

Cite this: *Chem. Sci.*, 2018, 9, 6091

# “Found in Translation”: predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models†

Philippe Schwaller,<sup>‡\*</sup> Théophile Gaudin,<sup>‡</sup> Dávid Lányi, Costas Bekas and Teodoro Laino

There is an intuitive analogy of an organic chemist's understanding of a compound and a language speaker's understanding of a word. Based on this analogy, it is possible to introduce the basic concepts and analyze potential impacts of linguistic analysis to the world of organic chemistry. In this work, we cast the reaction prediction task as a translation problem by introducing a template-free sequence-to-sequence model, trained end-to-end and fully data-driven. We propose a tokenization, which is arbitrarily extensible with reaction information. Using an attention-based model borrowed from human language translation, we improve the state-of-the-art solutions in reaction prediction on the top-1 accuracy by achieving 80.3% without relying on auxiliary knowledge, such as reaction templates or explicit atomic features. Also, a top-1 accuracy of 65.4% is reached on a larger and noisier dataset.

Received 28th May 2018  
Accepted 20th June 2018DOI: 10.1039/c8sc02339e  
rsc.li/chemical-science

## 1 Introduction

After nearly 200 years of documented research, the synthesis of organic molecules remains one of the most important tasks in organic chemistry. The construction of a target molecule from a set of existing reactants and reagents *via* chemical reactions is attracting much attention because of its economical implications.

Multiple efforts have been made in the past 50 years to rationalize the large number of chemical compounds and reactions identified, which form the large knowledge bases for solving synthetic problems. In 1969, Corey and Wipke<sup>1</sup> demonstrated that both synthesis and retrosynthesis could be performed by a machine. Their pioneering contribution involved the use of handcrafted rules made by experts, which are commonly known as reaction templates. The templates encode the local changes to the atoms' connectivity under certain conditions accounting for various subtleties of retrosynthesis. A similar algorithm emerged in the late 1970s<sup>2</sup> which also requires a set of expert rules. Unfortunately, rules writing is a tedious task, both time and labor-intensive, and may not cover the entire domain for complex organic chemistry problems. In such cases, profound chemical expertise is still required, and the solutions are usually developed by trained organic chemists. However, it can be extremely challenging even for them to

synthesize a relatively complex molecule, which may take several reaction steps to construct. In fact, navigating the chemical space of drug-related compounds by relying only on intuition may turn a synthesis into a nearly impossible task, especially if the problem is slightly outside the expert's knowledge.

Other approaches extract reaction templates directly from data.<sup>3–6</sup> In this specific context, candidate products are generated from the templates and then are ranked according to their likelihood. Satoh and Funatsu<sup>3,4</sup> used various hard-coded criterion to perform the ranking whereas more recent approaches<sup>5,6</sup> used a deep neural network. However, these types of approaches are fundamentally dependent on the rule-based system component and thus inherit some of its major limitations. In particular, these approaches do not produce sufficiently accurate predictions outside of the training domain.

Nevertheless, the class of algorithms<sup>1–6</sup> that is based on rules manually encoded by human experts or automatically derived from a reaction database is not the only way to approach the problem of organic synthesis. A second approach for predicting chemical reactions exploits the advancements in computational chemistry to evaluate the energy barriers of a reaction, based on first-principle calculations.<sup>7–9</sup> Although it is possible to reach very accurate levels of predictions for small systems (chemical reactions involving few hundred atoms), it is still a very computationally daunting task which limits, among other things, the sampling of the solvent degrees of freedom, possibly resulting in unrealistic entropy contributions. Therefore, while computational chemistry may intrinsically solve the problem of reaction prediction, its prohibitive cost does prevent the systematic treatment of all those degrees of freedom that may

IBM Research, Zurich, Switzerland. E-mail: {psh,tga,dla,bek,teo}@zurich.ibm.com

† Electronic supplementary information (ESI) available: Time-split test set and example predictions, together with attention weights, confidence and token probabilities. See DOI: 10.1039/c8sc02339e

‡ P. S. and T. G. contributed equally to this work.

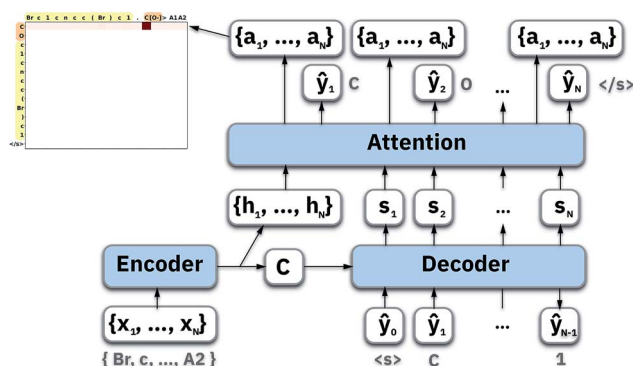






**Table 2** Data preparation steps to obtain source and target sequences. The tokens are separated by a space and individual molecules by a point token

Step	Example (entry 23 738, Jin's USPTO test set <sup>20</sup> ): reactants > reagents > products
(1) Original string	[Cl:1][c:2]1[cH:3][c:4]([CH3:8])[n:5][n:6]1[CH3:7].[OH:14][N+:15]([O-:16])=[O:17].[S:9]([O:10])=[O:11]([OH:12])[OH:13]>>[Cl:1][c:2]1[c:3]([N+:15])=[O:14][O-:16][c:4]([CH3:8])[n:5][n:6]1[CH3:7]
(2) Reactants and reagent separation	[Cl:1][c:2]1[cH:3][c:4]([CH3:8])[n:5][n:6]1[CH3:7].[OH:14][N+:15]([O-:16])=[O:17]>[S:9]([O:10])=[O:11]([OH:12])[OH:13]>[Cl:1][c:2]1[c:3]([N+:15])=[O:14][O-:16][c:4]([CH3:8])[n:5][n:6]1[CH3:7]
(3) Atom-mapping removal and canonicalization	Cc1cc(Cl)n(C)n1.O=[N+]([O-])O>S(=O)(O)O>Cc1nn(C)c(Cl)c1[N+]([O-])
(4) Tokenization	Cc1cc(Cl)n(C)n1.O=[N+]([O-])O>A1>Cc1nn(C)c(Cl)c1[N+]([O-])
Source	Cc1cc(Cl)n(C)n1.O=[N+]([O-])O>A1
Target	Cc1nn(C)c(Cl)c1[N+]([O-])



**Fig. 1** Illustration of an attention-based seq2seq model.

step  $t$  processes an element of the input  $x_t$  and the network's previous hidden state  $h_{t-1}$ . The output and the hidden state transition is defined by

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \otimes \tanh(c_{t-1}), \quad (5)$$

where  $i_t, f_t$  and  $o_t$  are the input, forget, and output gates;  $c$  is the cell state vector;  $W, U$  and  $b$  are model parameters learned during training;  $\sigma$  is the sigmoid function and  $\otimes$  is the entry-wise product. For the encoder, we used a bidirectional LSTM (BLSTM).<sup>37</sup> A BLSTM processes the input sequence in both directions, so they have context not only from the past but also from the future. They comprise two LSTMs: one that processes the sequence forward and the other backward, with their forward and backward hidden states  $\vec{h}_t$  and  $\overleftarrow{h}_t$  for each time step. The hidden states of a BLSTM are defined as

$$h_t = \{\vec{h}_t, \overleftarrow{h}_t\}. \quad (6)$$

Thus we can formalize our encoder as

$$C = f(W_e x_t, h_{t-1}), \quad (7)$$

where  $f$  is a multilayered BLSTM;  $h_t \in \mathbb{R}^R$  are the hidden states at time  $t$ ;  $x_t$  is an element of an input sequence  $x = \{x_0, \dots, x_T\}$ , which is a one-hot encoding of our vocabulary; and  $W_e$  are the learned embedding weights. Generally,  $C$  is simply the last of the encoder's hidden states:

$$C = h_T \quad (8)$$

The second part of the model – the decoder – predicts the probability of observing a product  $\hat{y} = \{\hat{y}_1, \dots, \hat{y}_M\}$ :

$$P(\hat{y}) = \prod_{i=0}^M p(\hat{y}_i | \{\hat{y}_1, \dots, \hat{y}_{i-1}\}) \quad (9)$$

and for a single token  $\hat{y}_i$ :

$$p(\hat{y}_i | \{\hat{y}_1, \dots, \hat{y}_{i-1}\}, c_i) = g(\hat{y}_{i-1}, s_i, c_i), \quad (10)$$

where  $g$  is a stack of LSTM, which outputs the probability  $\hat{y}_t$  for a single token;  $s_i$  are the decoder's hidden states; and  $c_i$  is a different context vector for each target token  $y_i$ . Bahdanau *et al.*<sup>38</sup> and Luong *et al.*<sup>31</sup> proposed attention mechanisms, *i.e.*, different ways for computing the  $c_i$  vector rather than taking the last hidden state of the encoder  $h_t$ . We performed experiments using both models and describe Luong's method, which yielded the best overall results.

#### 4.1 Luong's attention mechanism

To compute the context vector, we first have to compute the attention weights  $\alpha$ :

$$\alpha_{it} = \frac{\exp(s_i^\top W_\alpha h_t)}{\sum_{t'=0}^T \exp(s_i^\top W_\alpha h_{t'})} \quad (11)$$

$$c_i = \sum_{t=0}^T \alpha_{it} h_t, \quad (12)$$

The attention vector is then defined by

$$a_i = \tanh(W_a \{c_i; s_i\}). \quad (13)$$

Both  $W_\alpha$  and  $W_a$  are learned weights. Then  $a$  can be used to compute the probability for a particular token:



$$p(y_i | \{y_1, \dots, y_{i-1}\}, c_i) = \text{softmax}(W_p a_i), \quad (14)$$

where  $W_p$  are also the learned projection weights.

## 4.2 Training details

During training, all parameters of the network were trained jointly using a stochastic gradient descent. The loss function was a cross-entropy function, expressed as

$$H(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i) \quad (15)$$

for a particular training sequence. The loss was computed over an entire minibatch and then normalized. The weights were initialized using a random uniform distribution ranging from  $-0.1$  to  $0.1$ . Every 3 epochs, the learning rate was multiplied by a decay factor. The minibatch size was 128. Gradient clipping was applied when the norm of the gradient exceeded 5.0. The teacher forcing method<sup>39</sup> was used during training.

## 5 Architecture & hyperparameter search

Finding the best-performing set of hyperparameters for a deep neural network is not trivial. As mentioned in Section 4, our model has numerous parameters that can influence both its training and its architecture. Depending on those parameters, the performance of the model can vary notably. In order to select the best parameters efficiently, we build a framework around scikit-optimize.<sup>40</sup> After the evaluation of 10 random sets, a gradient-boosted tree model<sup>41</sup> was used as surrogate model together with expected improvement as acquisition function<sup>42</sup> to guide the hyperparameter search on a space defined in Table 3. The sets of hyperparameters were evaluated according to their accuracy on the validation set. In total, we trained 100 models for 30 epochs. The set of best hyperparameters found with this method is highlighted in bold. This model has been further trained to 80 epochs to improve its final accuracy.

## 6 Experiments

### 6.1 Reaction prediction

We evaluated our model on two data sets and compared the performance with other state-of-the-art results. After the

Table 3 Hyperparameters space, parameters for the best model in bold

Parameter	Possible values
Number of units	128, 256, 512 or <b>1024</b>
Number of layers	2, 4 or 6
Type of encoder	LSTM, <b>BLSTM</b>
Output dropout	0–0.9 ( <b>0.7676</b> )
State dropout	0–0.9 ( <b>0.5374</b> )
Learning rate	0.1–5 ( <b>0.355</b> )
Decay factor	0.85–0.99 ( <b>0.854</b> )
Type of attention	“ <b>Luong</b> ” or “Badhanau”

hyperparameter optimization, we continued to train our best model on the 395 496 reactions in Jin's USPTO train set and tested the fully trained model on Jin's USPTO test set. Additionally, we trained a second model with the same hyperparameters on 902 581 randomly chosen single-product reactions from the more complex and noisy Lowe dataset and tested it on a set of 50 258 reactions. As molecules are discrete data, changing a single character, such as in source code or arithmetic expressions, can lead to completely different meanings or even invalidate the entire string. Therefore we use full-sequence accuracy, the strictest criteria possible, as our evaluation metric by which a test prediction is considered correct only if all tokens are identical to the ground truth.

The network had to solve three major challenges. First, it had to memorize the SMILES grammar to predict synthetically correct sequences. Second, because we trained it on canonicalized molecules, the network had to learn the canonical representation. Third, the network had to map the reactants plus reagents space to the product space.

Although the training was performed without a beam search, we used a beam width of 10 without length penalty for the inference. Therefore the 10 most probable sequences were kept at every time step. This allowed us to know what probability the network assigned to each of the sequences. We used the top-1 probabilities to analyze the prediction confidence of the network.

The final step was to canonicalize the network output. This simple and deterministic reordering of the tokens improved the accuracy by 1.5%. Thus, molecules that were correctly predicted, but whose tokens were not enumerated in the canonical order, were still counted as correct. The prediction accuracies of our model on different datasets are reported in Table 4. For single product reactions, we achieved an accuracy of 83.2% on Jin's USPTO test dataset and 65.4% on Lowe's test set.

An additional validation can be found in the ESI,<sup>†</sup> where we used the model trained on Lowe's dataset to predict reactions from pistachio,<sup>45</sup> a commercial database of chemical reactions extracted from the patent literature. Because the Lowe's dataset used to train the model contained reactions until September 2016, we only predicted the reactions from 2017 and hence, we had a time split.

### 6.2 Comparison with the state of the art

To the best of our knowledge, no previous work has attempted to predict reactions on the complete US patent dataset of Lowe.<sup>33</sup> Table 5 shows a comparison with the Weisfeiler–Lehman difference networks (WLDN) from Jin *et al.*<sup>20</sup> on their USPTO test set.

Table 4 Scores of our model on different single product datasets

Test set	Size	BLEU, <sup>43</sup> ROUGE <sup>44</sup>	Accuracies in [%]		
			Top-1	Top-2	Top-3
Jin's USPTO <sup>20</sup>	38 648	95.9, 96.0	<b>83.2</b>	87.7	89.2
Lowe's <sup>33</sup>	50 258	90.3, 90.9	<b>65.4</b>	71.8	74.1





## 6.5 Limitations

Our model is not without limitations. An obvious disadvantage compared to template-based methods is that the strings are not guaranteed to be a valid SMILES. Incorporating a context-free grammar layer, as was done in ref. 12, could bring minor improvements. Fortunately, only 1.3% of the top-1 predictions are grammatically erroneous for our model.

Another limitation of the training procedure are multiple product reactions. In contrast to words in a sentence, the exact order in which the molecules in the target string are enumerated does not matter. A viable option would be to include in the training set all possible permutations of the product molecules.

Our hyperparameter space during optimization was restricted to a maximum of 1024 units for the encoder. Using more units could have led to improvements. On Jin's USPTO dataset, the training plateaued because an accuracy of 99.9% was reached and the network had memorized almost the entire training set. Even on Lowe's noisier dataset, a training accuracy of 94.5% was observed. A hyperparameter optimization could be performed on Lowe's dataset to improve the prediction accuracy.

## 7 Conclusion

Predicting reaction outcomes is a routine task of many organic chemists trained to recognize structural and reactivity patterns reported in a wide number of publications. Not only did we show that a seq2seq model with correctly tuned hyperparameters can learn the language of organic chemistry, our approach also improved the current state-of-the-art in patent reaction outcome prediction by achieving 80.3% on Jin's USPTO dataset and 65.4% on single product reactions of Lowe's dataset. Similar to the work of Nam and Kim,<sup>17</sup> our approach is fully data driven and free of chemical knowledge/rules and compared to their work,<sup>17</sup> we take full advantage of the attention mechanism. Also worth mentioning is the overall simplicity of our model that jointly trains the encoder, decoder and attention layers end-to-end. Our hope is that, with this type of model, chemists can codify and perhaps one day fully automate the art of organic synthesis.

## Conflicts of interest

There are no conflicts to declare.

## Acknowledgements

We thank Nadine Schneider, Greg Landrum and Roger Sayle for the helpful discussions on RDKit and the datasets. We also would like to acknowledge Marwin Segler and Hiroko Satoh for useful feedback on our approach.

## Notes and references

§ <https://github.com/wengong-jin/nips17-rexgen>.

¶ <https://ibm.box.com/v/ReactionSeq2SeqDataset>.

- 1 E. J. Corey and W. T. Wipke, *Science*, 1969, **166**, 178–192.
- 2 T. D. Salatin and W. L. Jorgensen, *J. Org. Chem.*, 1980, **45**(11), 2043–2051.
- 3 H. Satoh and K. Funatsu, *J. Chem. Inf. Comput. Sci.*, 1995, **35**, 34–44.
- 4 H. Satoh and K. Funatsu, *J. Chem. Inf. Comput. Sci.*, 1996, **36**, 173–184.
- 5 M. H. S. Segler and M. P. Waller, *Chem.–Eur. J.*, 2017, **23**, 6118–6128.
- 6 C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green and K. F. Jensen, *ACS Cent. Sci.*, 2017, **3**, 434–443.
- 7 W. R. J. Dolbier, K. Henryk, K. Houk and S. Chimin, *Acc. Chem. Res.*, 1996, **29**, 471–477.
- 8 D. Mondal, S. Y. Li, L. Bellucci, T. Laino, A. Tafi, S. Guccione and S. D. Lepore, *J. Org. Chem.*, 2013, **78**, 2118–2127.
- 9 O. Engkvist, P.-O. Norrby, N. Selmi, Y.-h. Lam, Z. Peng, E. C. Sherer, W. Amberg, T. Erhard and L. A. Smyth, *Drug Discovery Today*, 2018, **23**(6), 1203–1218.
- 10 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**(14), 31–36.
- 11 A. Cadeddu, E. K. Wylie, J. Jurczak, M. Wampler-Doty and B. A. Grzybowski, *Angew. Chem., Int. Ed.*, 2014, **53**, 8108–8112.
- 12 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2018, **4**, 268–276.
- 13 S. Jastrzębski, D. Leśniak and W. M. Czarnecki, Learning to SMILE(S), 2016, <http://arxiv.org/abs/1602.06289>.
- 14 M. J. Kusner, B. Paige and J. M. Hernández-Lobato, *ICML*, 2017.
- 15 E. J. Bjerrum, *SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules*, 2017, <http://arxiv.org/abs/1703.07076>.
- 16 M. H. S. Segler, T. Kogej, C. Tyrchan and M. P. Waller, *ACS Cent. Sci.*, 2018, **4**, 120–131.
- 17 J. Nam and J. Kim, *Linking the Neural Machine Translation and the Prediction of Organic Chemistry Reactions*, 2016, <https://arxiv.org/pdf/1612.09529.pdf>.
- 18 N. Schneider, D. M. Lowe, R. A. Sayle and G. A. Landrum, *J. Chem. Inf. Model.*, 2015, **55**, 39–53.
- 19 N. Schneider, N. Stiefl and G. A. Landrum, *J. Chem. Inf. Model.*, 2016, **56**, 2336–2346.
- 20 W. Jin, C. Coley, R. Barzilay and T. Jaakkola, *NIPS*, 2017, pp. 2607–2616.
- 21 J. N. Wei, D. Duvenaud and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2016, **2**, 725–723.
- 22 D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarelli, T. Hirzel, A. Aspuru-Guzik and R. P. Adams, *NIPS*, 2015.
- 23 D. M. Lowe, Ph.D. thesis, University of Cambridge, 2012.
- 24 M. H. S. Segler and M. P. Waller, *Chem.–Eur. J.*, 2017, **23**, 5966–5971.
- 25 J. Bauer, E. Fontain, D. Forstmeyer and I. Ugi, *Tetrahedron Comput. Methodol.*, 1988, **1**, 129–132.
- 26 P. Röse and J. Gasteiger, *Anal. Chim. Acta*, 1990, **235**, 163–168.
- 27 W. L. Jorgensen, E. R. Laird, A. J. Gushurst, J. M. Fleischer, S. A. Gothe, H. E. Helson, G. D. Paderes and S. Sinclair, *Pure Appl. Chem.*, 1990, **62**, 1921–1932.



- 28 W. A. Warr, *Mol. Inf.*, 2014, **33**, 469–476.
- 29 M. A. Kayala and P. Baldi, *J. Chem. Inf. Model.*, 2012, **52**, 2526–2540.
- 30 M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng and G. Brain, *OSDI*, 2016.
- 31 M.-T. Luong, H. Pham and C. D. Manning, *EMNLP*, 2015.
- 32 B. Liu, B. Ramsundar, P. Kawthekar, J. Shi, J. Gomes, Q. Luu Nguyen, S. Ho, J. Sloane, P. Wender and V. Pande, *ACS Cent. Sci.*, 2017, **3**, 1103–1113.
- 33 D. M. Lowe, Chemical reactions from *US pat.* (1976-Sep2016), 2017, [https://figshare.com/articles/Chemical\\_reactions\\_from\\_US\\_patents\\_1976-Sep2016\\_/5104873](https://figshare.com/articles/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873).
- 34 G. Landrum, B. Kelley, P. Tosco, S. Riniker, Gedeck, N. Schneider, R. Vianello, A. Dalke, S. Alexander, S. Turk, M. Swain, B. Cole, J. P. Strets123, JLVarjo, A. Pahl, P. Fuller, G. Doliath, M. Wójcikowski, D. Cosgrove, G. Sforna, M. Nowotka, J. H. Jensen, J. Domański, D. Hall, N. O'Boyle, W.-G. Bolick, Nhfechner and S. Roughley, *Rdkit/Rdkit: 2017\_09\_1 (Q3 2017) Release*, 2017, <https://zenodo.org/record/1004356#.Wd3LDY6l2EI>.
- 35 R. Zhao, T. Luong and E. Brevdo, *Neural Machine Translation (seq2seq) Tutorial*, 2017, <https://github.com/tensorflow/nmt>.
- 36 S. Hochreiter and J. Schmidhuber, *Neural Comput.*, 1997, **9**, 1735–1780.
- 37 A. Graves and J. Schmidhuber, *Neural Network.*, 2005, **18**, 602–610.
- 38 D. Bahdanau, K. Cho and Y. Bengio, *ICLR*, 2015.
- 39 R. J. Williams and D. Zipser, *Neural Comput.*, 1989, **1**, 270–280.
- 40 T. Head, N. Campos, M. Cherti, A. Fabisch, T. Fan, M. Kumar, G. Louppe, K. Malone, M. Pak, I. Shcherbatyi, T. Smith and Z. Vinícius, *Scikit-Optimize*, 2017, <http://scikit-optimize.github.io/>.
- 41 J. H. Friedman, *Ann. Stat.*, 2001, **29**(5), 1189–1232.
- 42 J. Mockus, *J. Global Optim.*, 1994, **4**, 347–365.
- 43 K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, *ACL*, 2001.
- 44 C.-Y. Lin, *ACL*, 2004.
- 45 N. Schneider, D. M. Lowe, R. A. Sayle, M. A. Tarselli and G. A. Landrum, *J. Med. Chem.*, 2016, **59**, 4385–4402.

