Automatically Explaining Fuzzy Cognitive Map (FCM) Simulations

Anish Shrestha, Angela Famera, Josh Crail

December 8, 2022

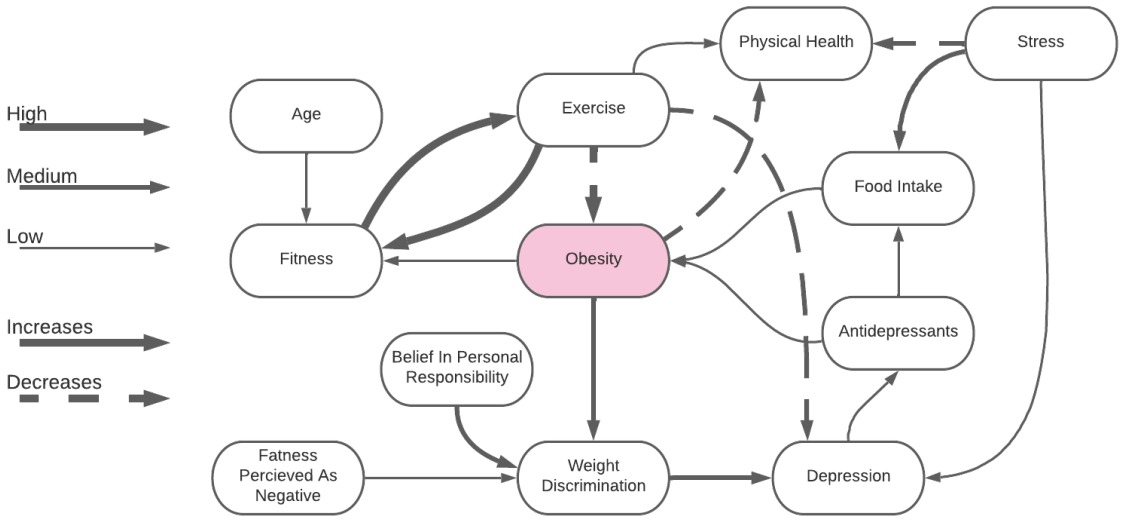

Figure 1: Obesity FCM

1 Introduction

Simulation
FCMs

2 Background

2.1 GPT-3

2.2 Graph-to-Text NLG Task

A large portion of the project is to be able to take a graph as input and convert it into a coherent and accurate textual representation of the information from the graph. Essentially, there is a graph containing information about relationships between attributes of information, and we want to convert it into text form without needing human intervention to manually translate it. The term "NLG" refers to "natural language generation", which means that we want text that not only gives an accurate representation of the information in the graph, but that also remains coherent and readable as though it were written by a human.

2.3 Table-to-Text NLG Task

2.4 Simulation

Fuzzy Cognitive Maps (FCMs) can be visualized as cognitive maps encompassing Fuzzy logic. A cognitive map is a mental representation of a central concept. These maps consist of components (nodes) and relationships (edges). Cognitive maps also maintain a standard weight when describing the relationship between nodes [1] (i.e. 1.0/0.0 for increase/decrease or 1.0/0.0 for positive/negative impact). FCM's, on the other hand, contain a degree of influence (Figure 1). This degree of influence measures how much one node or component impacts the other (whether that be a positive or negative relationship), and can have values between 0.0 and 1.0 [1].

There are two approaches of using an FCM within the FCM community [2]. First one is called *Causal* approach, where the values of concepts are pre-activated before the simulation with baseline starting values, and the model is run to equilibrium. Such scenario that are regarded as the *current* or *reference* situation is called a baseline scenario. To experiment different scenarios, the simulation analysis is performed by applying an external pressure or interventions(s) onto the baseline to see how rest of the factors in the FCM changes. Second approach is called *Dynamical* approach **??**, which is used to study how relative change in factor value(s) affects other factors. It helps to understand which factors are most influenced in a system, where large factor value changes signifies most influenced. Commonly, most factor values are initialized as zero, meanwhile the factor, whose influence is to be studied, is set to one. Among the two appraoches, we study FCM using causal approach.

2.5 Fine-tuning with GPT-3

GPT-3 uses few-shot learning to generate a plausible completion (i.e. output) given few examples and a related prompt. Fine-tuning improves this process by training on many more examples than can fit in the prompt. To fine-tune, the data needs to be in the format of a single input ("prompt") and an associated output ("completion"). Our "prompt" and completion are where our linearized representations come in.

2.6 Linearized Representations

The nature of FCM simulations are quantitative in nature. However, the pretrained models that we use is a sequence-to-sequence machine learning model. So, we need to be able to encode the information of an FCM simulation into linear textual format.

We follow the work of Shrestha *et.al* [3] to represent FCM results in the form of linearized aggregates shown in Section 2.7. We use delimiters using "<" and ">" to signify different properties of subsequent tokens [4]. $<S>$ signifies a subject, $<V>$ a qualitative value, $<E>$ the end, and $<C>$ as change. The process of converting an FCM simulation is explained in the Section 3.1.

2.7 Simulation Scenarios

Our states are stored such that each node in the obesity FCM is associated with a value between 0 and 1 inclusive. We started with the four scenarios consisting of baseline values for each concept node shown in table 1.

After generating 80 different variations of these states (as explained in 3), we produced "Initial" and "Difference" linearized representations known as RDF triples. We use this format because OpenAI recommends using special tokens to start and end prompts and completions [3].

The initial representation is the linearized version of our sample states, and the difference representation is the prediction based on our FCM. These linearized initial representations summarize a scenario, and each scenario has an associated description. For example, the produced initial linearized representation may look like this:

> Initial: <S> Age <V> adult <E><S> Antidepressants <V> elevated <E><S> Depression <V> average <E><S> Belief in personal responsibility <V> negligible <E><S> Exercise <V> very small <E><S> Fatness perceived as negative <V> very small <E><S> Food intake <V> average <E><S> Income <V> little <E><S> Knowledge <V> little <E><S> Obesity <V> average <E><S> Physical health <V> small <E><S> Stress <V> average <E><S> Weight discrimination <V> average <E>

Here, <S>, <V>, and <E> are separators that signify Subject, Verb, and End. The explanation of the scenario would look like:

> The given person is an adult on elevated antidepressants to treat average depression. The person exhibits negligible belief in personal responsibility, participates in very small levels of exercise, and has a very small negative perception of fatness. The person's food intake is average and has little income and knowledge. The person experiences average obesity, small physical health, average stress, and average weight discrimination.

In addition to the initial representation, we have a difference representation. The tokens remain the same as the initial representation, but we have an additional <C> token signifying change. The difference representation may look like this:

> Difference: <S> Antidepressants <C> increase <V> negligible <E><S> Physical health <C> grow <V> very small <E><S> Obesity <C> grow <V> noticeably low <E><S> Food intake <C> grow <V> small <E><S> Weight discrimination <C> increase <V> average <E><S> Exercise <C> increase <V> average <E><S> Belief in personal responsibility <C> grow <V> average <E><S> Fatness perceived as negative <C> grow <V> average <E>

The difference is meant to describe the prediction given by our FCM for a scenario. As a result, for each difference representation, we also wrote an associated explanation. An example explanation is shown below:

> The model predicts a negligible increase in antidepressants and very small growth in physical health. Noticeably low growth in obesity and food intake is predicted to lead to an average increase in weight discrimination. The model, therefore, predicts an average increase in exercise and average growth in personal responsibility. The person's negative perception of fatness is also expecting average growth.

Once we generated all of our linearized representations and their associated scenario explanations, we could run them through GPT-3 to produce our desired output.

2.8 Expected Output

The performance of our model is evaluated based on the BLEU [5] and METEOR [6] score. Similar to [7], these scores are used to compute the surface-level similarity between the generated texts and the human (reference) texts.

In addition to our surface-level quantitative evaluation, our output is expected to generate clear explanations of FCM scenarios, eliminating the need for a subject matter expert to interpret results. Our output should provide a summary of the changes and predication generated by our FCM. We expect our output to encompass the following: paragraph format; easy to read; grammatically correct; and no contradictions. Our output should be in paragraph format, meaning roughly three-six well-written, concatenated sentences. The paragraph should be easy to read and interpret, meaning no obscure or vague language, and in summary, needs to make sense to the reader. The paragraph should be grammatically correct not just to aid with reading comprehension, but also to facilitate translation using tools in the case of a language barrier. For example, if the explanations are written in English, and the person reading them knows English as a second language, they could use a tool to translate the information clearly from one language to another. Last, the paragraph should contain no contradictions. For example, it is unlikely that a person with very high physical and high exercise will see an increase in obesity.

3 Methods

This section will elaborate on the mechanism of explaining simulation scenarios. Section 3.1 elaborates on process of encoding states used in and produced by FCM simulation, also how they are converted to qualitative values.

Key	State 1	State 2	State 3	State 4
Age	0.6	0.8	0.2	0.2
Antidepressants	1.0	0.0	0.75	0.0
Depression	0.8	0.2	0.5	0.0
Belief in personal responsibility	0.8	0.9	0.2	1.0
Exercise	0.5	0.2	0.2	1.0
Fatness perceived as negative	0.7	0.9	0.1	0.8
Food intake	0.4	0.5	0.5	0.2
Income	0.8	0.2	0.3	0.5
Knowledge	0.7	0.9	0.3	0.8
Obesity	0.2	0.2	0.6	0.0
Physical Health	0.75	0.5	0.4	1.0
Stress	0.9	0.1	0.6	0.3
Weight discrimination	0.2	0.0	0.5	0.1

Table 1: Base Scenarios

3.1 Encoding Simulation Scenarios

To observe an impact of a change from the baseline, simulators pick one or more concepts of interests and alter their values to create a scneario. The new set of values are fed into FCM to observe how the simulation ends. We consider two states of the FCM simulations, which are the initial state (the scenario) and final or stabilized state (the simulation output).

As mentioned in section 2.6, we convert initial and final states, which are semi-quantitative in nature, into qualitative form by encoding them into linearized aggregates. The conversion of values into approximate values are done via a dictionary lookup as explained in figure 2 followed by concatenating with delimiters to distinguish between concepts, values, and changes. The linearized aggregate is used as the prompt argument in the dataset while using GPT-3.

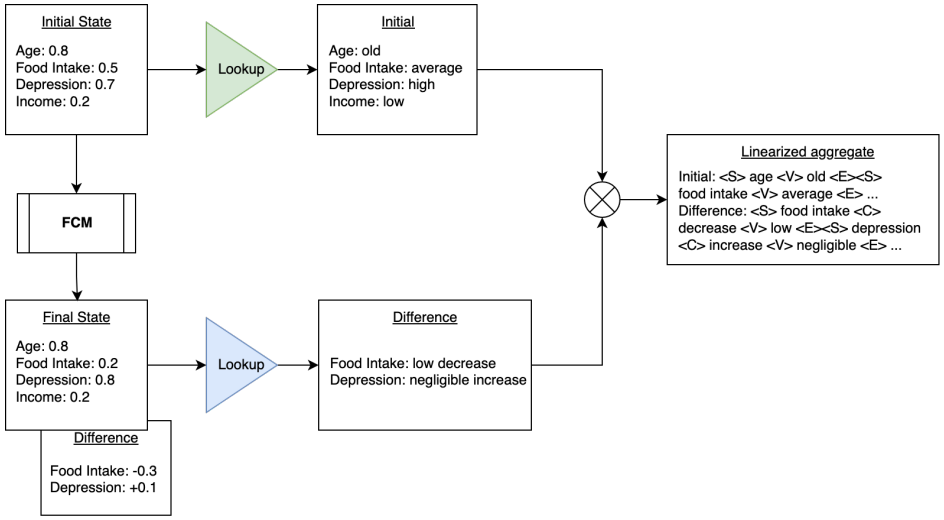

Figure 2: Encoding FCM simulation input and result into linearized aggregates

3.2 Data Augmentation

3.2.1 Data Augmentation - Permutation

3.2.2 Data Augmentation - Templates

In an effort to increase the amount of data, we created an algorithm to take in a dictionary of attribute keys with attribute values and produce a new initial and outcome sentence. Our intent is that the

sentences being generated will assist in creating more data that can be used to make up for requirement of bigger dataset that cannot be achieved from limitations of time and human annotations. Using templates is a common practice with NLG [8] and have only been replaced by neural network based methods. This inspired us to create a framework for creating templates and rules to generate sentences based on given input.

Initially, a set of simple templates are created by human annotators. Simple templates mean they lack complex conjunctions like double negations or conjunctive adverbs. These are deliberately ignored as templates become difficult to maintain and become less general. Furthermore, producing a diverse and flexible text with templates is impossible using templates [7]. Each of the templates contain variables, which can be replaced with appropriate quantified words when the template is evaluated and converted to text.

> "We considered a scenario of a {age} person with {depression_antidepressants} depression and antidepressants usage. Their food intake was {food-intake} ...",

Here you can see the variables (or attributes) that are needed are "age", "antidepressants", "depression", and "food-intake" are encased with curly brackets. We use an underscore between two variables as a separator and also they must have a similar values if they are put together with some tolerance. We set the tolerance to 0.1. Putting variables together removes a sense of robotic texts in the output, such that two templates may have different kinds of grouping. For example, in the example above, depression and antidepressants are grouped together, and hence their values are also expected to be similar. An input that would could use this template would be:

```
{"age":1, "depression":0.5, "antidepressants":0.4, "physical health":0.4, "food intake":0.4, ... }
```

Based on the available templates and input values, the algorithm chooses a valid template that exactly matches its requirements. We use dictionary to randomly pick a word based on the values to replace the template variable (figure 3).

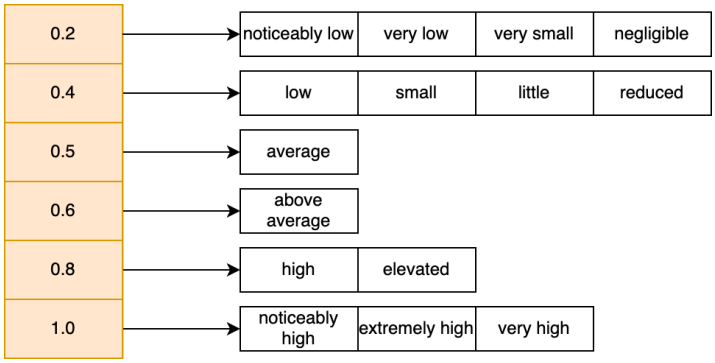

Figure 3: Dictionary to map numbers into qualitative words to use with template

The key in the dictionary is the maximum limit of the value associated with the possible words. The values for the attribute dictionary are the potential words that are randomly picked for replacement.

One thing that needs to be noted is that, some replacement may not ensure fluency while keeping meaning of the sentence the same. For such cases, more specific attributes like age are instead given a word from specific maps that are tailored to them. The age attribute dictionary would be the same structure but it would have more appropriate strings like youth, child, adult, senior, etc (where the values would be 0.2, 0.4, 0.6, 1.0). Also, there is a possibility for less coherent replacements like *low physical health* or elevated physical health instead of poor physical health or very good physical health respectively. After replacement of variables, we would receive texts such as,

> We considered a scenario of a old person with low depression and antidepressants usage. Their food intake was small ..."

For 100 input data, we generated 100 template-based texts.

4 Results

4.1 Model settings and training

We performed our experiments using OpenAI's GPT-3. The finetuning performed across its variants ada, babbage, curie, and davinci using unaugmented dataset produced better results with *curie* using automatic evaluations 2. This led us to continue our further experiments solely using curie because of its balanced benefit of ability to generate natural language and cost [1].

	BLEU-1	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**
Ada	0.50	0.23	0.12	0.07	0.42
Babbage	0.55	0.28	0.15	0.09	0.46
Curie	0.60	0.36	0.23	0.14	0.56
Davinci	0.55	0.32	0.22	0.14	0.53

Table 2: Scenarios created by varying the baseline scenario. *X5* signifies five repetitions for each of the characteristic scenario.

For FCM simulation, we manually crafted 80 scenarios that deviated from a given baseline scenario. A baseline scenario refers to the one that is considered to be a valid or agreed scenario that produces a result that FCM experts have agreed upon. Each deviated scenario had a particular kind of characteristic 3. We split our annotated dataset into training, validation, and test in the ratio of 80:10:10. The data augmentation 3.2 was performed only in the training set while validation and test datasets were reused for all different variations of augmentation 4. As GPT-3 is a model efficient in low shot learning [9], approximately 200 training samples were created for different kinds of augmentation 3.2

Characteristics
One concept changes consistently throughout five scenarios (x5)
Two concepts change consistently throughout five scenarios (x5)
Three concepts change consistently across five scenarios (x5)
Four random concepts change randomly across five scenarios (x5)

Table 3: Scenarios created by varying the baseline scenario. *X5* signifies five repetitions for each of the characteristic scenario.

We considered two stages of FCM simulations to explain it: the first iteration that represented the scenario for the FCM and last iteration that represented the stable outcome of the FCM. Once a combination of each scenario simulation was obtained and annotated manually we were able to build a training set. As OpenAI recommends at least a couple of hundreds of data input to finetune itself, we used data augmentation techniques like templates and permutation to inflate the dataset size.

We trained and evaluated our system using common automatic evaluation metrics like BLEU-1 through 4 and METEOR. We observed that BLEU-1 score and METEOR are close to 0.5 which indicates that auto-generated sentences are sufficiently similar to the reference sentences. However, it is also natural to observe that BLEU scores decrease as we consider higher n-gram scores 4.

4.2 Quality Analysis

Table 6 shows the explanations for different scenarios generated by our trained model and their expected representations. We observed that a large majority of concepts and their values reflected appropriately by the generated sentences. The table 6 shows concepts that are correct as unhighlighted, while incorrect ones are *red* in color. On the other hand, there are orange phrases in the training examples that are incorrectly annotated. For example, ... FIXME .

Furthermore, we came to learn that the order of concept-value pairs passed in as input did not really matter, as the training was done in the same fashion while permutations should have helped maintain the correct generation of sentences that reflect concept/value pairs without any regard to their order.

[1] https://beta.openai.com/docs/models/gpt-3

4.3 Ablation Study

We augmented our dataset to generalize our input scenarios such that order of concepts is not taken into account. This section discusses how each of the augmentation techniques assisted in improving the fine-tuned model.

4.3.1 Effect of permutations

The impact of augmenting the dataset with permuted representations is shown in table 4. The models performed better consistently when permutation augmentation across all variants of BLEU [10] and METEOR [11] measurements. For example, with the original dataset only (80 instances) BLEU-2 scored 0.36 while permutation augmentation (240 instances) raised the score to 0.61. The same behaviors can be seen across all other measures with over 0.2 points increase in the scores. Based on chosen measures we can see that the permutations help increase generalization of input linearized representations to help it learn the importance of concept-value pairs rather than their order, which supports the findings of Ampomah et. al [7].

We considered different permutation sizes into account before choosing 2 as the permutation size. As seen in the figure 4, we observe largest gain in the results for 2 permutations and afterwards the evaluate metrics scores have smaller gains. To note further, we begin to see text generations that are accurate to the expected sentences which means that finetuned model tends to overfit to the training set with higher permutations.

	Original	**Original + Template**	**Original + Permuted (2)**	**All combined**
BLEU-1	0.52	0.44	**0.76**	0.55
BLEU-2	0.36	0.20	**0.61**	0.34
BLEU-3	0.22	0.09	**0.54**	0.26
BLEU-4	0.14	0.04	**0.50**	0.21
METEOR	0.56	0.37	**0.75**	0.51

Table 4: Evaluation of NLG performance on original and augmented datasets. (+ Template) means we used custom-made templates to generate custom sentences. (+ Permuted) means the linearized representation (input to GPT-3) was permuted, where (2) refers to the number of permutations we used.

4.3.2 Effect of templates

We custom built 8 different kinds of templates that can possibly be used for different simulation scenarios, however not all of them apply for every possible scenario 3.2. Using the templates, the original dataset was augmented with 100 more instances by randomly applying a valid template for a given scenario. Unlike, permutation augmentation, template permutations reduced the evaluation scores across all metrics by at least 0.1. This is further bolstered by the result of *All combined* (which consists of Original, Permuted (2), and Templates), which shows all the scores falling compared to Original + Permuted (2) only.

5 Discussion and Conclusion

References

[1] Bart Kosko. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24(1):65–75, 1986.

[2] Pete Barbrook-Johnson and Alexandra S Penn. Fuzzy cognitive mapping. In *Systems Mapping*, pages 79–95. Springer, 2022.

[3] Anish Shrestha, Kyle Mielke, Tuong Anh Nguyen, and Philippe J. Giabbanelli. Automatically explaining a model: Using deep neural networks to generate text from causal maps. In *Proceedings of the 2022 Winter Simulation Conference*, 2022.

[4] Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*, 2021.

[5] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002.

[6] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[7] Isaac Ampomah, James Burton, Amir Enshaei, and Noura Al Moubayed. Generating textual explanations for machine learning models performance: A table-to-text task. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3542–3551, 2022.

[8] Martin Strauss and Michael Kipp. Eric: a generic rule-based framework for an affective embodied commentary agent. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 97–104, 2008.

[9] Bharath Chintagunta, Namit Katariya, Xavier Amatriain, and Anitha Kannan. Medically aware gpt-3 as a data generator for medical dialogue summarization. In *Machine Learning for Healthcare Conference*, pages 354–372. PMLR, 2021.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[11] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Appendix

5.1 Sentence Generation Experimentation

Expected	**Generated**
Initial: <S> Age <V> senior <E><S> Antidepressants <C> negligible <V> negligible <E>. Difference: <S> Exercise <C> decrease <V> negligible <E>.	The given person is a senior taking antidepressants. The model predicts that exercise has a negligible decrease in antidepressant usage.
Initial: <S> Age <V> senior <E><S> Antidepressants <C> negligible <V> negligible <E>. Difference: <S> Belief in personal responsibility <C> decrease <V> low <E>.	The current email is a senior taking antidepressants. The model predicts that a decrease in belief in personal responsibility will have a low impact on antidepressant usage.

Table 5: Faithful sentences with good coverage generated by GPT-3

5.2 Permutations

Expected	Generated
Initial: <S> Age <V> senior <E><S> Antidepressants <C> increase <V> risk <E> Difference: <S> Belief in personal responsibility <C> increase <V> risk <E>	We consider a scenario of a senior with low risk and In response to this scenario, the model predicts that the person's
Initial: <S> Age <V> senior <E><S> Antidepressants <C> increase <V> risk <E> Difference: <S> Antidepressants <C> increase <V> risk <E>	The individual in question is a senior citizen with a low level of The model predicts that the physical health will increase slightly

Table 6: Unfaithful sentences generated by GPT-3

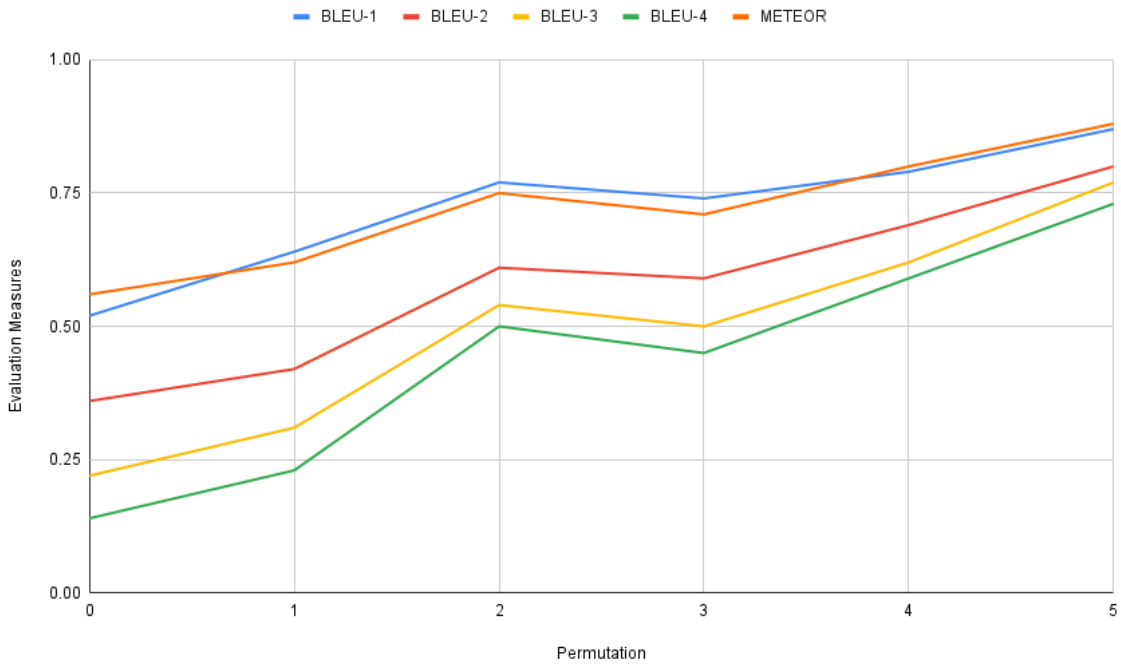

Figure 4: Variation of evaluation metrics according to permutation sizes. The increase in automatic scores also tends to overfit the finetuned model.