# Enhancing diversity in language based models for single-step retrosynthesis

Alessandra Toniato[1,2,*], Alain C. Vaucher[1,2], Philippe Schwaller[1,2], and Teodoro Laino[1,2]

[1]*IBM Research Europe, Saümerstrasse 4, 8803 Rüschlikon, Switzerland*
[2]*National Center for Competence in Research-Catalysis (NCCR-Catalysis),Zurich, Switzerland*
[*]*Correspondence: ato@zurich.ibm.com*

## Abstract

Over the past four years, several research groups demonstrated the combination of domain-specific language representation with recent NLP architectures to accelerate innovation in a wide range of scientific fields. Chemistry is a great example. Among the various chemical challenges addressed with language models, retrosynthesis demonstrates some of the most distinctive successes and limitations. Single-step retrosynthesis, the task of identifying reactions able to decompose a complex molecule into simpler structures, can be cast as a translation problem, in which a text-based representation of the target molecule is converted into a sequence of possible precursors. A common issue is a lack of diversity in the proposed disconnection strategies. The suggested precursors typically fall in the same reaction family, which limits the exploration of the chemical space. We present a retrosynthesis Transformer model that increases the diversity of the predictions by prepending a classification token to the language representation of the target molecule. At inference, the use of these prompt tokens allows us to steer the model towards different kinds of disconnection strategies. We show that the diversity of the predictions improves consistently, which enables recursive synthesis tools to circumvent dead ends and consequently, suggests synthesis pathways for more complex molecules.

## 1 Introduction

Finding the optimal combination of readily available chemical building blocks to produce a desired molecule is the Holy Grail of synthetic chemistry. The objective is to infer the

individual (reaction) steps leading to a target material from known starting materials. This method, known as retrosynthesis, is a technique that was long thought to be the exclusive domain of a small but dedicated group of experts. In today's world, retrosynthesis is crucial to solving many materials problems. Still, a growing number of experts is challenged by the complexity of the vast corpus of publicly available chemical information. Computers lead to the development of rule-based algorithms in which disconnection rules were applied to appropriate molecules to achieve the desired transformation. Recent research has leveraged the powerful Deep Learning models to solve the problem and automate the operation while still allowing for the skilled oversight of human chemists. Different models have been proposed [23, 5, 24, 6, 21, 4, 8, 25, 28, 13, 26], usually classified as template-based or template-free models. Template-based models are trained to predict pre-extracted rules, while template-free models learn the retrosynthetic rules from the training data. The principle that underlies all these methods is the same: a model is trained on some data (often the compound to synthesize, given as a text string, an embedding, or a graph) and then evaluated by comparing its output to a target (the set of "optimal" precursors). However, this perspective is sometimes at odds with the chemistry at hand. In fact, for each target molecule, there is generally a wide variety of valid disconnections that connect the target molecule to different sets of precursors. If the dataset were hypothetically perfectly balanced, all conceivable reactions leading to a target molecule would be evenly represented, but in practice this is far from being the case. Existing reaction datasets, and consequently models, give more weight to well-represented reaction classes, thus penalising more interesting but less frequent disconnections. For example, Figure 1 shows an insufficient diversity for the proposed list of disconnections. Here, we interpret diversity as "chemical class diversity", considering a model more diverse in its predictions if these belong to different reaction classes as defined by NameRXN [2].

To increase the diversity of the predictions in single-step text-based retrosynthesis models and counteract the effect of imbalanced datasets, we propose a prompt-based scheme to enhance and guide more diversity in the language model predictions. We introduce a modified transformer-based model [29, 21]. Inspired by works in natural language processing for prompt-based learning [15, 11, 18, 9], we show that concatenating a class information during training (as an additional token), leads to more diverse predictions at inference. We experiment with different classification strategies, including clustering reaction fingerprints [22] to evaluate the adequate number of tokens. We compare the cluster token prompt model to a baseline translation model in terms of top$n$ accuracy, round-trip accuracy, class diversity and coverage. After training our model on the proprietary Pistachio [3] data, we increased the class diversity of the predictions to an average of 5.3 for each reaction target compared to 1.9 of the pristine model, while retaining a high value of 62% for the round-trip accuracy of the disconnections.

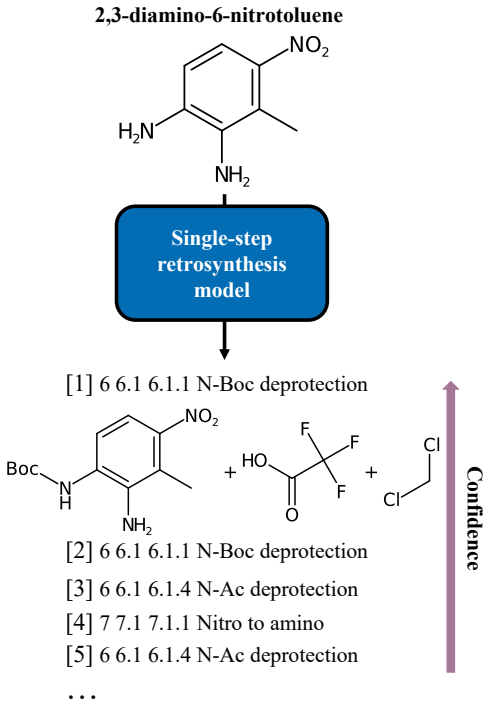


Figure 1: Classes of the single-step baseline predictions for the 2,3-diamino-6-nitrotolune molecule, as produced by Schwaller et al. [21]. As can be noted, all but one are different forms of deprotection ordered by model confidence.

# 2 Results and discussions

## 2.1 Introducing the cluster token prompt

We built our one-step retrosynthesis model out of the Transformer model [29, 21, 26]. Transformer models learn a representation of each token in the input string. To represent molecules, we use the simplified molecular-input line-entry system (SMILES) language [30, 31], where atoms and bonds are codified as specific combinations of text characters. Schwaller et al. [20] developed the tokenization regex used to tokenize the SMILES. Examples of SMILES strings can be found in Figure 2. The embeddings learned for each token depend on the context, which allows the model to encode much more subtle information than a pure one-hot encoding of an atom or bond. To increase diversity, we prepended during training a new token, corresponding to the chemical class of the reaction, in front of each input SMILES product molecule. The classification was the one provided by the NameRXN classification software and is as following: 0→Unrecognized, 1→Heteroatom alkylation and arylation, 2→Acylation and related processes, 3→C-C bond formation, 4→Heterocycle formation, 5→Protections, 6→Deprotections, 7→Reductions, 8→Oxidations, 9→FGI, 10→FGA, 11→Resolutions. The data-preprocessing procedure for training can be visualized in Figure 2 (top).

At test time, the input product molecule can be concatenated to all the available

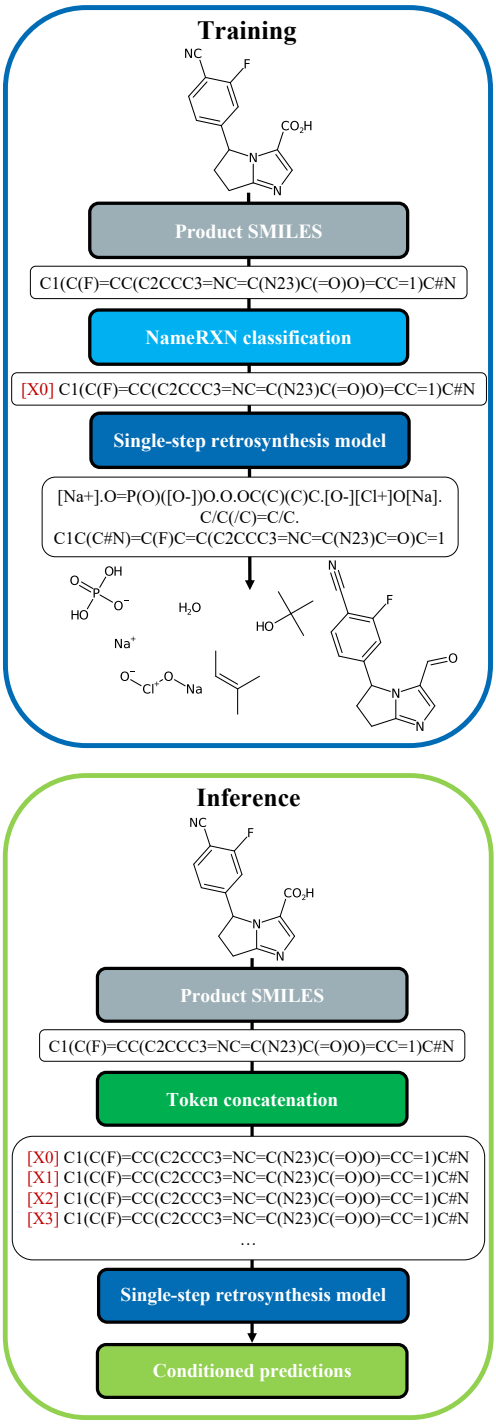


Figure 2: Top: Data-preprocessing procedure for training. The cluster token is prepended to the product SMILES of the reaction and the set of precursors is used as the target. Bottom: Data-preprocessing procedure for inference. A new string is generated for each molecule where each different available cluster token is concatenated. Conditioned predictions are then collected for each molecule.

cluster tokens derived from the classification schema (see Figure 2, bottom), generating $X$ equivalent inputs, where $X$ is the number of cluster tokens used. The first token seen by the transformer is the cluster token. This will steer the predictions towards typical disconnections for that class. Collecting all the top1 predictions for the $X$ class-

tokens (and possibly additional predictions for each of the $X$ class-tokens) leads to a set of disconnections more diverse than the top$N$ outputs of a regular Transformer model, which we use as a baseline. The advantage of this strategy is that the steering acts as a weak influencer of the predictions, rather than a forcing term, such as using a certain template, which can either lead to failure or success. In comparison to the baseline model, the cluster token prompt approach allows the model to "select" from a limited pool of options while yet leaving it with much flexibility. In the following section, we present our models and the results in more details.

## 2.2 High diversity single-step retrosynthesis models

As a training corpus, we utilized the proprietary reaction dataset Pistachio [3], consisting of 2'447'596 unique reactions with both precursors and products in SMILES format. In addition, we tested the procedure on the public dataset USPTO 50k [14], processed by Ramsundar et al. [17], which we provide together with the code (available as additional material). Results for this dataset can be found in Appendix 6.
The data were first suitably pre-processed (see Section 3.1). We used two ways to produce the cluster tokens to prepend in front of each reaction: the first one relies on the NameRXN classification and the second one on a K-means clustering algorithm. For the K-means clustering, we identified the clusters with the reaction fingerprints [22] (see Section 3.3 for details). The models tested are described below:

- **baseline**: a Transformer model [29, 21] with no cluster-token information.
- **12clusters**: a model that utilizes as tokens all the first level classification available from NameRXN (i.e. classes from '0' to '11').
- **3clustersRandom**: a model built on top of the 12 classes from NameRXN which we grouped randomly in 3 clusters.
- **4clustersRandom**: same as the model above, but with 4 clusters.
- **3clustersKmeans**: this model results from the application of the K-means clustering algorithm with 3 clusters on the 3 dimensions obtained from a PCA analysis of the reaction fingerprints.
- **4clustersKmeans**: same as the model above, but with 4 clusters.
- **optimalKmeans**: in this model, we estimated the optimal PCA dimension for the fingerprints (14) and the optimal number of clusters (10). The procedure is described in Section 3.3.

Once the token was identified for each reaction, it was prepended to the SMILES string with the following format: $[i]$ for $i = 0...X$ (see Figure 2), with $X$ being the number of tokens available in each of the models.

For the models evaluation, we splitted randomly the data into a training/validation/test set with a proportion of 80/10/10 for five different random seeds, and we proceeded as follows:

1. We chose one of the splits randomly and we trained all the cluster token prompt models. We tested them against the validation set and chose the best performing model.
2. Then, we merged the train and validation set for the five different seeds and trained the best prompt-based model plus the baseline model.
3. We compared the so-trained baseline and best models against the test sets.

Each of the trained models, including the baseline, was trained for 260000 steps with 1 GPU (approximately 48 hours of training).

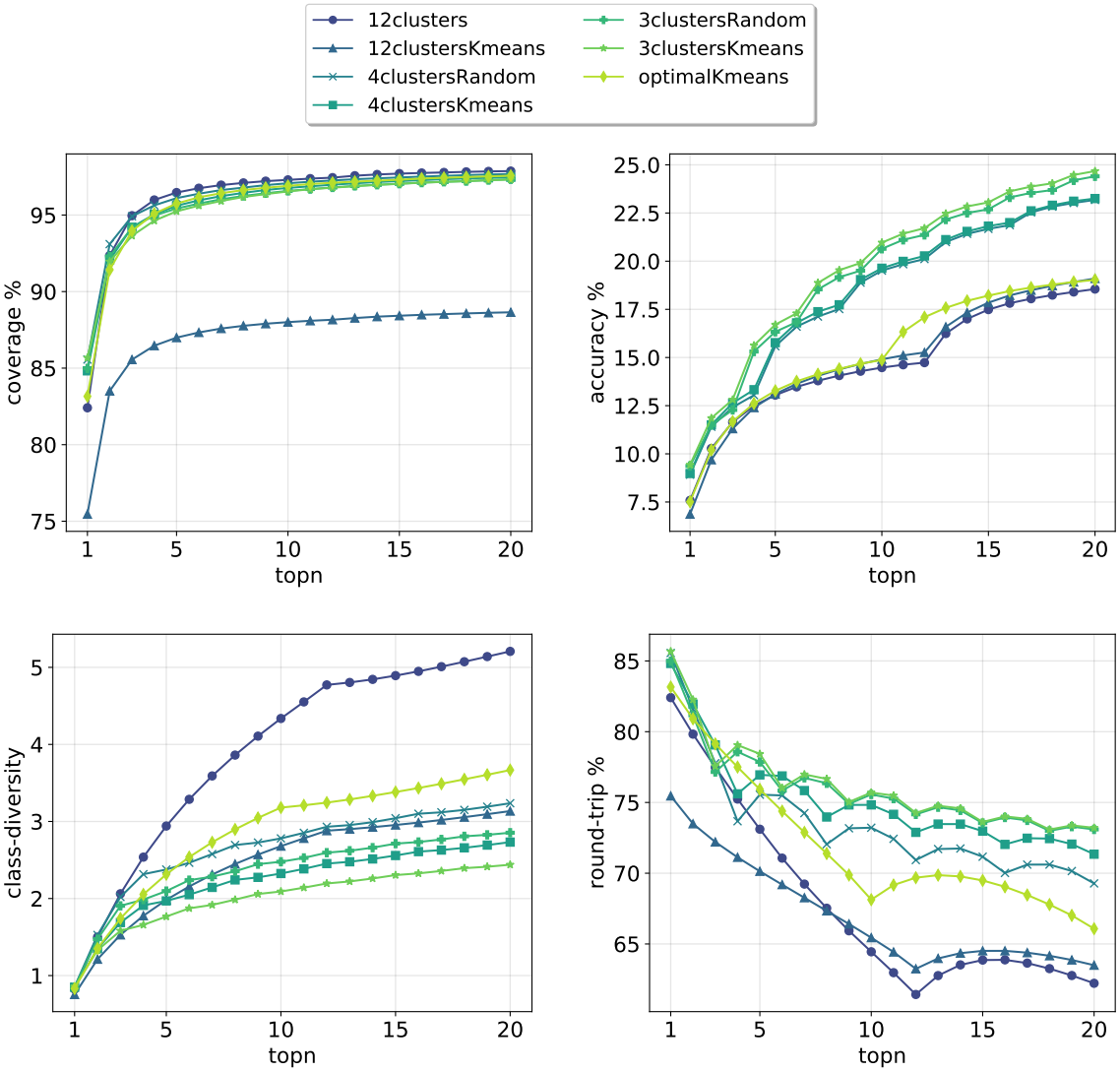


Figure 3: Model metrics. Top left: coverage. Top right: top$n$ accuracy. Bottom left: class diversity. Bottom right: round-trip accuracy.

In Figure 3, we report the results for the prompt-based models evaluated on the validation set. For each model, we retained the top24 predictions as $X * \text{top}k = 24 = \text{top}N$ where $X$ is the number of class tokens for each model and top$k$ is the number of predictions retained for each token-concatenated sample (e.g. for the **12clusters** model, $X = 12$ and top$k = 2$). The plots report 4 metrics of interest as a function of the number of top$N$ predictions analyzed (see Section 3.4 for the metrics definition). To properly compare models, we looked only at top20 predictions (and not top24), as for the **optimalKmeans** model only 20 predictions per sample were produced (2 for each token-conditioned input).

All cluster token models show a good coverage (above 95%) after top3 predictions. The **12clustersKmeans** model is the only one performing poorly from this point of view. Looking at the accuracy, we see that it increases slowly and reaches a top20 value between 18% and 25% for all models. In addition to reactants, our retrosynthesis models predicts a wide range of precursors, and is not limited to the disconnected fragments only. Therefore, many times the ground truth appears with a slightly different set of reagents, justifying the low accuracy values. Accordingly, when a model can produce multiple correct answers, accuracy is not the most crucial metric to consider. We consider the value of the round-trip accuracy to be more interesting (see Section 3.4). This value measures the ability to recover the input molecule by running a forward reaction model on top of the predicted precursors (details on the forward model are in Section 3.2). This metric decreases with the number of top$N$ predictions considered. The decay is more consistent for models utilizing a greater number of tokens (**12clusters**, **12clustersKmeans**). Note that this is to be expected, since we are asking for disconnection conditions that may be impossible to satisfy for some input molecules. However, a high value of coverage guarantees at least one proposed valid disconnection per input molecule. It is important to note that round-trip accuracy does not take into consideration that the top20 predictions for a sample, even if correct, can all collapse into one. This happens for example if the model predicts an identical set of reactants (i.e. molecules into which the target is disconnected) and a different solvent. For this reason, the final metric that we report, the class diversity, is perhaps the most interesting one. It measures the average of the different (NameRXN) classes predicted for a given input, considering only the valid predictions (see Section 3.4). The value highly depends on the number of cluster tokens used and differs from one strategy (NameRXN) to the other (K-means clustering). Using more tokens results in more diversity in the predictions (5.2 for the **12clusters** model at top20 predictions), but also a higher number of incorrect predictions. The **12clustersKmeans** model instead loses in round-trip accuracy without a relevant compensation on the class diversity side. The most interesting models are the **12clusters**, from the point of view of the increased class diversity, and the **optimalKmeans**, which reaches decent values of class diversity and could be used also in a setting where the reaction classification labels are not available.

In a second step, we chose the best models (**12clusters** and **optimalKmeans**), and

| Model | Coverage | Accuracy | Round-trip accuracy | Class diversity |
|---|---|---|---|---|
| **Baseline** | 96.58 ± 0.06 % | **28.28 ± 0.05 %** | **79.50 ± 0.68 %** | 1.90 ± 0.01 |
| **optimalKmeans** | 97.69 ± 0.04 % | 19.02 ± 0.47 % | 66.27 ± 0.95 % | 3.67 ± 0.02 |
| **12clustersKmeans** | **97.94 ± 0.06 %** | 18.42 ± 0.31 % | 62.03 ± 0.53 % | **5.27 ± 0.05** |

Table 1: Comparison of the prompt-based models against the baseline on the test set. Uncertainity bounds are computed based on the standard error and reported in the table.

compared their performance against the baseline. We evaluated our models on five randomly chosen test splits, where, this time, the validation set was included in the training. The results on the top20 predictions are reported in Figure 4. As can be seen, the prompt-based model does indeed boost the diversity of the predictions. On the test set, we achieve an average boost of class diversity of about 3.4 points for the **12clusters** model. For completeness, we report in Appendix 7 the behaviour of the baseline model and the best models as a function of the top$n$ predictions, with standard errors.

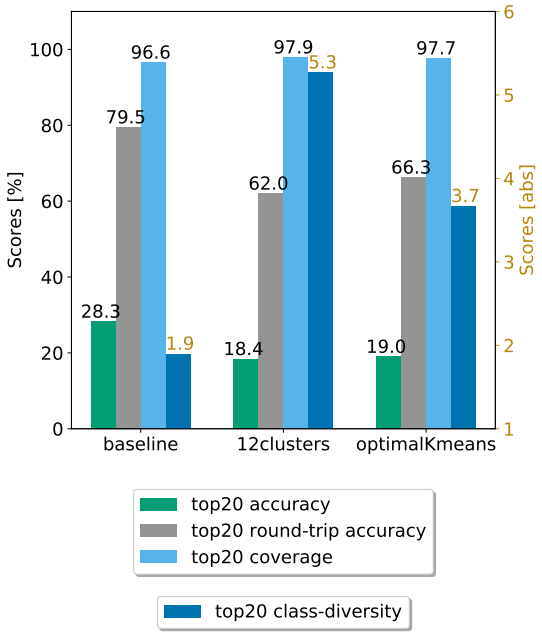


Figure 4: Final comparison of the best prompt-based models and the baseline against the test set. The values of the metrics reported are averaged across five random seeds. For convenience, standard error values are reported in Table 1 .

Table 1 shows the (top20) metrics with standard error bounds for the three models under consideration, generated from the five different random seed experiments.

For comparison, we report in Figure 5 an example of prediction with the baseline model and the **12clusters** model. While for the **baseline** the proposed disconnections all belong to the class of Saponification reactions (6), for the **12clusters** model we observe much

more diversity in terms of reaction classes. Also, looking at the main reactants generated, the prompt-based model proposes different alternatives (e.g. Acylation reaction versus Saponification).

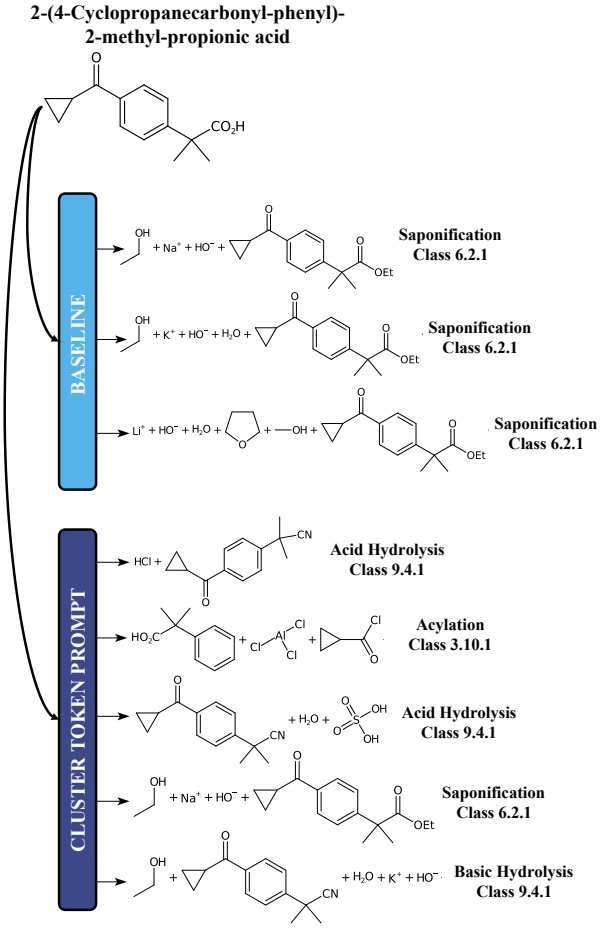


Figure 5: A chemical example predictions with the baseline retrosynthesis model and the prompt-based model.

# 3 Methods

## 3.1 Data

The reaction data set used was the proprietary Pistachio [3], derived by text-mining chemical reactions in US patents. All reactions went through a cleaning procedure, outlined below (the RDKit library was used [12]):

- removal of duplicates and invalid reactions
- merge reactants and reagents: in chemistry reactants are the main actors in the reaction, but they are helped by other molecules that allow the reaction to take place (e.g. solvents) even if not contributing atoms to the final product. In our

work we merged reactants and reagents (also known as 'precursors') on the left hand side of the reaction (e.g. `A>B>C` → `A.B>>C`).

- set on the precursors: given no real relationship between the number of times a molecule appears in the patent reaction and the stoichiometry, we made molecules unique.
- removal of multi-products reactions: this operation was performed after removing residual precursors molecules from the product side.
- removal of reactions where the product contains atom types not present in the precursors side.
- removal of single-atom products.
- removal of reactions where the absolute formal charge exceeded the value of 2.
- removal of reactions where the maximum number of tokens was above 500.
- removal of reactions with the same set of precursors, but different products.

We provide the already cleaned public dataset [14] together with the code.

The cleaned data set was randomly split into training, test and validation sets (80%/10%/10%) for five different random seeds. One of these splits was used to choose the best cluster token model, while the comparison to the baseline was performed against all five random seeds, merging validation and train set.

## 3.2 Models

Our Deep Learning approach to single-step retrosynthesis does not rely on reaction templates and takes into consideration both reactant and reagents as the target set. We formulate the problem of going from the product to the target precursors as a machine translation task, similar to Schwaller et al. [21]. The molecules were codified as SMILES strings, tokenized, and fed to the Transformer model [29]. We used the OpenNMT framework [10] and PyTorch [16] to build the models. The hyperparameters were the same used in related work [21, 27] and were kept fixed throughout all simulations. The transformer is made up of a set of encoder layers and a set of decoder layers. The tokens of the input SMILES string are encoded into (learned) hidden vectors by the encoder. Those vectors are then fed to the decoder to predict the output sequence, one token at a time. The model size and hyperparameters where taken from previous literature [20]. The number of layers in both the encoder and decoder was set to 4 (size 384). The main characteristic of the transformer is the presence of multi-head attention and the number of these heads was set to 8. Dropout was also included in the model at a rate of 0.1. An Adam optimizer was used for loss minimization and the starting learning rate was set to 2. An exhaustive file with all the parameter values used can be found in the code.

### 3.2.1 Forward and Classification models

To better evaluate the single-step retrosynthesis models, two additional models are necessary. The first model is the forward prediction model used for reaction prediction (from precursors to product). This model was built with the same dataset used for the retrosynthesis one, switching source and target. Training files are available together with the code. The second model is a classification model to classify the retro predictions. For this, we also relied on transformers. The procedure is the one of Schwaller et al. [22], model 'Transformer enc4-dec1', applied to the same reaction dataset as the retro and forward model.

## 3.3 K-means analysis

To evaluate whether adequately conditioned predictions can be obtained without relying on ad-hoc classification, we generated the conditioning tokens starting from reaction fingerprints [22] and applied a K-means clustering algorithm. Since the fingerprints live in a high-dimensional space, we first reduced their dimension with principal component analysis (PCA). To choose the best number of components we performed a variance analysis and identified the components which capture the greatest amount of variance in the data (see Figure 6).

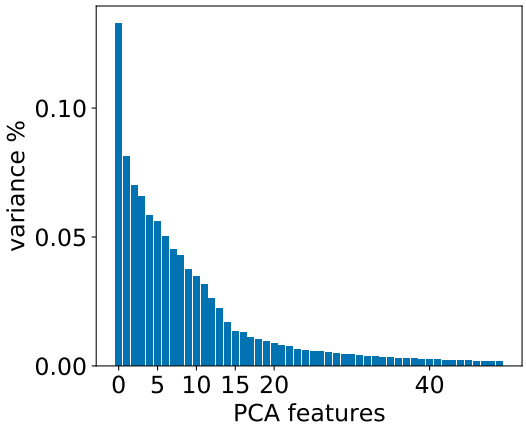


Figure 6: Relevant PCA components analysis. Two drops in the variance are observed around the 2nd/3rd component and a smaller one around the 14th component.

For the **12clustersKmeans**, **3clustersKmeans** and **4clustersKmeans** models, we kept only the first three components. For the **optimalKmeans** model we shot further and included all the first 14 components. Subsequently, for the K-means clustering, we relied on a fixed number of clusters for the first models (**12clustersKmeans**, **3clustersKmeans** and **4clustersKmeans**). On the other hand, for the **optimalKmeans** model, we first performed an analysis to determine the optimal grouping [7]. This can be done by measuring the sum of the squared distances to the nearest cluster center (inertia). This

allows computing a plot of the inertias against the number of clusters used. The optimal $k$ is said to coincide with the elbow of the plot, where the inertia value change starts to be less significant. The inertia plots can be found in Appendix 8.

Figure 7 shows the clusters generated for the **optimalKmeans** model. The plots for the other K-means-models can be found in Appendix 8.

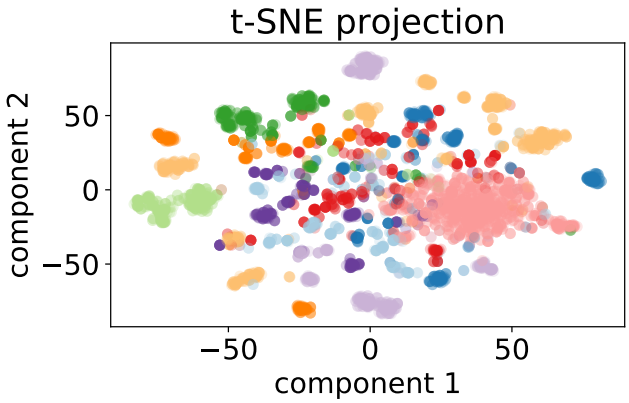


Figure 7: t-SNE projection for 50000 samples of the **optimalKmeans** model. The different colours represent the different Kmeans clusters.

## 3.4 Metrics

For the single-step retrosynthesis model evaluation we computed four different metrics. The first one is the top$n$ accuracy, which is over-evaluated for this kind of task for the reasons we have explained before. In this section, we explain in more detail the other three metrics considered [21, 27].

### 3.4.1 Round-trip accuracy

The round-trip accuracy metric, unlike the top$n$ accuracy, takes the top$k$ predictions for a molecule and applies on top of them a forward prediction model (see Section 3.2). If the original molecule is recovered through the forward model, then that reaction contributes positively to the accuracy. This is also how we define a prediction to be valid. More specifically, given $X = \{(x_i, y_i)...(x_N, y_N)\}$ our dataset of $N$ target products with target precursors, we define the top$k$ round-trip accuracy ($RT_k$) as follows:

$$RT_k(X) = \frac{1}{N * k} \sum_{i=0}^{N} \sum_{j=0}^{k} x_{i,j} \equiv F(R(x_{i,j})) \tag{1}$$

Where $x_{i,j}$ is the $j$th prediction for the $i$th sample, $R$ is the retrosynthesis model and $F$ is the forward translation model.

### 3.4.2 Class diversity

Class diversity is the most interesting metric in our analysis. It measures the average number of reaction classes predicted for each target product molecule. For example, considering the first $k$ predictions for a set of test molecules, a class diversity of 5 means that, on average, the (valid) precursors predictions belong to five different reaction classes. Here "valid" is in the sense of round-trip accuracy. Again, in mathematical terms, given $X = \{(x_i, y_i)...(x_N, y_N)\}$ our dataset of $N$ target products with target precursors, we define the top$k$ class diversity ($CD_k$) as follows:

$$CD_k(X) = \frac{1}{N}\sum_{i=0}^{N} |set(\{C(R(x_{i,j}), x_{i,j})\}_{j=1}^{k} | x_{i,j} \equiv F(R(x_{i,j})))| \tag{2}$$

Where $set()$ is the set operation on the elements and $||$ is the set cardinality. As above, $x_{i,j}$ is the $j$th prediction for the $i$th sample, $R$ is the retrosynthesis model and $F$ is the forward translation model. $C$ is the classification model used to predict the classes (see Section 3.2).

### 3.4.3 Coverage

The coverage is the fraction of test samples for which there exists at least one valid prediction (in the round-trip accuracy sense). Given $X = \{(x_i, y_i)...(x_N, y_N)\}$, our dataset of $N$ target products with target precursors, we define the top$k$ coverage ($CV_k$) as follows:

$$CV_k(X) = \frac{1}{N}\sum_{i=0}^{N} any(\{x_{i,j} \equiv F(R(x_{i,j}))\}_{j=1}^{k}) \tag{3}$$

Where $any()$ outputs 1 if at least one valid prediction exists among the top$k$ for that sample.

# 4 Data and code availability

The code used to train the high diversity models can be found at https://github.com/rxn4chemistry/rxn_cluster_token_prompt. Moreover, we provide the cleaned open-source dataset on which it is possible to reproduce the procedure, as well as the models trained on USPTO (details in the GitHub repository). Results for the open-source dataset are reported in Appendix 6.

The cluster token prompt models trained with Pistachio are also accessible through the IBM RXN for Chemistry website [1].

# 5 Conclusions and Outlook

Exploration and diversity are at the heart of any application of language models to retrosynthesis algorithms. Current retrosynthesis models focus mainly on predicting the reported ground truth, and do not take into account the ability to generate alternatives. Our work is the first approach tackling and analysing diversity directly. We have presented a cluster token prompt-based model that effectively increases diversity in predictions for single-step retrosynthesis. In addition to improving on other measures, our approach can increase class variety by a factor of two or more over the baseline. Incorporating a diversity-boosted single-step retrosynthesis model, into a multi-step pipeline (for example, Beam Search) to recursively build disconnection trees, offers a set of very diverse reactions from which to choose. This strategy improves the search for less obvious and more engaging paths. It becomes even more of interest in an interactive framework where chemists can be assisted by AI to plan their retrosynthetic route relying on a wide variety of chemical disconnection recommendations and indirectly less bias.

# Ethics Statement

This material is the authors' own original work, which has not been previously published elsewhere. The manuscript is not currently being considered for publication elsewhere and it reflects the authors' own research and analysis in a truthful and complete manner.

# Acknowledgments

This publication was created as part of NCCR Catalysis (grant number 180544), a National Centre of Competence in Research funded by the Swiss National Science Foundation.

# References

[1] IBM RXN for chemistry. URL URLhttps://rxn.res.ibm.com. https://rxn.res.ibm.com. Accessed: Oct 1, 2022.

[2] Nextmove software. NameRxn, 2022. URL https://www.nextmovesoftware.com/namerxn.html. https://www.nextmovesoftware.com/namerxn.html. Accessed: Oct 1, 2022.

[3] Nextmove software Pistachio, 2022. URL http://www.nextmovesoftware.com/pistachio.html. http://www.nextmovesoftware.com/pistachio.html. Accessed: Oct 1, 2022.

[4] Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: Learning retrosynthetic planning with neural guided A* search. In *The 37th International Conference on Machine Learning (ICML)*, 2020. doi: 10.48550/ARXIV.2006.15820.

[5] Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS Central Science*, 3(12): 1237–1245, 2017. doi: https://doi.org/10.1021/acscentsci.7b00355.

[6] Connor W. Coley, Dale A. Thomas, Justin A. M. Lummiss, Jonathan N. Jaworski, Christopher P. Breen, Victor Schultz, Travis Hart, Joshua S. Fishman, Luke Rogers, Hanyu Gao, Robert W. Hicklin, Pieter P. Plehiers, Joshua Byington, John S. Piotti, William H. Green, A. John Hart, Timothy F. Jamison, and Klavs F. Jensen. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453), 2019. doi: 10.1126/science.aax1566.

[7] Dmitriy. Principal component analysis and k-means clustering to visualize a high dimensional dataset. URL https://medium.com/more-python-less-problems/principal-component-analysis-and-k-means-clustering-to-visualize-a-high-dimensior Blog post (accessed: May 2022).

[8] Jingxin Dong, Mingyi Zhao, Yuansheng Liu, Yansen Su, and Xiangxiang Zeng. Deep learning in retrosynthesis planning: datasets, models and tools. *Briefings in Bioinformatics*, 23(1):bbab391, 2021. doi: https://doi.org/10.1093/bib/bbab391.

[9] Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv:1909.05858, Version 2*, 2019. doi: https://doi.org/10.48550/arXiv.1909.05858.

[10] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics, 2017. URL https://aclanthology.org/P17-4012.

[11] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1378–1387, 2016. doi: 10.48550/ARXIV.1506.07285.

[12] Greg Landrum, Paolo Tosco, Brian Kelley, Sriniker, Gedeck, NadineSchneider, Riccardo Vianello, Andrew Dalke, Ric, Brian Cole, AlexanderSavelyev, Samo Turk, Matt Swain, Alain Vaucher, Dan N, Maciej Wójcikowski, Axel Pahl, JP, Francois Berenger,

Strets123, JLVarjo, Noel O'Boyle, David Cosgrove, Patrick Fuller, Jan Holst Jensen, Gianluca Sforna, DoliathGavid, Karl Leswing, Susan Leung, and Jeff Van Santen. rdkit/rdkit: 2019_03_4 (q1 2019) release, 2019.

[13] Cheng-Hao Liu, Maksym Korablyov, Stanisław Jastrzebski, Paweł Włodarczyk-Pruszyński, Yoshua Bengio, and Marwin Segler. RetroGNN: Fast estimation of synthesizability for virtual screening and de novo design by learning from slow retrosynthesis software. *Journal of Chemical Information and Modeling*, 62:2293–2300, 2022. doi: https://doi.org/10.1021/acs.jcim.1c01476.

[14] Daniel Mark Lowe. Extraction of chemical structures and reactions from the literature. 2012. doi: 10.17863/CAM.16293.

[15] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv:1806.08730, Version 1*, 2018. doi: 10.48550/ARXIV.1806.08730.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://pytorch.org.

[17] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837.

[18] Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *Computing Research Repository*, arXiv:2001.07676, Version 3, 2020. doi: 10.48550/ARXIV.2001.07676.

[19] Nadine Schneider, Nikolaus Stiefl, and Gregory A. Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *Journal of Chemical Information and Modeling*, 56(12):2336–2346, 2016. doi: 10.1021/acs.jcim.6b00564.

[20] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019. doi: https://doi.org/10.1021/acscentsci.9b00576.

[21] Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H. Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical Science*, 11:3316–3325, 2020. doi: https://doi.org/10.1039/C9SC05704H.

[22] Philippe Schwaller, Daniel Probst, Alain C. Vaucher, Vishnu H. Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence*, 3(2):144–152, 2021. doi: https://doi.org/10.1038/s42256-020-00284-w.

[23] Marwin H. S. Segler and Mark P. Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry - A European Journal*, 23(25): 5966–5971, 2017. doi: https://doi.org/10.1002/chem.201605499.

[24] Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555(7698):604–610, 2018. doi: 10.1038/nature25978.

[25] Vignesh Ram Somnath, Charlotte Bunne, Connor Coley, Andreas Krause, and Regina Barzilay. Learning graph models for retrosynthesis prediction. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 9405–9415. Curran Associates, Inc., 2021. doi: https://doi.org/10.48550/arXiv.2006.07038.

[26] Igor V. Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature Communications*, 11(1), 2020. doi: 10.1038/s41467-020-19266-y.

[27] Alessandra Toniato, Philippe Schwaller, Antonio Cardinale, Joppe Geluykens, and Teodoro Laino. Unassisted noise reduction of chemical reaction datasets. *Nature Machine Intellingence*, 3:485–494, 2021. doi: https://doi.org/10.1038/s42256-021-00319-w.

[28] Zhengkai Tu and Connor W. Coley. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction. *arXiv:2110.09681, Version 1*, 2021. doi: https://doi.org/10.48550/arXiv.2110.09681.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. doi: https://doi.org/10.48550/arXiv.1706.03762.

[30] David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling*, 28(1):31–36, 1988. doi: https://doi.org/10.1021/ci00057a005.

[31] David Weininger, Arthur Weininger, and Joseph L. Weininger. SMILES. 2. algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences*, 29(2):97–101, 1989. doi: https://doi.org/10.1021/ci00062a008.

# Appendix

## 6 Open-source dataset results

The procedure applied to the proprietary Pistachio dataset [3] in the main manuscript was also applied to an open-source dataset, USPTO 50k [14], for reproducibility reasons. This dataset was chosen because it is the only open-source dataset with public chemical reactions classification, performed by Schneider et al. [19].

The whole data processing procedure, the dataset, the scripts and the models are available with the code. For this smaller dataset, we built three random models and three models based on clustering of reaction fingerprints. We used 2, 5 and 10 tokens for the clustering. As for Pistachio, we chose the best cluster token prompt-based models by comparing them against the validation set. We concluded the analysis with the confrontation against the baseline on five random seeds on the test set.

In Figure 8, we compare the cluster token prompt-based models trained on USPTO 50k, while in Figure 9, we compare the final best models.
Differently from the results with Pistachio we notice that the models can better predict the ground truth precursors. It is to be noted that USPTO 50k is a smaller dataset where only reactants and not reagents are reported (differently from Pistachio), so the training task is much easier than with Pistachio. At the same time, though, the round-trip accuracy has a quite low value, even if the forward model for the evaluations was trained with the same USPTO 50k dataset and reached an accuray of 77.46% (and 95.29% accuracy on the classification model). This behaviour can be ascribed to the fact that the dataset is too small and it is not able to generalize sufficiently well.

Looking at Figure 9, we see that for the **10clusters** model, corresponding to using all the reaction classes ids as single tokens, the class diversity increases to 3.1. The best top20 accuracy as well as the round-trip accuracy is reached by the **10clustersKmeans** model.

We also report the standard error values at top20 predictions for the best models, computed with the same random seeds. The values can be found in Table 2. We observe that the error bar is more significant for the open-source models. This can be ascribed to the smaller dataset. Indeed, for only 50k data points we cannot create sufficiently general splits as for the 2 million data samples from Pistachio. The **10clustersKmeans** model is the best compromise through all metrics.

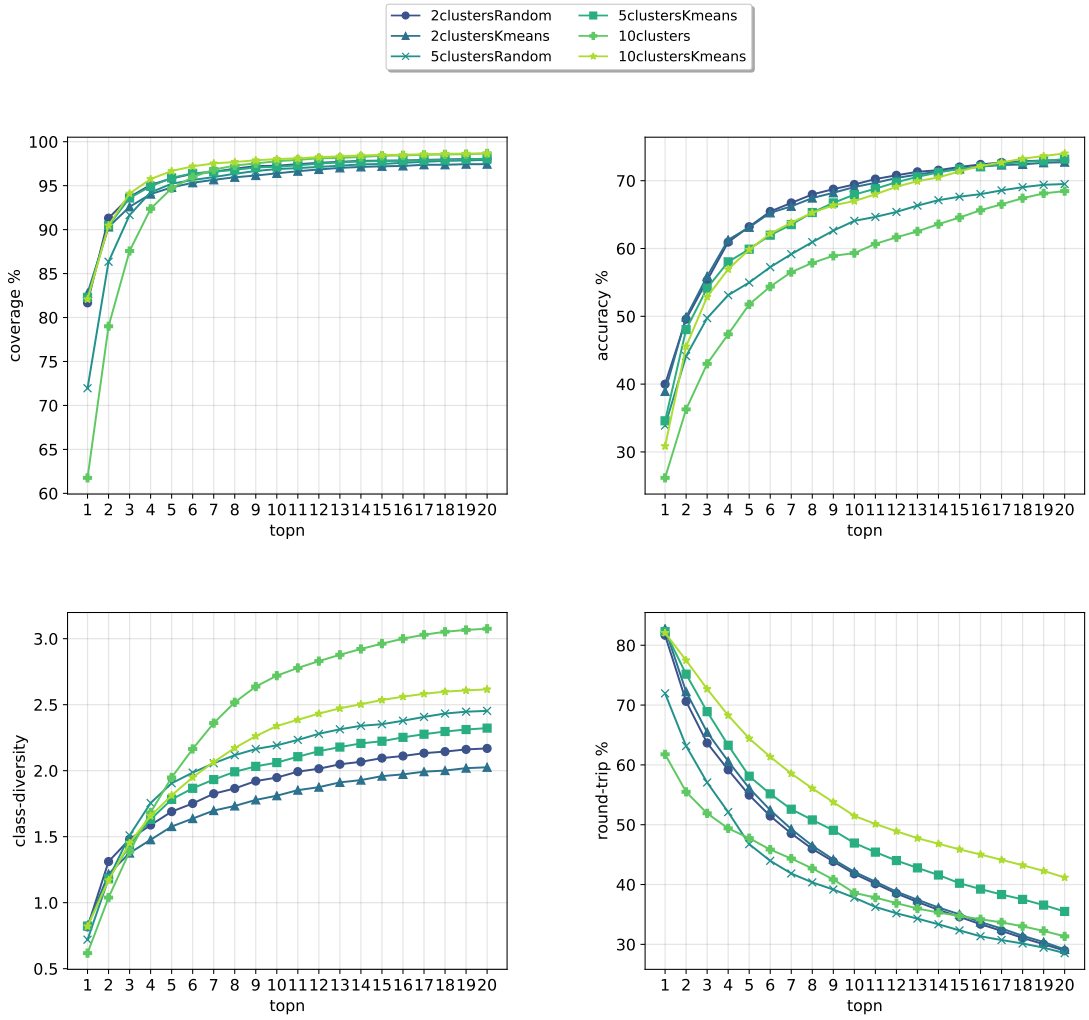


Figure 8: Metrics for the models trained on USPTO. Top left: coverage. Top right: top$n$ accuracy. Bottom left: class diversity. Bottom right: round-trip accuracy.

# 7 Baseline and other plots

Figure 10 shows the values for the metrics of interest for the baseline model. The shades mark the standard error bounds for class diversity and round-trip accuracy.

The same plots are reported in Figure 11 and 12 for the **12clusters** and the **optimalKmeans** models. For all models, it can be observed that the standard error on the class diversity is quite high, changing a lot across compounds, but it is the same for the cluster token prompt-based models and the baseline.

# 8 Kmeans plots

In this section we report the inertia plots for the K-means algorithm (Figure 13), as well as the clustering plots for all the prompt-based K-means models (Figure 14). The clustering plot for the **optimalKmeans** model can be found in Figure 7.

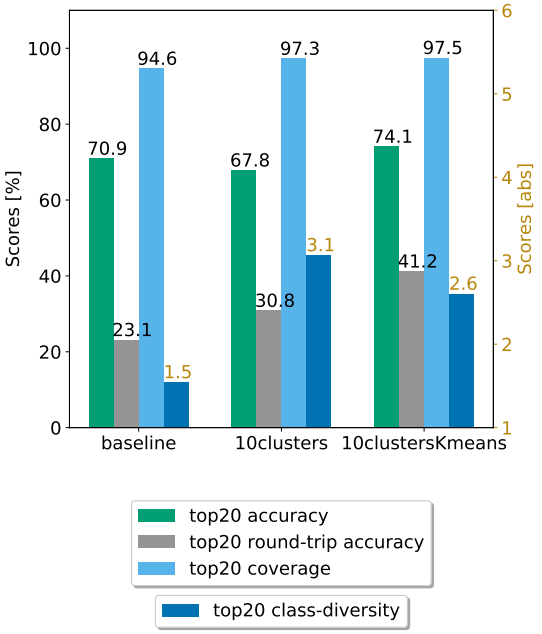


Figure 9: Final comparison of the best cluster token prompt-based models and the baseline against the test set for the open source dataset. The values of the metrics reported are averaged across 5 random seeds. For convenience, standard error values are reported in Table 2 .

| Model | Coverage | Accuracy | Round-trip accuracy | Class diversity |
|---|---|---|---|---|
| **Baseline** | 94.64 ± 0.97 % | 70.94 ± 0.31 % | 23.13 ± 0.46 % | 1.54 ± 0.29 |
| **10clusters** | 97.28 ± 0.10 % | 67.84 ± 0.47 % | 30.84 ± 0.65 % | **3.06 ± 0.07** |
| **10clustersKmeans** | **97.49 ± 0.15 %** | **74.09 ± 0.17** % | **41.21 ± 0.69** % | 2.60 ± 0.04 |

Table 2: Comparison of the cluster token prompt-based models for USPTO 50k against the baseline on the test set. Uncertainity bounds are computed based on the standard error and reported in the table.

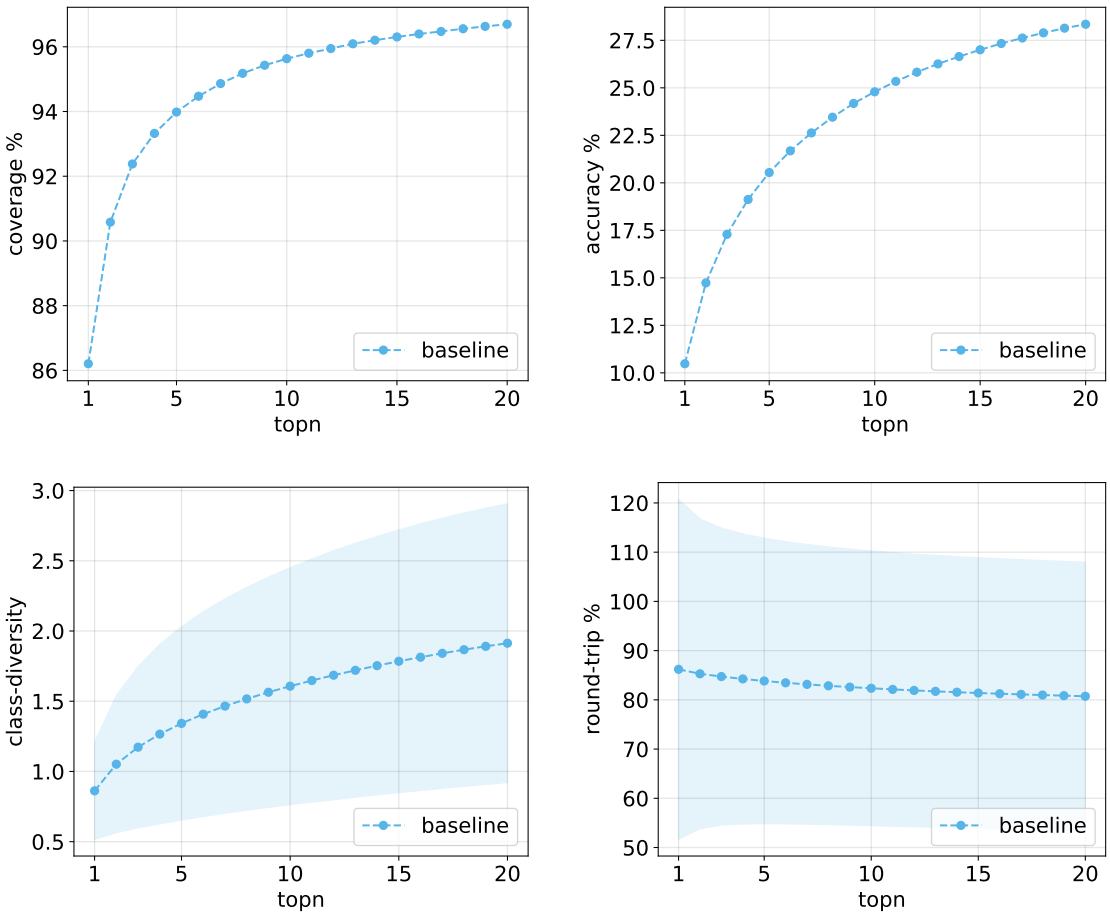


Figure 10: Metrics for the **baseline** model trained on Pistachio. Top left: coverage. Top right: top$n$ accuracy. Bottom left: class diversity. Bottom right: round-trip accuracy.

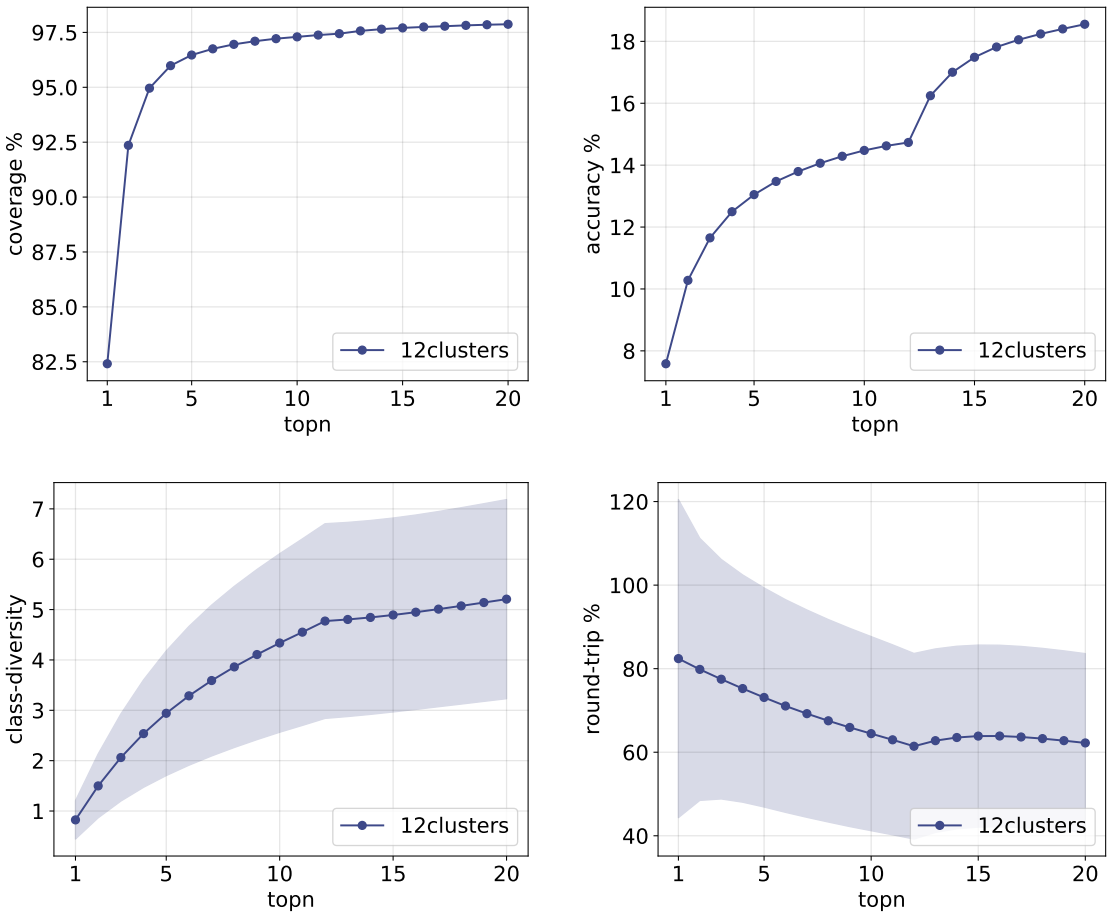


Figure 11: Metrics for the **12clusters** model trained on Pistachio. Top left: coverage. Top right: top$n$ accuracy. Bottom left: class diversity. Bottom right: round-trip accuracy.

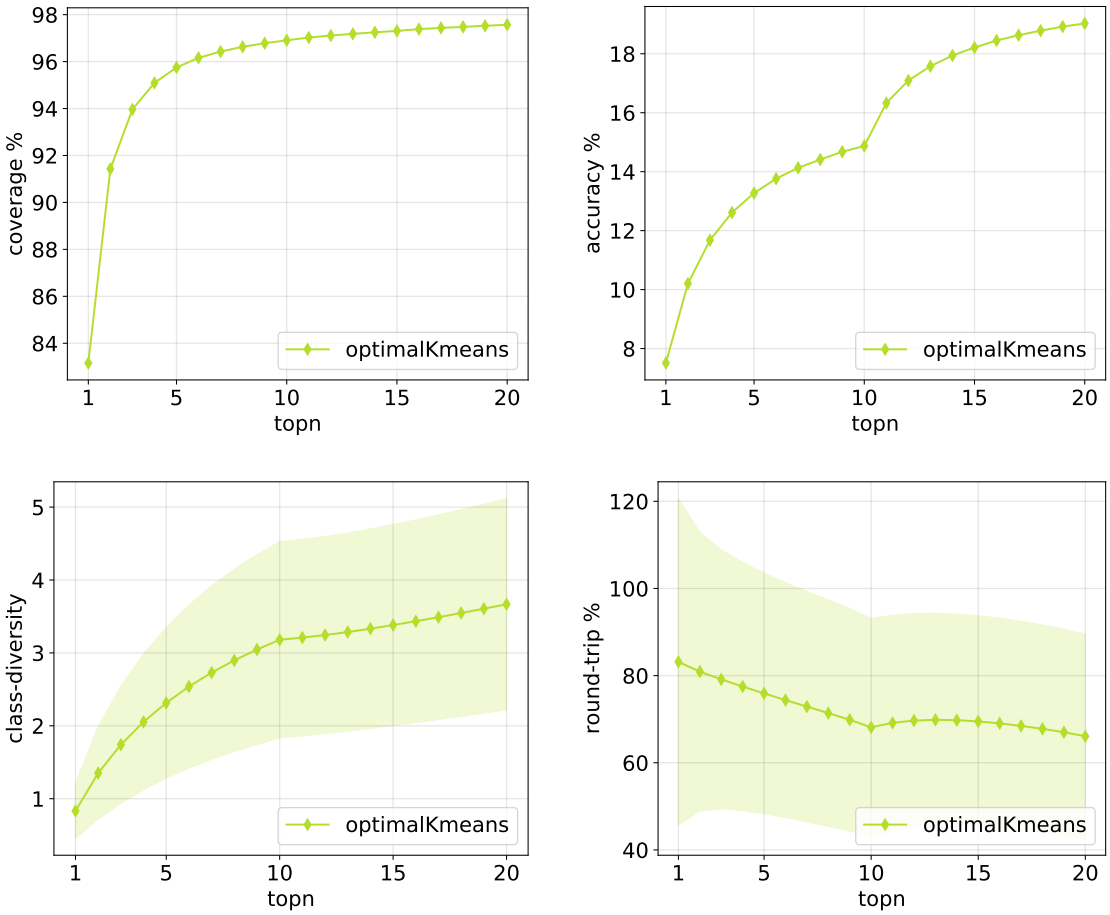


Figure 12: Metrics for the **optimalKmeans** model trained on Pistachio. Top left: coverage. Top right: top$n$ accuracy. Bottom left: class diversity. Bottom right: round-trip accuracy.

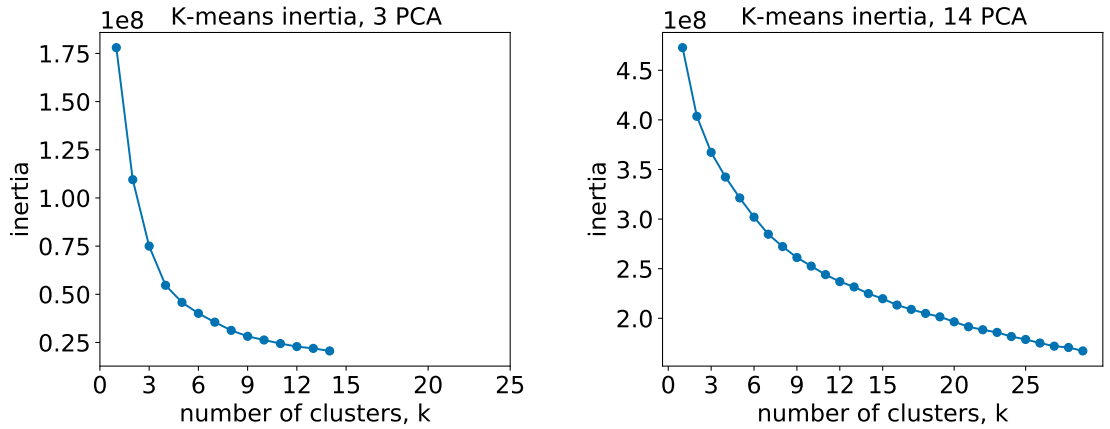


Figure 13: Inertia plots for the K-means clustering. Different K-means algorithms with increasing number of clusters were run on both the 3-components fingerprints (left) and the 14-components fingerprints (right).

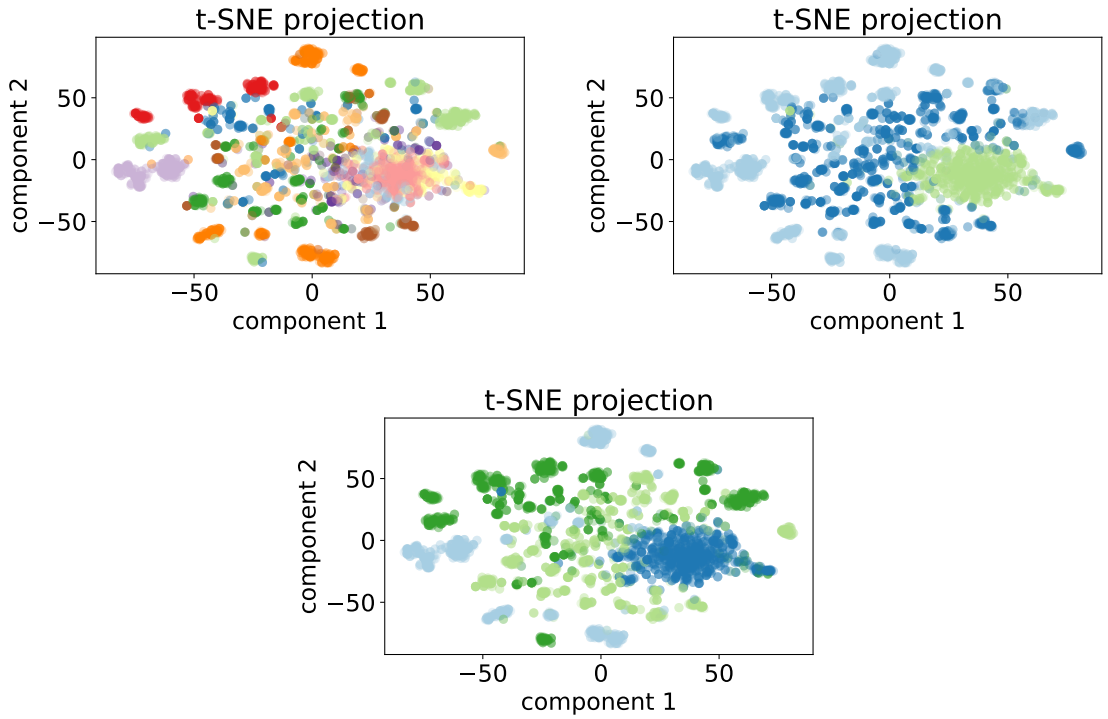


Figure 14: Top left: t-SNE projection for 50000 samples of the **12clustersKmeans** model. Top right: t-SNE projection for 50000 samples of the **3clustersKmeans** model. Bottom: t-SNE projection for 50000 samples of the **4clustersKmeans** model.