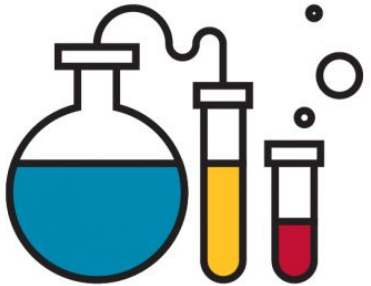
Mapping the Space of Chemical Reactions Using Attention-Based Neural Networks

Machine Learning in Chemistry

Machine Learning in Chemistry

- predicting chemical/physical properties:
  - protein 3D structure (AlphaFold)
  - quantum chemistry computations
- creation of optimal synthesis routes
- ML-aided drug design
- identification of optimal reaction conditions
- sensors for IoT
- other tasks: for example classifying organic reactions



https://alphafold.ebi.ac.uk/, https://arxiv.org/pdf/1904.10370.pdf

Chemistry revision

Organic chemistry

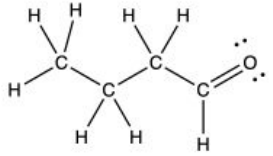
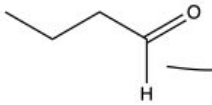
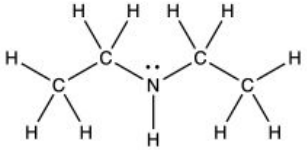
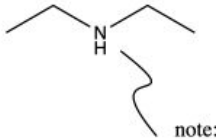
“**Organic chemistry** is the study of the structure, properties, composition, reactions, and preparation of **carbon-containing compounds**. Most organic compounds contain **carbon** and **hydrogen**, but they may also include any number of other elements (e.g., nitrogen, oxygen, halogens, phosphorus, silicon, sulfur). Originally limited to the study of compounds produced by living organisms, organic chemistry has been broadened to include human-made substances (e.g., plastics).”

In daily life:

- health care, cosmetics, drugs, vitamins
- biofuels
- textiles, rubber, plastic
- food additives

https://www.acs.org/content/acs/en/careers/chemical-sciences/areas/organic-chemistry.html ,
https://www.thoughtco.com/organic-chemistry-in-everyday-life-608694

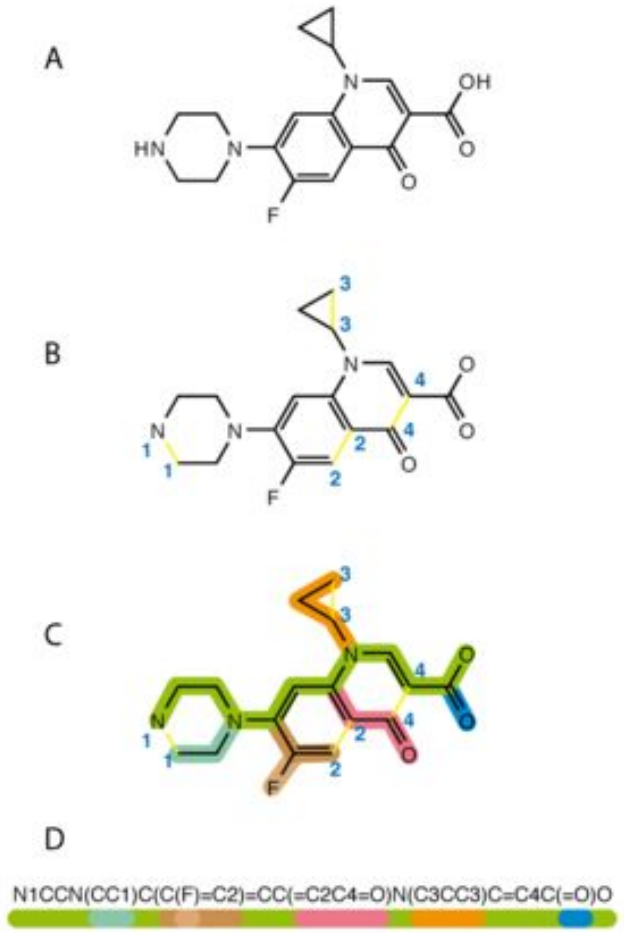
Notation, part 1

Lewis / Kekule Structure	Line Structure	Condensed Formula
		$CH_3(CH_2)_2CHO$
	or	
	 note: a C=O with a hydrogen attached is called an aldehyde. It is very common to label this hydrogen on an aldehyde.	
		$CH_3CH_2NHCH_2CH_3$ or $(CH_3CH_2)_2NH$
	note: hydrogens on heteroatoms (such as N or O) are usually labeled, unlike hydrogens on carbons	

Notation part 2 - SMILES

SMILES = Simplified Molecular-Input Line-Entry System

1. Break cycles
2. Find the backbone
3. Write the backbone and the branches off it



Reaction classes

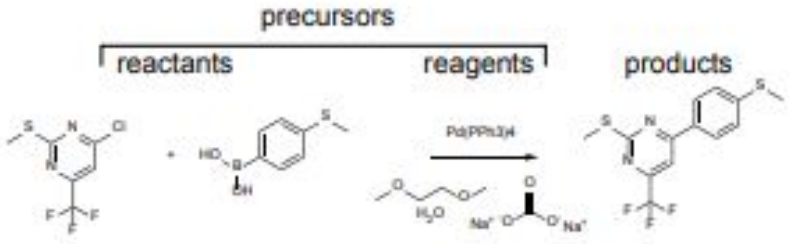
Organic reactions are usually assigned to classes containing reactions with similar reagents and mechanisms.

They are useful to:

- navigate large databases
- infer optimal reactions conditions
- communicate efficiently with other chemists
- assess the quality of reaction prediction

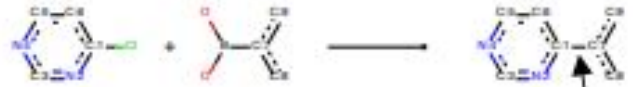
Doing it using ML was the goal of the paper. So far it is done using software based on expert-written set of rules.

Example



COCCOC.CSc1ccc(B(O)O)cc1.CSc1nc(Cl)cc(C(F)(F)F)n1.O.
O=C([O-])[O-].[Na+].[Na+].c1ccc(cc1)[P](c1ccccc1)(c1ccccc1)[Pd]([P]
(c1ccccc1)(c1ccccc1)c1ccccc1)([P](c1ccccc1)(c1ccccc1)c1ccccc1)[P]
(c1ccccc1)(c1ccccc1)c1ccccc1>>CSc1ccc(-c2cc(C(F)(F)F)nc(SC)n2)cc1

Named reaction:	Chloro Suzuki coupling	3.1.2
Superclass:	C-C Bond Formation	3



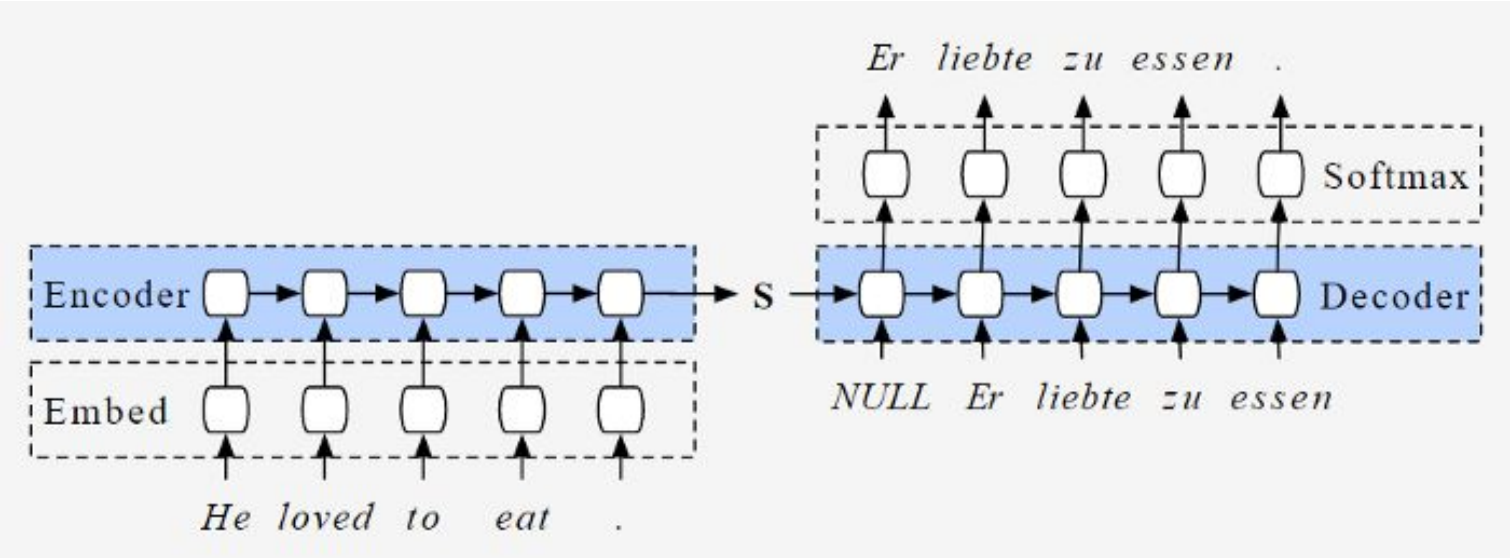
Organochloride Boronic acid C-C bond formed

Organic reactions classification

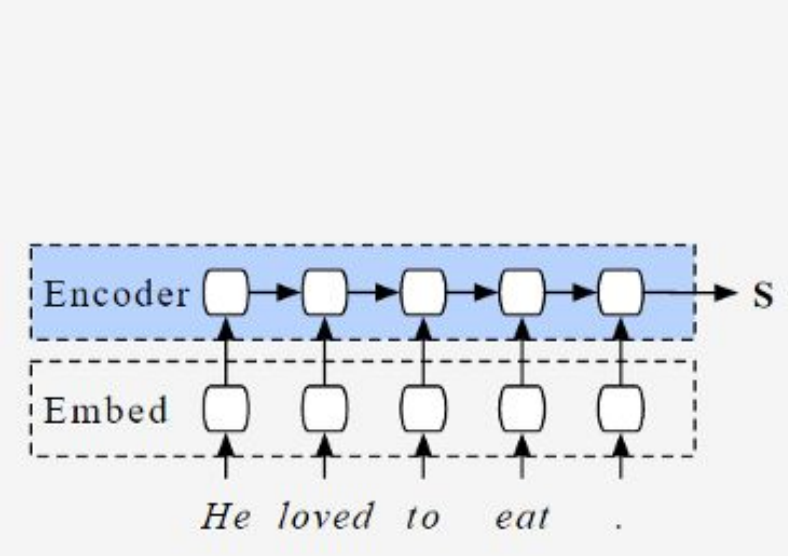
Data

- Pistachio dataset (2.6 m reactions)
- USPTO 1k TPL (445 k reactions)
- Both strongly imbalanced
- Reaction data was classified using RXN, a rule-based software that classifies roughly 1000 different reactions' names
- Reactions are represented with SMILES
- SMILES are then tokenized (roughly symbol by symbol) to be used in NLP models

Model 1: encoder-decoder transformer for seq-2-seq tasks

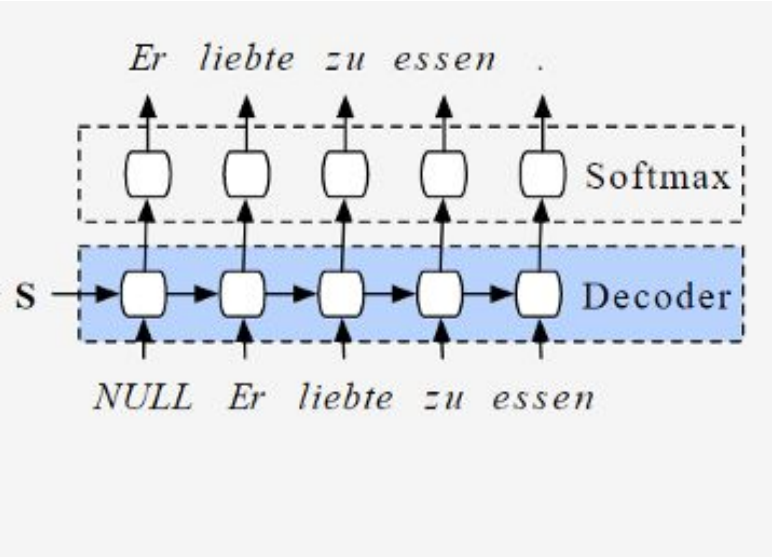


Encoder layer



- LSTM/GRU recurrent unit (or several)
- Input sequence is processed one token at a time
- Recurrent unit accepts the input token and previous hidden state, updates the hidden state and produces output
- Final hidden state is used as initial hidden state of decoder layer (embedding)
- 2 layers

Decoder layer

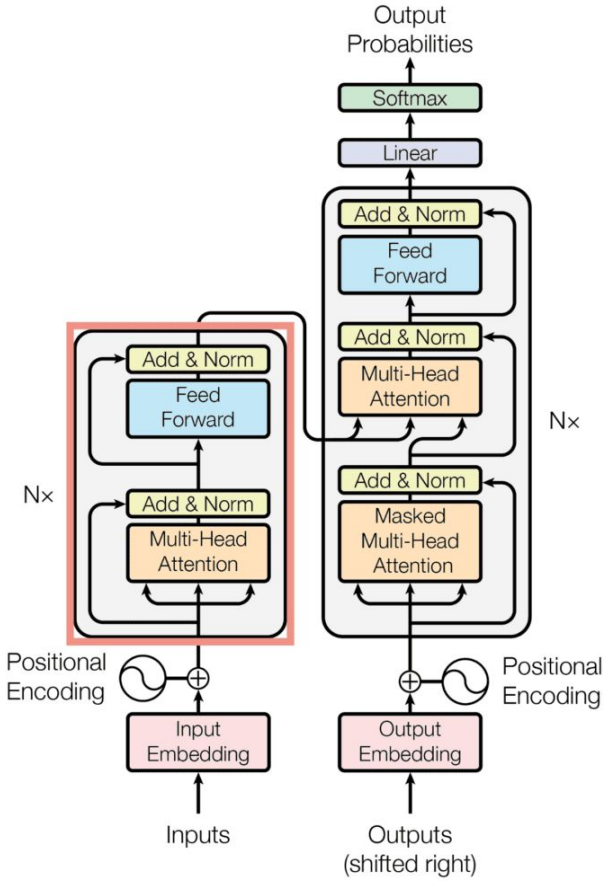


- LSTM/GRU recurrent unit (or several)
- It’s initial hidden state is set to final hidden state of encoder layer
- Recurrent unit accepts the previous output and hidden state, updates the hidden state and produces next output (here: class prediction: superclass + category + reaction name)
- 1 layer

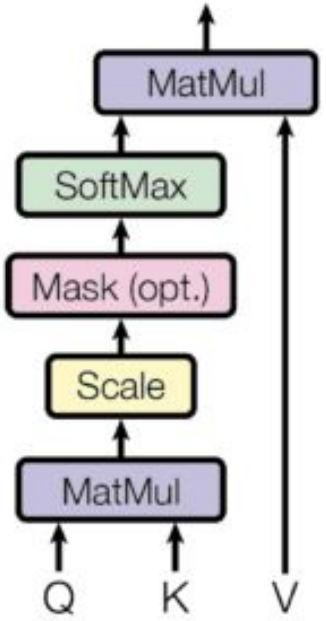
Model 2: BERT

BERT = Bidirectional Encoder Representations from Transformers

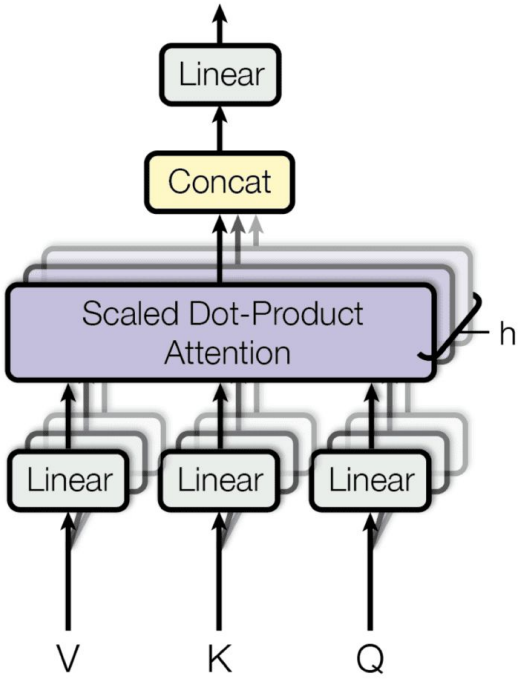
- Input: Tokenized sequence
- Encoding: normally uses word2vec + positional encoding; here not specified
- Encoder consists of stacks of layers composed of multi-head attention mechanisms + normal feed forward network + residual connections



Model 2: BERT - attention mechanism



Dot-Product Attention



Multi-head Attention

- Allows model to focus on inputs that are relevant to query, while also taking into consideration other inputs and relations between them
- Q, K, V are generated by multiplying the input embeddings by 3 weight matrices
- Multiple heads => different functionalities of heads

BERT training

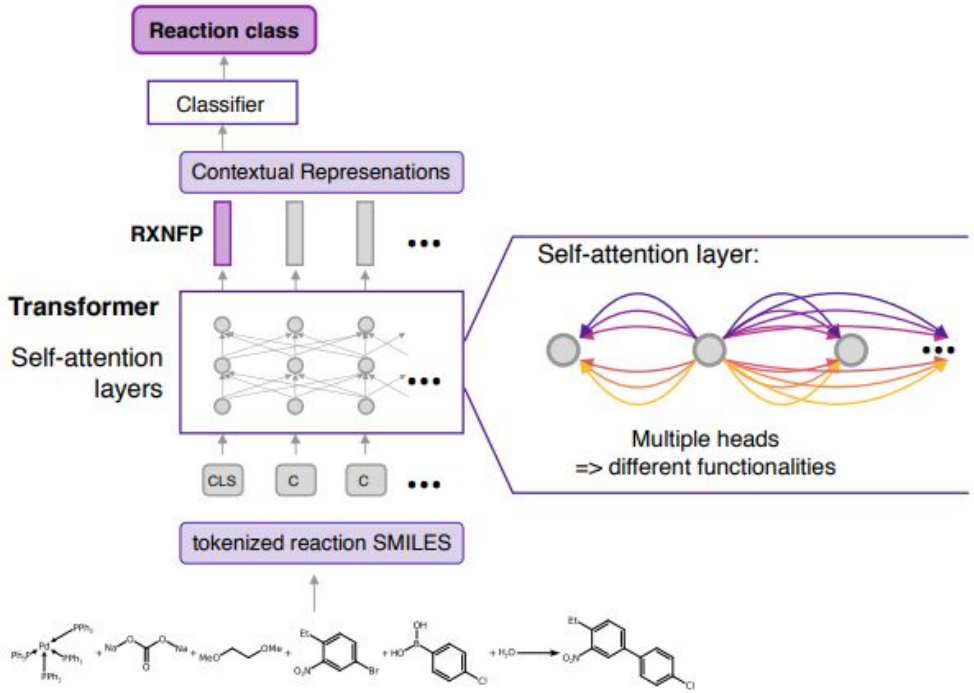
For NLP tasks:

1. Pretraining: masking 15% of words in sequence and predicting them + next sentence prediction (whether second sentence from a pair comes after the first)
2. Fine-tuning to specific task

Here:

1. Pretraining: masking 15 % of tokens
2. Fine-tuning: classification of reaction to classes (DNN with CLS embeddings as input)

BERT for classification



- [CLS] token is never masked during pretraining + it is always at the beginning => it's encoding depends only on the reaction
- Intuition: model uses [CLS] to embed a global description of reaction (reaction embedding)

Metrics for imbalanced datasets: MCC

MCC = Matthew Correlation Coefficient:

$$cov(X,Y) = \sum_{i,j,k=1}^{|C|} \Big(Matrix(i,i)Matrix(k,j) - Matrix(j,i)Matrix(i,k) \Big)$$

$$cov(X,X) = \sum_{i=1}^{|C|} \left[\Big(\sum_{j=1}^{|C|} Matrix(j,i) \Big) \Big(\sum_{k,l=1,k\neq i}^{|C|} Matrix(l,k) \Big) \right]$$

$$cov(Y,Y) = \sum_{i=1}^{|C|} \left[\Big(\sum_{j=1}^{|C|} Matrix(i,j) \Big) \Big(\sum_{k,l=1,k\neq i}^{|C|} Matrix(k,l) \Big) \right]$$

$$MCC = \frac{cov(X,Y)}{\sqrt{cov(X,X) \times cov(Y,Y)}}.$$

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Metrics for imbalanced datasets: CEN

CEN = Confusion Entropy (of confusion matrix)

$$P_{i,j}^{j} = \frac{Matrix(i,j)}{\sum_{k=1}^{|C|}\left(Matrix(j,k) + Matrix(k,j)\right)}, \qquad P_{i,j}^{i} = \frac{Matrix(i,j)}{\sum_{k=1}^{|C|}\left(Matrix(i,k) + Matrix(k,i)\right)}$$

$$CEN_j = -\sum_{k=1,k\neq j}^{|C|}\left(P_{j,k}^{j} log_{2(|C|-1)}\left(P_{j,k}^{j}\right) + P_{k,j}^{j} log_{2(|C|-1)}\left(P_{k,j}^{j}\right)\right)$$

$$P_j = \frac{\sum_{k=1}^{|C|}\left(Matrix(j,k) + Matrix(k,j)\right)}{2\sum_{k,l=1}^{|C|} Matrix(k,l)}$$

$$CEN = \sum_{j=1}^{|C|} P_j CEN_j$$

Results

Pistachio	Accuracy	CEN	MCC
Traditional fp^{25} + 5-NN classifier	0.410	0.365	0.305
Transformer enc2-dec1	0.952	0.039	0.946
BERT classifier	**0.982**	**0.014**	**0.980**
rxnfp (pretrained) + 5-NN classifier	0.819	0.121	0.797
rxnfp + 5-NN classifier	**0.989**	**0.010**	**0.988**
USPTO 1k TPL	Accuracy	CEN	MCC
Traditional fp^{25} + 5-NN classifier	0.295	0.424	0.292
BERT classifier	**0.989**	**0.006**	**0.989**
rxnfp (pretrained) + 5-NN classifier	0.340	0.392	0.337
rxnfp + 5-NN classifier	**0.989**	**0.006**	**0.989**

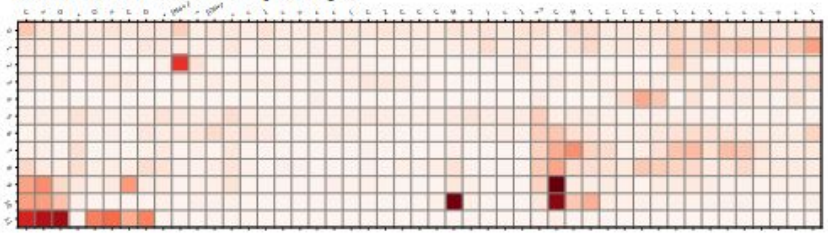
- Important: “ground truth” is generated by software
- Most of errors are related to “Unrecognised” reactions which were classified correctly by models, but not by software (ex. tautomers)
- Robust against errors in SMILES

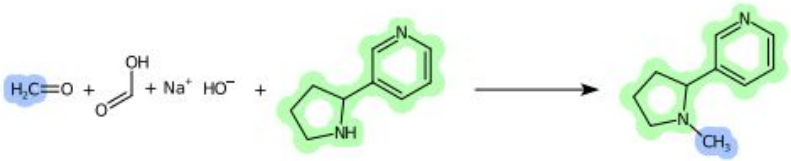
Results: visualization of weights

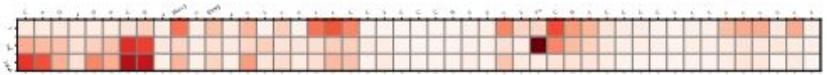
Eschweiler-Clarke methylation [1.2.4]

C=O.O=CO.[Na+].[OH-].c1cncc(C2CCCN2)c1>>CN1CCCC1c1cccnc1

BERT: [CLS] attention per layer





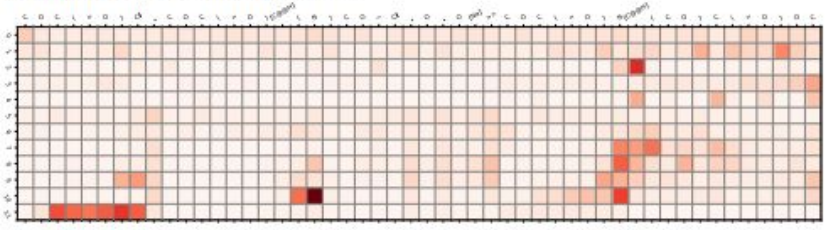


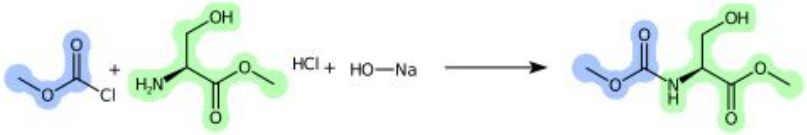
Seq2Seq: encoder-decoder attention

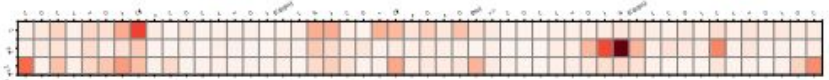
Amide Schotten-Baumann reaction [2.1.1]

COC(=O)Cl.COC(=O)[C@@H](N)CO.Cl.O.O[Na]>>COC(=O)N[C@@H](CO)C(=O)OC

BERT: [CLS] attention per layer







Seq2Seq: encoder-decoder attention

Reaction fingerprints

Fingerprint - fixed-size vectors encodings of molecules/reactions.

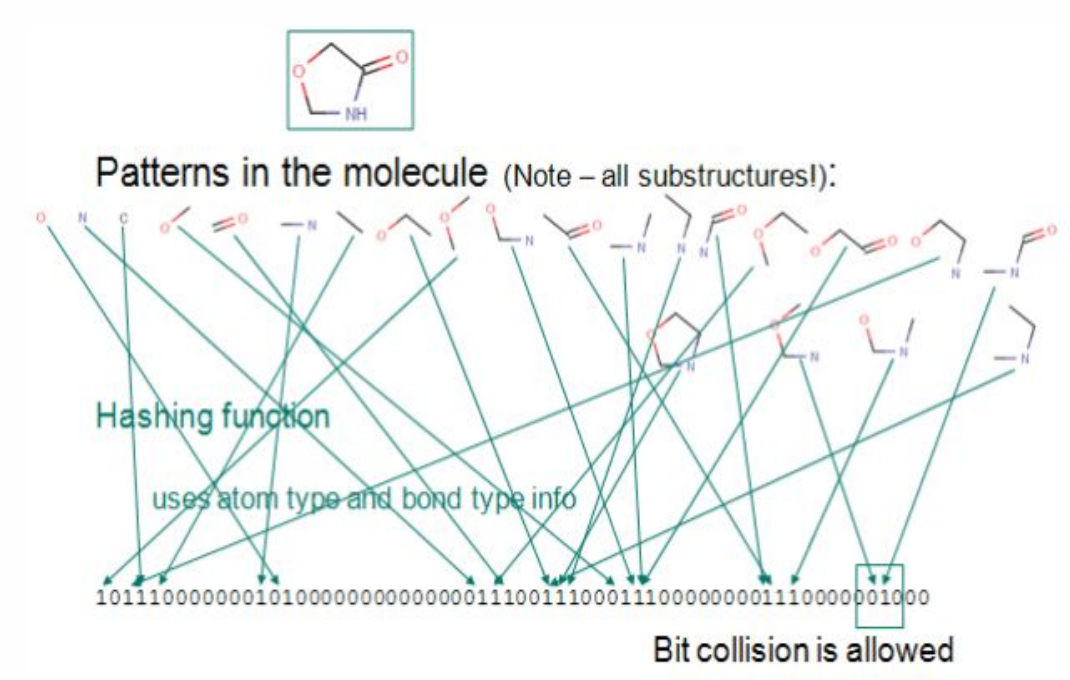
Traditionally handcrafted. Since 2016:

Attempts to learn them,

but very limited (fixed reaction scheme).

Usage:

- input to other ML models
- to search (fast) for similar reactions in database:
  - increase the explainability of blackbox models
  - easy access to metadata (ex. reaction conditions)



Reaction embeddings from NLP = good fingerprints

Pistachio	Accuracy
Traditional fp[25] + 5-NN classifier	0.410
Transformer enc2-dec1	0.952
BERT classifier	**0.982**
rxnfp (pretrained) + 5-NN classifier	0.819
rxnfp + 5-NN classifier	**0.989**
USPTO 1k TPL	Accuracy
Traditional fp[25] + 5-NN classifier	0.295
BERT classifier	**0.989**
rxnfp (pretrained) + 5-NN classifier	0.340
rxnfp + 5-NN classifier	**0.989**

- Created fingerprints are independent of the reaction scheme, don't require atom mapping or reactant-reagant separation = applicable to any reaction database
- They perform much better than traditional ones
- Already used to predict chemical reaction yields and activation energies

https://pubs.rsc.org/en/content/articlepdf/2021/sc/d0sc04896h,
https://iopscience.iop.org/article/10.1088/2632-2153/abc81d/pdf

Visualization (TMAP)

TMAP = indexing based on locality-sensitive hashing + KNN graph + minimal spanning tree

https://rxn4chemistry.github.io/rxnfp/tmaps/tmap_ft_10k.html

Thanks for your attention!

Resources

Mapping the Space of Chemical Reactions using Attention-Based Neural Networks (Philippe Schwaller, Daniel Probst, Alain C. Vaucher, Vishnu H. Nair, David Kreutter, Teodoro Laino, Jean-Louis Reymon): https://chemrxiv.org/engage/chemrxiv/article-details/60c753a0bdbb89acf8a3a4b5

Code: https://github.com/rxn4chemistry/rxnfp

Tutorials: https://rxn4chemistry.github.io/rxnfp/