Discussion Group Summary: Graphics Syntax in the Deep Learning Age

Bertrand Coüasnon[1], Ashok Popat[2], and Richard Zanibbi[3]

[1] Univ Rennes, CNRS, IRISA, F-35000 Rennes, France,
couasnon@irisa.fr
[2] Google Research, Mountain View, CA 94043 (USA),
popat@google.com
[3] Rochester Institute of Technology, NY, USA,
rxzvcs@rit.edu

1 Topics of Discussion

Summary

- Deep learning powerful for object detection & parsing natural language.
- Deep learning data-hungry: labeled graphic datasets often small/absent.
- Graphics recognition distinct from text recognition: harder due to 2D vs. 1D input, importance of distant relationships (e.g., key signature in music)
- Maturity of graphics recognition lags behind text recognition. Should these methods be adapted for 2D graphics, or is a different approach needed?
- Where syntax may help: expressing infrequent patterns in an a priori manner (e.g., in a grammar) rather than inferring them using statistical methods (e.g., deep nets): reduce data dependency and model complexity

Deep learning has produced very good results for object detection and parsing natural language. The discussion started on the specificities of graphics recognition compared to natural language processing: bi-dimensionality; the importance of long-distance relationships; the fact that labeled datasets are often small or absent in graphics, and are very costly to build. As deep learning methods need huge amounts of labeled data, it seems difficult to directly apply them to graphics recognition. Should those methods be adapted for 2D graphics?

As both graphics and natural language are strongly structured by syntax, it seems interesting to answer yes - but it can be hard to find sufficient training data to capture rare long-distant relationships and infer infrequent patterns. Perhaps it is easier to express these less frequently patterns in an a priori manner (e.g, using a grammar). These discussions led to other discussions presented in section 3 on approaches to parsing using deep learning methods, to extend them to 2D, and in section 4 on combining grammatical techniques with deep learning. Before these discussions we had exchanges on 2D structure representations, reported in the following section 2.

Fig. 1. Our discussion group at GREC 2017

2 2D Structure Representations

Summary

- **Comment.** Few representations for graphics structure include cycles. We did not identify non-hierarchical outputs used for graphics recognition.
- Unique ground truth graphs definable when input primitives over-segment recognition targets and are small in number (e.g., PDF symbols, handwritten strokes with at most one symbol).
  - Can use labeled adjacency ('lg') graphs with label *sets* on nodes and edges (per CROHME [1] competitions) for graphs with or without cycles
  - All differences between 'lg' graphs directly identifiable, measurable through *input primitives* fixed across recognition algorithms. Tools available.[4]
  - Possible future work: develop learning/parsing methods over 'lg' graphs
- When exactly matching ground truth impractical (e.g., symbol detection in images), can still compute exact differences in output graphs, but target matching must be approximate (e.g., thresholding intersection-over-union vs. identical locations).
  - May prevent direct learning from 'lg' graphs in this case... future work?
- Editable representations (e.g., CAD, XML) help design & development, provide synthetic training data.

Representation of 2D graphics structure is important for outputs of recognition, ground truth, evaluation, constructing training data, etc. We observed that few representations for graphics structure include cycles and we did not identify non-hierarchical outputs used for graphics recognition. It was pointed out that it is possible to build unique ground truth graphs when input primitives over-segment recognition targets and are small in number, as with handwritten strokes or PDF

[4] CROHME LgEval library: `https://www.cs.rit.edu/~dprl/Software.html`

symbols. An example label graph was demonstrated for the math expression $2 + 3^x$ (see Fig. 2 on the whiteboard). Tools exist that identify and evaluate *all* differences between ground truth and output representations. Possible future work includes learning/parsing methods operating directly upon label graphs.

However, when recognition targets are, for example, symbols detected in images, exact differences in output graphs is still possible but target must be approximated with, for example, intersection-over-union (IoU), label graphs may not be used for learning. This could be explored as future work.

The possibility of generating synthetic training data by viewing the recognition problem as the inverse or dual of graphics authoring, suggests using an editable authoring representation as the output representation of recognition. In particular, vast amounts of training labeled data could then be generated by rendering and distorting instances of the output representation, e.g., using some CAD or desktop publishing XML schema. Coupled with an end-to-end deep learning recognizer, this approach could recover a particularly useful level of semantics, namely that at which a human author would operate.

3 Approaches to Parsing using Deep Learning Methods

Summary

- **Questions.** Can we extend methods for 1D data to 2D? Or is a distinct approach needed - can syntactic pattern recognition techniques be extended/ combined with deep learning?
- **Opinion.** Benefit in deep methods in part from increased reliance upon raw input data (and continuous features) vs. *inferred* discrete entities used in syntactic pattern recognition (e.g., parsing using *recognized* symbols).
- NLP: using recurrent nets to parse text: sentence → parse tree.
- Sequential methods (e.g., LSTM) lose 2D context. Multi-dimensional LSTMs improve this, still do not interpret directly within 2D input space.
- Opportunities

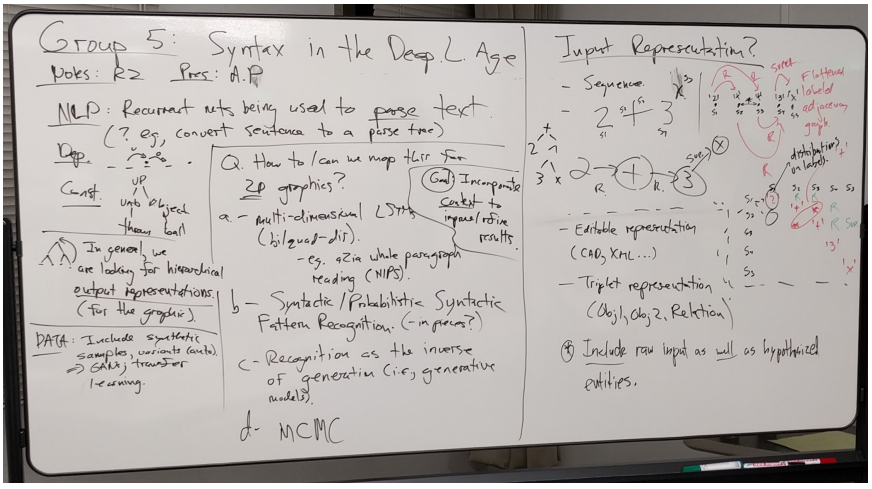

Fig. 2. The whiteboard after our discussion

- Exploiting correlations in feature maps (e.g., a2ia paragraph reading modules use multi-directional LSTMS).
- Constrain problems (e.g., in steps, output graph detail)
- Use loss function forcing network to learn to solve the problem (e.g., identifying target graph)
- Develop *generative models* - clean synthetic data can be helpful for this.

Several questions were asked about the possibility to extend deep learning-based parsing methods from 1D to 2D, and about the possible combination of syntactic pattern recognition and deep learning techniques. One of the most compelling properties of deep methods is their ability to learn features and to work from raw input data; syntactic pattern recognition methods use discrete recognized symbols, generating difficulties arising from making hard decisions early (e.g., for segmentation) and the rapid explosion in combinations when alternative hypotheses are explored. To extend from 1D to 2D, we discussed first recurrent networks, which which are used to parse text (1D) in Natural Language Processing. Recurrent networks such as LSTM lose 2D context, but have been extended to multi-dimensional (MD)LSTM to try to integrate more bi-dimensional information. They still do not use the full 2D input space directly, and instead register/align 1D views.

Some opportunities were discussed including exploiting correlations in feature maps for paragraph reading with multi-directional LSTMs, the definition of loss functions adapted for 2D parsing, and developing generative models using synthetic data.

4 Combining Grammatical Techniques with Deep Learning

Summary

- Preserving uncertainty about hypotheses (i.e., 'weak decisions,' 'late commitment')
- Interface at the triplet level? (object1, object2, relation)
- Strategy: identify sub-problems which are data driven, and where *lots of data is available.*
  - Training Data / Data Expansion (e.g., GAN, transfer learning)
  - Strategy: use grammars to define rare/distant language elements that are hard to infer from data.

The last discussion was on the combination of grammatical techniques with deep learning. This combination offers the possibility to limit the use of grammars to elements for which labeled data is scarce, or where long distance relationships are needed. When sufficient training data is available to infer (probabilistic) syntax reliably, it makes sense to use deep learning techniques. Even more when data is not available, grammars can provide a way to generate training data and complex contextual information for deep learning. For example, grammars can contextually select sub-regions of the graphic document associated with a

contextually reduced vocabulary, to make possible application of techniques like GAN (Generative Adversarial Networks) to automatically generate datasets for future training, or application of data expansion. The combination can also allow a simplification of the grammar definition, in particular offloading segmentation tasks to deep learning modules.

Acknowledgements. We thank the GREC organizers for hosting this event, and all the discussion participants for an engaging and animated discussion.

References

1. H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain. ICFHR2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 607–612, October 2016.