the LSTM decoder with an embedding of the desired target language. The hidden state of the LSTM decoder is still initialized with the last state of the encoder as usual.
| noise function | behavior                                                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| swap           | applies a random permutation $\sigma$ of words/facts with $\forall i \in \{1, \ldots, n\}, |\sigma(i) - i| \leq k$; $k = 3$ for text, $k = +\infty$ for KGs                                                                                                        |
| blank          | replaces each fact/word with a probability of $p_{\text{blank}}$ by a special symbol $\text{blanked}$                                                                                         |
| drop           | removes each fact/word with a probability of $p_{\text{drop}}$                                                                                                                                  |
| repeat         | inserts repetitions with a probability of $p_{\text{repeat}}$ in a sequence of facts/words                                                                                                       |
| rule           | applies the rule-based systems graph2text and text2graph                                                                                                                                             |

Table 1: Noise functions and their behavior on graphs and texts.

**Noisy source samples.** Lample et al. (2018a) introduced denoising auto-encoding as a helpful auxiliary task to train a decoder’s language model and make an encoder robust to noisy input. The training examples for this task consist of a noisy version of a sentence as source and the original sentence as target side. We adapt this idea and propose the following noise functions in the domains of serialized graphs and texts: swap, blank, drop, repeat, rule. Refer to Table 1 for a detailed description of their behavior. swap, blank and drop are direct adaptations from (Lample et al., 2018a), but in our setup facts in serialized graphs take the role of words in text. As order should not matter in graph representations, we drop the locality constraint in the permutation in swap for facts by setting $k = +\infty$. repeat follows the intuition that serialized KGs should not contain repetitions of the same fact. That means denoising samples that were generated by repeat corresponds to detecting and removing redundant information in a graph representation. We use repeat analogously for textual input; in the case of text, repeat mimics a behavior often observed with insufficiently trained neural MT models, i.e., repeating words a model considers important. rule is the only noise function that introduces domain knowledge about the correspondence between English sentences and serialized KG representations. Here the denoising task becomes a real translation task (although with noisy automatically generated source side). In summary, we consider the following loss terms:

$$L_{\text{sup}} = \mathbb{E}_{(x,y) \sim S}[- \log P_{y \rightarrow t}(y|x) - \log P_{t \rightarrow g}(x|y)]$$

(1)

$$L_{\text{noisy sup}} = \mathbb{E}_{(x,y) \sim S}[- \log P_{y \rightarrow t}(y|C(x)) - \log P_{t \rightarrow g}(x|C(y))]$$

(2)

$$L_{\text{lm}} = \mathbb{E}_{x \sim G}[- \log P_{g \rightarrow t}(z|x) + \mathbb{E}_{y \sim T}[- \log P_{t \rightarrow g}(w|y)]$$

(3)

where $C \in \{\text{swap}, \text{blank}, \text{drop}, \text{repeat}, \text{rule}\}$ chosen randomly each epoch for each sample and loss term independently.

**Unsupervised training** As in the work of (Lample et al., 2018b), our setup can also be easily modified for the unsupervised case. We start training our model only with $L_{\text{lm}}$ for one epoch using all noise functions on all unsupervised samples. In subsequent epochs, we go back to using $L_{\text{lm}}$ in its original way, i.e., sample only one type of noise per sample, and use the sum of $L_{\text{lm}}$ and $L_{\text{back}}$ as training signal where

$$L_{\text{back}} = \mathbb{E}_{x \sim G}[- \log P_{t \rightarrow g}(z^*(x))] + \mathbb{E}_{y \sim T}[- \log P_{g \rightarrow t}(w^*(y))]$$

(4)

$$z^*(x) = \arg \max_z P_{g \rightarrow t}(z|x)$$

(5)

$$w^*(y) = \arg \max_w P_{t \rightarrow g}(w|y)$$

(6)

This means we use backtranslation with the current models to obtain a supervised signal in all epochs but the first.

4 Experiments

For our experiments, we only consider region graphs from Visual Genome that are related to some kind of ball sport event. For this, we identify all images where at least one region graph contains at least one fact that mentions an object ending with ball and take all the regions from them. This yields approximately 390k region graphs and descriptions. We randomly split these data into 80% train, 10% validation (val) and 10% test portions, removing in a second step any regions from train that were also found in one of the images from val or test.

4.1 Training details

We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $10^{-4}$, word embeddings of size 300, a hidden size of 250 for LSTMs,
|                          | val  | test |
|--------------------------|------|------|
| Rule-based Graph2text    | 0.344| 0.358|
| CopyNet (Gu et al., 2016)| 0.231| 0.235|
| Combined graph ↔ text    | 0.229| 0.231|
| + noise objectives       | 0.219| 0.221|
| Unsupervised (iteration 0)| 0.087| 0.090|
| Unsupervised (iteration 1)| 0.083| 0.089|
| Unsupervised (iteration 2)| 0.154| 0.160|
| Unsupervised (iteration 3)| 0.167| 0.172|

Table 2: Performance on text generation from graphs measured in BLEU.

a dropout rate of 0.2 and a batch size of 10. We use these same settings for all experiments.

Supervised models are trained for a maximum number of 20 epochs but stopped early when \( L_{\text{sup}} \) does not decrease on the validation set during 10 epochs. All models capable of translating both directions are trained with homogenous batches of one target language at a time.

To facilitate comparison with the supervised setting, we use the same graphs and texts as used in the supervised train portion also for the unsupervised training. We simply ignore the respective gold target side.

4.2 Results

Table 2 shows the results for the text generation task. From the first line we can see that the rule-based approach is already a very strong baseline according to BLEU. When we look at the example translation in Fig. 1, however, we see how cumbersome these translations become especially for larger knowledge graphs.

The combined approach for both translation directions (graph ↔ text) performs on par with the unidirectional one (CopyNet) while using the same amount of parameters for two tasks instead of one. If we look at the task of information extraction (Table 3), we see a similar picture. The combined approach performs only slightly worse than the unidirectional one, showing that it is a promising direction for future research. The rule-based approach here is consistently outperformed by the neural models, especially in terms of recall.

The unsupervised models achieve a performance already very close to the supervised ones, which shows the general adequateness of the approach for data to text generation. From both tables we see, however, that performance does not always increase from one iteration to the next. These fluctuations in performance should be investigated in future work.

Although our noise functions seem to be adequate to facilitate unsupervised learning, they do not lead to improvements in the supervised case. Further research might shed light on the impact of different types of noise.

5 Related Work

There is a large body of literature about text generation from structured data, notably about the creation of sports game summaries from statistical records (Robin, 1995; Tanaka-Ishii et al., 1998). Recent efforts make use of neural encoder-decoder mechanisms (Wiseman et al., 2017; Puduppully et al., 2019). Although text creation from relational databases is related, we specifically address text creation from graph-like structures such as knowledge graphs.

Recently, Koncel-Kedziorski et al. (2019) used knowledge graphs as the source for text generation. Instead of serializing the labeled and potentially disconnected graph, however, they had to convert their input to a connected and unlabeled version in order to use it with their graph encoder architecture. Our model does not need such preprocessing steps and – as opposed to specialized and separated encoders for graphs and texts – our model can make use of shared structures.

Methodologically, our work resembles most (Lample et al., 2018b). We apply the principles they identified for unsupervised translation from one language to another to the field of text generation from structured data. We adapt the noise model of Lample et al. (2018a) to the domain of serialized knowledge graphs and introduce two new noise functions specific to that domain.

6 Conclusion

We presented a joint model for open information extraction and text generation from text by formalizing both tasks as sequence translation. We presented promising first results on a subset of Visual Genome scene graphs, both for supervised and fully unsupervised settings.
|                        | val   | test  |
|------------------------|-------|-------|
| Rule-based Text2graph  | 0.197 | 0.197 |
| CopyNet (Gu et al., 2016) | 0.222 | 0.211 |
| Combined graph ↔ text  | 0.221 | 0.212 |
| + noise objectives     | 0.205 | 0.203 |
| Unsupervised (iteration 0) | 0.014 | 0.017 |
| Unsupervised (iteration 1) | 0.205 | 0.206 |
| Unsupervised (iteration 2) | 0.181 | 0.180 |
| Unsupervised (iteration 3) | 0.199 | 0.199 |

Table 3: Performance on open information extraction measured in micro-averaged precision, recall and F1 score.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannanve Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

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. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. CoRR, abs/1602.07332.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In International Conference on Learning Representations.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium. Association for Computational Linguistics.

Edward Loper and Steven Bird, 2002. Nltk: The natural language toolkit. In In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics.

Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2016. Visual relationship detection with language priors. In European Conference on Computer Vision.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-Text Generation with Content Selection and Planning. In Proceedings of the 33rd Conference on Artificial Intelligence.

Jacques Pierre Robin. 1995. Revision-based Generation of Natural Language Summaries Providing Historical Background: Corpus-based Analysis, Design, Implementation and Evaluation. Ph.D. thesis,
Kumiko Tanaka-Ishii, Koiti HASIDA, and Itsuki NODA. 1998. Reactive content selection in the generation of real-time soccer commentary. In COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics.

Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. 2008. Handbook of knowledge representation, volume 1. Elsevier.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark. Association for Computational Linguistics.