Machine Learning with Applications 11 (2023) 100452

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Text augmentation using a graph-based approach and clonal selection algorithm

Hadeer Ahmed [a,*], Issa Traore [a], Mohammad Mamun [b], Sherif Saad [c]

[a] ECE Department, University of Victoria, British Columbia, Canada
[b] National Research Council Canada, New Brunswick, Canada
[c] School of Computer Science, University of Windsor, Ontario, Canada

ARTICLE INFO

Keywords:
Data augmentation
Unstructured data
Cybersecurity
Text generation
Clonal selection

ABSTRACT

Annotated data is critical for machine learning models, but producing large amounts of data with high-quality labeling is a time-consuming and labor-intensive process. Natural language processing (NLP) and machine learning models have traditionally relied on the labels given by human annotators with varying degrees of competency, training, and experience. These kinds of labels are incredibly problematic because they are defined and enforced by arbitrary and ambiguous standards. In order to solve these issues of insufficient high-quality labels, researchers are now investigating automated methods for enhancing training and testing data sets. In this paper, we demonstrate how our proposed method improves the quality and quantity of data in two cybersecurity problems (fake news identification & sensitive data leak) by employing the clonal selection algorithm (CLONALG) and abstract meaning representation (AMR) graphs, and how it improves the performance of a classifier by at least 5% on two datasets.

1. Introduction

The recent advances in machine learning (ML) and deep learning (DL) gave rise to powerful prediction models with applications in many sectors. Machine learning and deep learning models, in particular, are data-driven algorithms that rely heavily on data to solve problems. Instead of using explicit rules to solve problems, ML and DL models learn to solve problems by analyzing data (aka "training data") in a process known as model training. Therefore, the availability of training data is crucial for designing and developing reliable systems that utilize machine learning and deep learning models. Data availability is a well-known problem in machine learning; the unavailability of training data is a common challenge facing data scientists when developing ML or DL models. Moreover, some sectors are known to struggle to collect enough training data. For example, in healthcare, finance, and cybersecurity, many technical and business constraints prevent organizations from obtaining enough training data to build reliable machine learning models. Additionally, the datasets available are quite often unreliable. Not only are these datasets used for training, but they are also used to test and evaluate models to assure their functionality and establish their overall efficacy. Hence, the quality of the data has a significant effect on the quality of the models. It is possible to train machine learning models using unreliable data. But doing so could lead to predictions that do not match reality because of biases or mistakes. In some instances, even when the data originates from a reliable source, it is unsafe to depend on data that has not been sufficiently verified.

In general, machine learning models that have been verified or trained against well-known benchmarks are considered cutting-edge for the problem they were created to solve. But because there is not enough data, these benchmarks are frequently used to train various machine learning models that address different, non-identical tasks. When this occurs, the datasets utilized are unlikely to be an accurate reflection of the data that some of those models would be applied to in the real world, leading to incorrect predictions by the models (Joshi, 2021).

There are also problems with incorrect labels, which are the annotations that many models utilize to identify correlations in data. A significant number of these labels are assigned by human operators in order to identify the data class. In reality, labeling data manually by such individuals based on their own biases and opinions has an impact on the overall quality of the labeling (Wiggers, 2022). For example, it was discovered that sites such as Amazon Mechanical Turk, where many academics recruit people to classify data, provide poor quality data. Workers are often driven by pay rather than interest and have a tendency to rush the process, especially when treated poorly and paid below-market rates (Dreyfuss, 2018).

To address these issues, researchers have investigated a variety of solutions, including the creation and use of simulated data, the augmentation of data, the use of transfer learning, and a combination of these

* Corresponding author.
E-mail addresses: hsahmed@uvic.ca (H. Ahmed), itraore@ece.uvic.ca (I. Traore), mohammad.mamun@nrc-cnrc.gc.ca (M. Mamun), shsaad@uwindsor.ca (S. Saad).

https://doi.org/10.1016/j.mlwa.2023.100452
Received 14 December 2022; Received in revised form 21 January 2023; Accepted 21 January 2023
Available online 23 January 2023

techniques (Roque, 2021). Simulated datasets are created artificially and formed from datasets taken from the actual world, which allows us to reproduce the qualities of real world data. This is beneficial because it enables machine learning models to retain their efficacy without substantial modifications, and the results obtained by real and simulated data are comparable (Matthew, 2018). Further, the use of transfer learning, as the name indicates, is the approach of employing existing, pre-trained AI models and the information accumulated to solve one problem and applying it to a new but similar problem. Transfer learning eliminates the requirement for substantial data labeling and filtering for machine learning. However, if the issues we are trying to address are not relevant or new, it is challenging to employ this method.

Data augmentation is another promising solution for the data unavailability problem. The term "data augmentation" refers to different methods to increase training data by generating new data from existing data. Recently, data augmentation techniques have become an active research topic, mainly because of the widespread adoption of deep learning models that require huge amounts of training data. Most of the literature focuses on tabular and image data augmentation techniques. However, recently, data augmentation techniques for textual data and NLP applications have become necessary. However, the task of augmenting textual data has proven to be challenging. The text augmentation could quickly generate text that does not preserve the original text's context, semantics, and grammatical structures. Moreover, it is challenging to guarantee that the generated text will maintain domain-specific keywords for specific problems or domains. Finally, text augmentation techniques are unsuitable for problems or applications that involve model drift. Most methods using language models for text generation rely on language models that assume the language is stable (does not change), so model drift is not considered.

In this paper, we tackle some key limitations of text augmentation by introducing a new method for improving the quality and volume of data using the clonal selection algorithm (CLONAG) and abstract meaning representation (AMR) graphs. The use of an AMR graph to keep the structure of the initial text enables generating different augmented texts while retaining control over the mutations made to the text. This allows us writing rules that preserve domain-specific keywords and preserve the texts' domain. In addition, by using the fitness function of the CLONAG to discard unsuitable samples generated during each iteration of the text mutation, we can generate a larger number of augmented texts from small input samples while preserving the label of the generated text and preventing any drift from occurring in the text. In addition, our method enables traceability and interpretability, as we are able to identify the impacts of each mutation iteration on the generated text and track their progression.

Our data augmentation model was evaluated using two data sets relating to two well-known cybersecurity issues. One is identifying fake news. It is an issue that has intensified over the past decade, negatively affecting the decisions and perspectives of some users (Lee, 2022). It is not a new problem, but it is difficult to spot due to the tendency of humans to trust deceptive information and the lack of control over its growth (Ahmed, Traore, & Saad, 2017). In the case of employing ML to detect fake news, it is challenging to obtain high-quality, well-labeled data that is void of issues (Asr & Taboada, 2019). The second challenge is sensitive data leakage. Experts in enterprise security continue to have difficulty in identifying unstructured sensitive data leaks. Most data leak prevention methods can recognize well-formatted patterns in sensitive structured data using rule-based or signature-based techniques. However, detecting sensitive unstructured data using rule-based methods is challenging. Machine learning algorithms are preferred for handling sensitive unstructured data, but they lack training data because of the difficulty of legally obtaining leaked sensitive data samples and other data quality challenges.

The rest of this paper is arranged in the following manner. Section 2 summarizes previous research on data augmentation in natural language processing. Section 3 presents our proposed method for data generation using AMR graphs and CLONALG algorithm. Section 4 presents and discusses experimental results on data we collected. Section 5 summarizes our work and discusses our future plan.

2. Literature review

This section will discuss and outline relevant research on data augmentation in natural language processing from the available literature.

Data augmentation refers to techniques for expanding the variety of training data without acquiring more data. The majority of solutions either supplement existing data with modest modifications or produce synthetic data, with the goal of the augmented data acting as a regularizer and reducing overfitting when training machine learning models (Peng, 2021). Adding extra training data and exposing the model to several versions of the same data class makes the training process more robust since it is more likely to generalize to the test set. This is often utilized in computer vision, where cropping, flipping, and color jittering, in addition to adding noise to the picture and randomly interpolating two images, are standard model training methods.

When it comes to data augmentation in natural language processing, we can divide the methods into two categories: rule-driven approaches and model-driven approaches (Feng et al., 2021). Currently, rule-driven techniques are the most widely used form of data augmentation strategy, whereas model-driven approaches are still being investigated and developed. Although rule-based solutions are easy to implement, the performance gains they produce are usually insignificant. A rule-based strategy is one in which rules are defined in order to generate better samples of data. This involves using "if-else" clauses and regex rules to enhance current data and insert, delete, and reorganize existing data using pre-defined scripts. In Wei and Zou (2019), the authors proposed easy data augmentation (EDA): a set of token-level random alteration techniques that include synonym replacement, random insertion, random swap, and random deletion.

Regular expression filtering is frequently used for cleaning data collected from the Internet and other data sources. However, it is also used in augmentation to uncover common languages and construct rules that correlate to graph-structured grammars using matching text patterns to expand them (Spasic & Nenadic, 2020). In Jin, Jin, Zhou, and Szolovits (2020), researchers proposed TextFooler, which calculates word significance scores by examining the output after each word is deleted, and then, picks the terms that had the most significant impact on the results for synonym replacement. The researchers in Wang and Yang (2015) proposed applying k-nearest-neighbor (KNN) and cosine similarity to locate a related term for replacement in text. Alternatively, it is common to employ pre-trained traditional word embeddings such as word2vec, GloVe, and FastText to perform the same function. In Atliha and Šešok (2020), the authors constructed a signed graph over the data for paraphrase identification, with individual sentences as nodes and label pairs (positive and negative) as "signed" edges. They infer augmented sentence from the graph using rules that use balancing theory and transitivity.

Model-based solutions have the potential to greatly improve performance, but developing and implementing them is relatively complex. In Feng, Li, and Hoey (2019), the authors proposed semantic text exchange (STE), which modifies the whole semantics of a text to match the context of a newly introduced word/phrase; it does this task using a pre-trained language model. Ding et al. (2020) proposed GDAUG, a tool that builds synthetic data using pre-trained transformer language models. In Nie, Tian, Wan, Song, and Dai (2020), a pre-trained embedding space was used to enrich word representations with context-sensitive attention-based mixing of their semantic neighbors. Anaby-Tavor et al. (2019) fined-tuned GPT-2 by utilizing training data that was used to produce candidate examples for each of the available classes. After training a classifier on the initial training set, the top k-candidate samples that confidently belong to the relevant class are selected for augmentation. Other researchers, such as (Peters et al., 2018; Radford, Narasimhan, Salimans, Sutskever, et al., 2018; Radford et al., 2019), chose to utilize data noise strategies, such as modifying words in the input of self-encoder networks in order to create a new phrase, or adding noise at the word-embedding level. Although a feasible approach when no access to a formal synonym model exists, this method

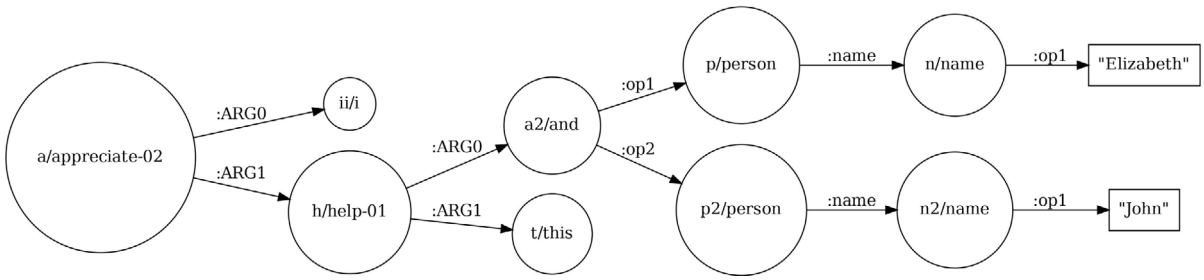

Fig. 1. AMR graph for the sentence, "I appreciate Elizabeth and John's help with this".

requires considerable training data. In addition, the employment of graphs as a potential strategy was also studied in recent research. The authors in Zhao et al. (2020) proposed a graph edge augmentation method that exposes Graph Neural Networks (GNNs) to probable edges while reducing exposure to improbable edges. The proposed approach achieved an average accuracy improvement of 5% across six popular data-sets.

Another common approach is round-trip translation based models, which involve translating the input data into a second language and then back into its original language. It is promising because of its great label preservation and highly valuable paraphrase capabilities. But when there is not much data to work with, the ability of these round-trip translation models is limited because they cannot generate a diverse sample of data from limited resources. In Gao et al. (2019) and Xie, Dai, Hovy, Luong, and Le (0000), the authors have devised strategies that use translation models to generate artificial training data. When training classification models with the additional data, both Gao et al. (2019) and Xie et al. (0000) achieved improved performance results including increases of +0.9 in accuracy and +1.9 in F1 score, respectively.

Existing models, as discussed above, only minimally modify texts with random behavior, such as by replacing synonyms or introducing spelling errors. As a result, they are less effective at producing sophisticated new data. Typically, the created data is nearly identical to the previous samples, and they rarely improve classifier performance. On the other hand, using word2vec, GloVe, and other word-embedding replacement methods, the context of the data is not kept intact, and they tend to add grammatical errors to the text. The big concern with pre-trained language models is the lack of control over the generated text. That is, we can generate texts on a given topic, but we have no control over the content of the texts. In addition, they require either to be fine-tuned or to be trained using an adequate amount of data to function successfully on domain-specific tasks. Hence, they are not ideally suited for solving the "big wall problem" phenomenon (Shorten & Khoshgoftaar, 2019), which refers to the difficulty for smaller companies, research groups, and businesses to get the same volume of data as large corporations. Furthermore, many such models do not offer any traceability, and it is hard to interpret their performance results.

GPT-2 and pre-trained language models tend to outperform all the other models mentioned in this study. However, their results vary across datasets; for example, the Anaby-Tavor et al. (2019) method improves accuracy by 10% on one dataset but only by 1.5% on another. To our knowledge, no one has explained why improvements differ across datasets or why a text augmentation approach performs better on some datasets than others. Thus, the introduction of graph-structured representations for text data has significant potential for enriching text data, as graphs are created from relationships and entities that reflect deeper knowledge, such as grammatical structures, linguistic data etc. Our proposed method employs AMR graphs that maintain the grammatical and semantic structure of text. When paired with the CLONALG algorithm, it enables us to control the change and the quality of the produced text while maintaining the statistical naturalness of the original text.

3. Methodology

This section will describe our proposed method for text augmentation, which involves three stages: the first stage consists of generating AMR graphs from the text; the second stage involves altering the AMR graphs using the CLONALG algorithm; and the third stage involves converting the altered AMR graphs to text using a pre-trained language model.

3.1. AMR parsing and text generation

A brief introduction to Abstract Meaning Representation (AMR) is necessary to comprehend the proposed approach. Natural Language Understanding (NLU), or the ability to represent meaning in natural language, is important for dealing with huge volumes of data in the real world. The goal of gaining perfect semantic comprehension of sentences in real-world data has had the interest of scholars for many years. Semantic representation, in particular, is an advantageous method that enables a computer to comprehend the meaning of a phrase by mapping the natural language sentence to some semantic representation in the form of a graph. AMR was initially introduced in 2013 by Banarescu and colleagues (Banarescu et al., 2013).

AMR representation allows representing sentences as root directed acyclic graphs with labeled edges and nodes. The nodes represent concepts, whereas the edges indicate whether or not there is a relationship between the concepts. Understanding these concepts and their relationships is critical for understanding the meaning of a sentence and for employing natural language processing technologies in applications such as data extraction, question answering, and machine learning. Furthermore, AMR graphs show reentrancy, which sets them apart from trees in that the same concept/node can engage in several relations simultaneously. Additionally, because an AMR graph is abstract, it may represent several sentences concurrently, which enables us to construct many sentences from a single AMR graph by altering its structure and nodes (Foland & Martin, 2017; Kandru, 2021; Wang, 2018).

Fig. 1 shows an example of AMR annotation from real-world data. The majority of the nodes are identifiable by their variables, such as "a" labeled with the notion "appreciate". Relations (e.g., ARG0) are the labeled edges linking the nodes. Constants are nodes in the network that do not have variables, such as "John", and they often represent a name, number, or time. The majority of AMR concepts may be correlated with a single word in a sentence, forming a one-to-one mapping.

We used the T5 model proposed by Raffel et al. (2019), to convert text to AMR and AMR to text. The T5 model provides a framework for representing a broad range of natural language processing problems as text-to-text problems for sentences-to-graph parsing and graph-to-sentence generation. To convert a text to a linearized AMR graph, we employed a trained T5-Large model, as Mishra and colleagues demonstrated (Mishra, 2021; Xu, Li, Zhu, Zhang, & Zhou, 2020). Both the parser and the generation model are trained on the AMR-3 (LDC2020T02) dataset (Lee, 2022).

Table 1
Examples of output text resulting from various mutation procedures.

Mutation operation	Input text	Output text
Focus replacement	Vaccine hesitancy is relevant amp it is important to differentiate it from antivaxx vaccine denial these are not the same thing but we also need to stop calling it vaccine hesitancy when a major underlying cause is equitable access to healthcare #covidvaccine #covid19ab	Access to healthcare is also important because of the hesitancy to vaccines (CovidVaccine COVID19AB) and the importance of distinguishing between hesitancy and denial of antivaxx vaccines (this is not the same, we need to stop calling it hesitancy antivaxx)
Concept replacement	Astonishing to know there is no data to demonstrate there is protection from the 1st dose of Pfizer after 21 days Again further negligence from our government in protecting peoples lives.	It is weird to know that there is no data showing that the first dosage of pfizer is good for protection again after 21 days.
Relationship replacement	It is weird to know that there is no data showing that the first dosage of pfizer is good for protection again after 21 days	It is weird to know that no data has shown that the first dosage of pfizer is good for protection again after 21 days.
Constant replacement	Astonishing to know There is NO data to demonstrate there is protection from the 1st dose of Pfizer after 21 days Again further negligence from our government in protecting peoples lives	Astonishing to know There is NO data to demonstrate there is protection from the 1st dose of moderna after 26 days Again further negligence from our government in protecting peoples lives.

3.2. Altering the AMR graphs

Following the initial stage of our model, we modified the existing AMR graphs using the clonal selection algorithm (CLONALG). CLONALG is an algorithm inspired by the acquired immunity hypothesis and based on the clonal selection principle (Brownlee, 2004). It focuses on well-known theories in computing algorithms that simulate processes in the biological immune system through selecting, cloning, and, mutation procedures (Jantan, Sa'dan, & Baskaran, 2016). In general, the CLONALG algorithm selects candidate solutions based on a match against a predefined pattern or an evaluation function. Candidates are cloned and mutated until the halting condition is reached (Brownlee, 2004). Our version of the algorithm works by cloning the native population for a certain amount of time, depending on pre-defined distance metrics. Following that, random alterations are introduced into the population to diversify it further. Thus, we can say the procedure may be divided into two stages: initialization and generation.

The initialization stage begins by establishing an initial pool of samples of size N and the preparation of the data. This is accomplished by randomly picking N samples from the data. We always divide our data into two categories: topic one and topic two, which correspond to a positive and negative class, respectively. When dealing with unlabeled data, clustering methods are used to group similar samples together if the data is jumbled and poorly labeled. Additionally, after clustering the data, we determine the shortest distance between each sample and the center of the cluster to which it belongs in order to determine the future number of clones. The clustering procedure is skipped if the data are properly labeled. In order to represent each class of data, we transfer them into AMR graphs; the AMR graphs that are created are collectively referred to as graph banks (aka gold graphs). Following that, the algorithm proceeds by executing several iterations to modify the sample data. A generation is defined as a single cycle of modification or iteration. We can configure the number of generations that the algorithm performs or specifies a condition that will stop the process if the algorithm encounters it. Each iteration includes cloning (copying) each AMR graph n times, where n is proportional to the average similarity between this graph and its parent graph in the prior population. The following equation is used to determine the number of clones:

$$number\ of\ clones = |(5 \times d_m - 1) \times 10| \quad (1)$$

where d_m is the distance between the current graph and its parent graph.

We constructed this equation in such a way that the number of generated clones is inversely proportional to the distance; therefore, when the distance is small, the number of generated clones is high, and vice versa. This is because we wanted the data to be as diverse as possible; if the generated AMR is similar to its parent, it will be cloned more frequently, which allows for more variation. Each clone should get one or more random modifications, such as word replacement, focus change, relationship alteration, etc. To be more precise, we developed four operations that will modify the AMR graphs: focus replacement, concept replacement, relationships replacement, and constant replacement.

The *focus replacement* operation modifies the AMR graph's focus; as previously explained, the graph is rooted; when the focus is modified, the root node is replaced with another node already present in the graph. This function will iterate over all variables and choose candidates for the new root/top variable. We eliminate nodes with incoming edges to drastically limit the candidate pool. Then, a new root is chosen at random among the candidates, and the graph is rearranged to accommodate the new root. The *concept replacement* operation replaces existing nodes with those that have a similar meaning. To do this, we employ two approaches to identify words that are comparable to the current concept: a wordnet (Miller, 1995) and gensim word2vec (Rehurek & Sojka, 2011). This method will iterate over all variables and choose a candidate at random. Then, a synonym reverting function is used to return a new word replacement and the word's initial letter is used as a potential variable to the old concept node. A function will check if the new variable is available, and if it is, it will replace the new concept with the old one. If the new variable is unavailable, the function will alter the variable and try an alternative replacement. The *relationship replacement* operation replaces an existing relationship with another; for instance, it will randomly replace the relationship "ARG0" with "ARG1" and so on. It achieves this by retrieving the graph's edges and altering their relationship tags randomly or according to predefined rules. Finally, the *constant replacement* procedure substitutes a predefined constant for an existing one. For instance, we may use this method to change the name "Jane" in all samples to "Jared". It achieves this by retrieving the constants in the graphs and altering their values based on constant type (e.g., name, country). We can see examples in Table 1 of how the various operations alter the text after modifying the input text in the AMR graph.

The mutation is executed by randomly modifying the graph using one or a combination of the operations we developed. The distance between each newly generated AMR and its parent is calculated using AMR similarity metrics (Opitz, Parcalabescu, & Frank, 2020); in this case, we utilize SMATCH (Song & Gildea, 2019). This metric works by extracting n-grams from two AMR graphs and comparing them. We determined whether to keep the newly produced AMR or to discard it based on a pre-defined threshold. If the generated AMR is retained, the distance is preserved and utilized to calculate the number of clones in the next iteration. This process will be repeated until we can calculate population growth (for example, the number of fit clones) and determine whether or not the stop requirements are fulfilled based on population growth changes. After the generating process was completed, we used a pre-trained T5 algorithm to convert the generated AMR graphs to text. Algorithm 1 shows the complete algorithm.

4. Experiment results and discussion

This section presents the experimental evaluation of the proposed model, and a discussion of the results obtained from the data we collected.

Algorithm 1: Data Augmentation Algorithm Based on CLONALG

```
Data: Input amr Graph
Result: Text
begin
    Cycle= 1;
    Cells = Initial population;
    while Cycle ⩽ MaxNumberIterations do
    for cell in Cells do
    switch mutationRate do
        case Concept do
            randomNode = getNode(triples.nodes);
            newNode = wordSynonym(random_node.text);
            newGraph = replaceNode(newNode, randomNode,
              triples );
            if getDistance (c,newGraph) ⩾ threshold then
                Cells += newGraph ;
        case Relationships do
            currentEdge = random (triples.edges);
            newGraph =replaceEdge(currentEdge,
              relationshipList, triples );
            if getDistance (c,newGraph) ⩾ threshold then
                Cells += newGraph ;
        case Constant do
            currentconstant = random (triples.constant);
            newGraph =replaceNode(currentconstant,
              constantList, triples );
            if getDistance (c,newGraph) ⩾ threshold then
                Cells += newGraph ;
        case focus do
            randomNode = getNode(triples.nodes);
            newGraph =rearrangeGraph(randomNode, triples);
            if getDistance (c,newGraph) ⩾ threshold then
                Cells += newGraph ;
    ;
    Cycle = Cycle+1 ;
```

4.1. Datasets and data generation

As previously described, we will assess the effectiveness of our proposed approach against two cybersecurity challenges: the problem of fake news and the difficulty of training machine learning models to recognize sensitive unstructured data. The datasets used to evaluate the model are discussed in the next section.

4.1.1. Sensitive dataset

To evaluate the model's effectiveness in detecting sensitive data, we chose to utilize COVID Twitter datasets from kaggle (Kash, 2019; Preda, 2020). In the case of sensitive information detection issues, we created a case study where a company is attempting to flag sensitive data in COVID tweets. They are primarily looking for COVID tweets that mention the manufacturing process, raw ingredients, or potential side effects of the COVID vaccination. Thus, we needed to develop two classes of tweets: one about the COVID vaccination (sensitive) and the other about COVID in a general sense (non-sensitive).

To prepare for the experiment, we chose and combined 2000 tweets from two Twitter data sets that discuss COVID and COVID vaccination based on pre-defined keywords. Also, due to the overlap in topics between these two datasets, we decided to apply a clustering method to further categorize the data, which was previously unlabeled and abundant, and separate it into two categories. In this case, the elbow clustering approach was employed to estimate the best number of clusters, which was two. Fig. 2 depicts the top 100 words in two output clusters identified using frequency word analysis. As can be seen, both

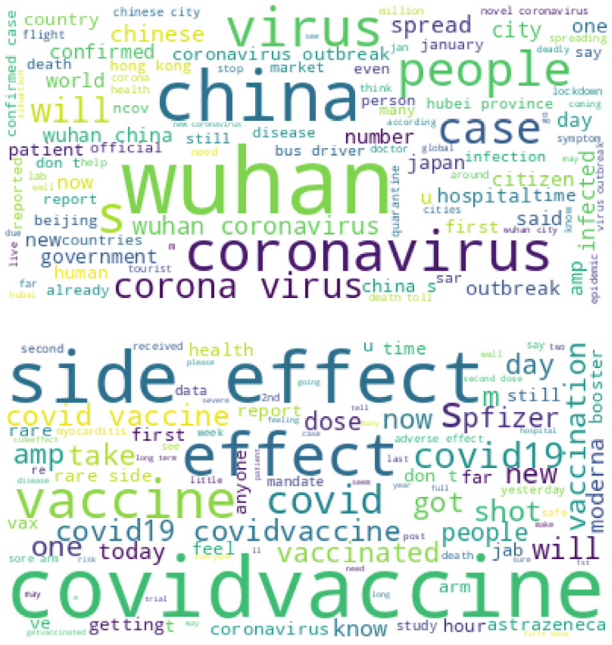

Fig. 2. Top 100 words in Cluster #0 (top) and Cluster #1 (bottom).

clusters correspond to our two classes, sensitive and non-sensitive. One cluster is comprised of tweets mentioning COVID in general, while the other is comprised of tweets discussing the effect of the COVID vaccination.

Additionally, we applied topic identification on the clusters to guarantee that the COVID datasets were accurately sorted into our two predefined classes (sensitive and non-sensitive). We can see the top three topics associated with Cluster 0 in Table 2, and the top three topics associated with Cluster 1 in Table 3. After confirming that the data is split into two classes, up to 100 tweets were randomly selected from each output cluster to serve as the initial population. Those 200 tweets were used to generate new samples of data, while the remaining tweets were used to validate the model.

4.1.2. Fake-News dataset

We used two datasets (Shu, Mahudeswaran, Wang, Lee, & Liu, 2018; Shu, Sliva, Wang, Tang and Liu, 2017; Shu, Wang and Liu, 2017) to evaluate the method's efficacy in improving the detection of fake news.

The first dataset consists of political news titles taken from "PolitiFact", which is a fact-checking website that rates the accuracy of statements made by political figures. The PolitiFact news dataset contains a collection of statements made by politicians, along with a rating indicating the accuracy of the statement. The ratings range from "true" to "false" and include several intermediate levels such as "mostly true" and "half true". The dataset also includes additional information such as the statement text, the speaker, and the date the statement was made. It is used to train machine learning models that can automatically fact-check statements made by politicians.

The second dataset is a collection of celebrity gossip news titles from "Gossip Cop", a website that evaluates the credibility of celebrity reporting (Shu, Mahudeswaran, Wang, Lee, & Liu, 2020). Gossip Cop is a website that fact-checks celebrity gossip and rumors. They report whether stories about celebrities are true or false and provide sources for their research. It is not clear if Gossip Cop provides a dataset with their fact-checked stories for public use or for research.

As explained previously, we simulated a situation where we had insufficient data. However, we skipped the clustering stage because both datasets are much smaller (contains 300 titles per class) and better organized than the COVID datasets. The data is divided into two

Table 2
Top 3 topics in cluster #0.

Topic	Keyword #1	Keyword #2	Keyword #3	Keyword #4
0	0.083*corona	0.059*virus	0.024*coronavirus	0.016*people
1	0.069*corona	0.065*virus	0.014*coronavirus	0.013*china
2	0.022*coronavirus	0.013*corona	0.012*china	0.011*outbreak

Table 3
Top 3 topics in cluster #1.

Topic	Keyword #1	Keyword #2	Keyword #3	Keyword #4
0	0.088*effect	0.083*covidvaccine	0.025*vaccine	0.017*covid19
1	0.051*effect	0.044*covidvaccine	0.021*vaccine	0.015*covid19
2	0.066*effect	0.060*covidvaccine	0.034*vaccine	0.025*covid19

Table 4
Size of the initial and augmented training sets and the testing sets used in the experiment.

Dataset	# Initial	# Augmented	# Testing
Covid tweets	200	800	400
PolitiFact news	100	400	200
Gossip Cop news	100	400	200

Table 5
Clustering prediction results.

Class	Accuracy
$Class_c$	80%
$Class_v$	94%

categories: fake titles and real titles. We selected 50 random samples to alter from each class. These 100 titles were used to generate new data samples, while the remaining ones were used for model validation.

4.2. Results and discussion

For the sensitive dataset, we converted each sample into an AMR graph and then ran our model separately on all 200 samples. We ran the algorithm until 800 samples were produced from sensitive and non-sensitive samples, respectively. The sizes of the testing sets and the training sets before and after augmentation[1] for all datasets are shown in Table 4 in detail. We evaluated the generated samples/text using two methods. The first determined whether or not the newly created text/samples clustered correctly. For example, if we generated text t_i from AMR graph A_i that belonged to $Cluster_0$ (non-sensitive), and t_i clustered to $Cluster_0$, then it was clustered successfully.

For the sake of convenience, we will refer to samples from Cluster 0 as $Class_c$ (non-sensitive) and samples from Cluster 1 as $Class_v$ (sensitive). As seen in Table 5, 80% of text generated by samples from $Class_c$ successfully cluster to $Cluster_0$, whereas 94% of text generated by samples from $Class_v$ cluster to $Cluster_1$.

The second assessment strategy was performed by examining whether the performance of a CNN text classifier improved when trained on newly generated data as opposed to when trained using the initial population only (200 samples); the test set was the remaining 1800 tweets set aside for evaluation. In addition, we selected three models from the literature review to allow us to compare our model's performance. As can be seen from Table 6, after using the newly generated data for training, the overall performance has improved significantly, and our model outperformed the other models.

For the fake news datasets, we transformed the 100 titles from each dataset (PolitiFact and Gossip Cop) into an AMR graph, and then we applied our augmentation method. We ran the algorithm until 400 samples/text were produced from fake and real samples, respectively. Similar to previously, we trained a CNN classifier using both the original population and newly generated text from each data set and compared its performance to three models from the literature review. As can be seen from Tables 7 and 8, the overall performance has greatly improved when using the new data and our model.

[1] https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/augmented-datasets/.

Table 6
Results of the CNN classifier on the COVID dataset.

Training data	Accuracy	Precision	Recall	F1-score
Original data	0.70	0.74	0.74	0.70
Our method	0.83	0.85	0.85	0.84
EDA	0.72	0.80	0.75	0.72
Back-translation	0.70	0.79	0.75	0.70
Contextual/Gpt2	0.80	0.84	0.83	0.81

Table 7
Results of the CNN classifier on the PolitiFact fake news dataset.

Training data	Accuracy	Precision	Recall	F1-score
Original data	0.56	0.71	0.57	0.48
Our method	0.72	0.72	0.72	0.72
EDA	0.63	0.70	0.70	0.70
Back-translation	0.67	0.68	0.68	0.67
Contextual/Gpt2	0.64	0.65	0.64	0.64

Table 8
Results of the CNN classifier on the Gossip Cop fake news dataset.

Training data	Accuracy	Precision	Recall	F1-score
Original data	0.58	0.60	0.58	0.57
Our method	0.70	0.73	0.70	0.71
EDA	0.60	0.65	0.60	0.63
Back-translation	0.62	0.65	0.68	0.67
Contextual/Gpt2	0.70	0.75	0.64	0.70

Our augmentation method was able to increase the performance of CNN text classifiers when trained on generated text from a small number of samples. Nevertheless, throughout our testing phases, we noticed that in order to achieve the best results, we had to limit the graph alterations. In the early rounds of testing, we sought to generate diverse samples/text from the original population; hence, we set the distance evaluation criterion between 0.4 and 0.9, which enabled the generation of more diverse data. However, because the created text topics were different from the original data, we disturbed the natural distribution of keywords in the data, resulting in a decrease in classifier performance. Several rounds of testing led us to believe that the best threshold for distance thresholds during the text mutation stage was between 0.6 and 0.8. We selected optimal thresholds by analyzing the BLEU scores (Song, Ning, Zhang, & Wu, 2021), a commonly used metric for evaluating the quality of machine-generated text. BLEU compares the machine-generated text to a set of reference texts and calculates a score based on the number of matching n-grams between the two. As shown in Fig. 3, we compared the thresholds to the BLEU score. A higher BLEU score indicates that the machine-generated text is more similar to the reference translations and is therefore considered to be of higher quality. Although thresholds above 0.9 had better performance, we eliminated them due to the desire to avoid generating identical text to the original samples. In addition we included a ROC curve shown in Fig. 4 that contains the performance of a CNN trained with data generated at various threshold levels as measured by the true positive rate and false positive rate. Furthermore, because each text sample in the original population underwent many rounds of mutation, it was possible to substitute words and concepts with ones that were much closer to the original concept/word but were inadequate in the original data. This had a detrimental effect on classifier performance and required that the mutation be confined to the targeted job.

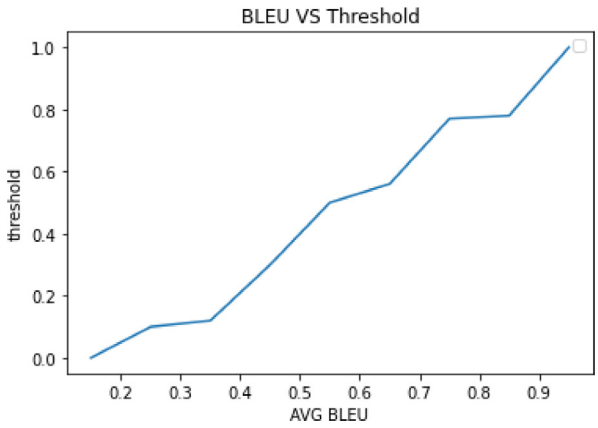

Fig. 3. Comparison of average BLEU score and thresholds.

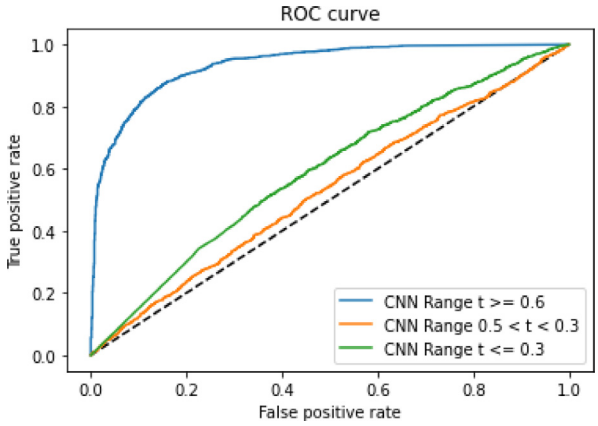

Fig. 4. ROC curve showing the performance of a CNN trained with data generated at various threshold levels.

5. Conclusion

We presented and assessed a new approach for textual data enhancement based on graphs and a clonal selection algorithm. We represented the data using AMR graphs and created a method for altering the graph's concepts and relationships in order to produce new, updated graphs. Using T5, a pre-trained language model, we transformed graphs to text and vice versa. Additionally, we demonstrated that the produced text considerably enhances the performance of machine learning classifiers. By training a CNN classifier with the new data, we observed an improvement in performance by more than 5%. We believe that by refining the proposed method, we may not only replace current concepts and relationships, but also add new ones to the AMR graphs. Although data augmentation is gaining popularity and appears to have tremendous potential, there are several downsides. For instance, many data augmentation techniques can only generate high-quality enhanced data if the underlying data set is sufficiently large. Using our strategy, however, we were able to enhance 100 or fewer data samples and still increase the performance of a machine learning classifier. Our future work will include incorporating branches (concept and relationship) into the AMR throughout the generation cycle and assessing the quality of the resulting data samples. Furthermore, we will explore introducing the concept and data drift to the augmented data. Previously, we stated that, after running an experiment with our model, we observed that the accuracy of predictive models reduces if the mutation is not constrained. We believe this is because we altered the natural distribution of the original data, causing either a data drift (Machiraju, 2021) or concept drift (Iwashita & Papa, 2019). Using the developed distance-based fitness function of the clonal selection method, we were able to prevent drift by restricting the mutation and eliminating new samples that might have caused drift if the mutation continued. Our next step is to expand our approaches so that we can exert greater control over the drift and not just prevent it, but even induce different sorts of targeted drift in text. The goal is to model different types of concepts and data drift of our data, such as sudden drift or gradual drift (Bayram, Ahmed, & Kassler, 2022) in text and NLP applications. In addition, we will determine if we can use this technique to improve and strengthen the performance of the prediction model by introducing previously unknown patterns.

CRediT authorship contribution statement

Hadeer Ahmed: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Issa Traore:** Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Mohammad Mamun:** Writing – original draft, Writing – review & editing. **Sherif Saad:** Conception and design of study, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hadeer Ahmed reports financial support was provided by National Research Council Canada.

Data availability

Data will be made available on request.

Acknowledgments

This project was supported in part by collaborative research funding from the National Research Council of Canada's Artificial Intelligence for Logistics Program. Approval of the version of the manuscript to be published.

References

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using N-gram analysis and machine learning techniques. *Lecture Notes in Computer Science*, 127–138. http://dx.doi.org/10.1007/978-3-319-69155-8_9, URL: https://link.springer.com/chapter/10.1007/978-3-319-69155-8_9.

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., et al. (2019). Not enough data? Deep learning to the rescue! URL: https://arxiv.org/abs/1911.03118.

Asr, F. T., & Taboada, M. (2019). Big data and quality data for fake news and misinformation detection. *Big Data & Society*, *6*(1), Article 2053951719843310. http://dx.doi.org/10.1177/2053951719843310, arXiv:https://doi.org/10.1177/2053951719843310.

Atliha, V., & Šešok, D. (2020). Text augmentation using BERT for image captioning. *Applied Sciences*, *10*(17), http://dx.doi.org/10.3390/app10175978, URL: https://www.mdpi.com/2076-3417/10/17/5978.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., et al. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse* (pp. 178–186). Sofia, Bulgaria: Association for Computational Linguistics, URL: https://aclanthology.org/W13-2322.

Bayram, F., Ahmed, B. S., & Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. http://dx.doi.org/10.48550/ARXIV.2203.11070, URL: https://arxiv.org/abs/2203.11070.

Brownlee, J. (2004). Clonal selection theory & clonalg. URL: https://researchbank.swinburne.edu.au/file/35a37376-d7b6-4cdf-b0a3-9b38abd10fa1/1/PDF%20%2843%20pages%29.pdf.

Ding, B., Liu, L., Bing, L., Kruengkrai, C., Nguyen, T. H., Joty, S., et al. (2020). DAGA: Data augmentation with a generation approach for low-resource tagging tasks. URL: https://arxiv.org/abs/2011.01549.

Dreyfuss, E. (2018). A bot panic hits amazon mechanical turk. *Wired*.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., et al. (2021). A survey of data augmentation approaches for NLP. URL: https://arxiv.org/abs/2105.03075.

Feng, S. Y., Li, A. W., & Hoey, J. (2019). Keep calm and switch on! Preserving sentiment and fluency in semantic text exchange. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. http://dx.doi.org/10.18653/v1/d19-1272, URL: https://arxiv.org/abs/1909.00088.

Foland, W., & Martin, J. H. (2017). Abstract meaning representation parsing using LSTM recurrent neural networks. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 463–472). Vancouver, Canada: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P17-1043, URL: https://aclanthology.org/P17-1043.

Gao, F., Zhu, J., Wu, L., Xia, Y., Qin, T., Cheng, X., et al. (2019). Soft contextual data augmentation for neural machine translation. (pp. 5539–5544). URL: https://aclanthology.org/P19-1555.pdf.

Iwashita, A. S., & Papa, J. P. (2019). An overview on concept drift learning. *IEEE Access*, *7*, 1532–1547. http://dx.doi.org/10.1109/access.2018.2886026, URL: https://ieeexplore.ieee.org/document/8571222.

Jantan, H., Sa'dan, S. A., & Baskaran, A. M. F. (2016). Artificial Immune Clonal Selection Based Algorithm in Academic Talent Selection. *Journal of Informatics and Mathematical Sciences*, *8*(4), 225–234. http://dx.doi.org/10.26713/jims.v8i4.554.

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8018–8025. http://dx.doi.org/10.1609/aaai.v34i05.6311, URL: https://ojs.aaai.org//index.php/AAAI/article/view/6311.

Joshi, N. (2021). 4 ways to tackle the lack of machine learning datasets. URL: https://www.allerin.com/blog/4-ways-to-tackle-the-lack-of-machine-learning-datasets.

Kandru, P. (2021). Guide to abstract meaning representation (AMR) to text with TensorFlow. https://analyticsindiamag.com/guide-to-abstract-meaning-representationamr-to-text-with-tensorflow/.

Kash (2019). Covid vaccine tweets. URL: https://www.kaggle.com/kaushiksuresh147/covidvaccine-tweets.

Lee, J. M. (2022). How fake news affects U.S. elections | university of central florida news. URL: https://www.ucf.edu/news/how-fake-news-affects-u-s-elections. [Online; accessed 19. Apr. 2022].

Machiraju, S. (2021). Why data drift detection is important and how do you automate it in 5 simple steps. URL: https://towardsdatascience.com/why-data-drift-detection-is-important-and-how-do-you-automate-it-in-5-simple-steps-96d611095d93.

Matthew, J. S. (2018). *Answering questions with data: introductory statistics for psychology students*. Open Textbook Library, URL: https://crumplab.com/statistics/simulating-data.html.

Miller, G. A. (1995). WordNet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41. http://dx.doi.org/10.1145/219717.219748.

Mishra, P. (2021). Understanding T5 model : Text to text transfer transformer model. URL: https://towardsdatascience.com/understanding-t5-model-text-to-text-transfer-transformer-model-69ce4c165023.

Nie, Y., Tian, Y., Wan, X., Song, Y., & Dai, B. (2020). Named entity recognition for social media texts with semantic augmentation. URL: https://arxiv.org/abs/2010.15458.

Opitz, J., Parcalabescu, L., & Frank, A. (2020). AMR similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, *8*, 522–538. http://dx.doi.org/10.1162/tacl_a_00329.

Peng, L. (2021). A short survey on implicit data augmentation - towards data science. URL: https://towardsdatascience.com/a-short-survey-on-implicit-data-augmentation-40b3eb1f7430.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. URL: https://arxiv.org/abs/1802.05365.

Preda, G. (2020). COVID19 tweets. URL: https://www.kaggle.com/gpreda/covid19-tweets.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. http://dx.doi.org/10.48550/arXiv.1910.10683, arXiv:1910.10683.

Rehurek, R., & Sojka, P. (2011). *Gensim–python framework for vector space modelling* (p. 3). Brno, Czech Republic: NLP Centre, Faculty of Informatics, Masaryk University.

Roque, L. (2021). Transfer learning and data augmentation applied to the simpsons image dataset. URL: https://towardsdatascience.com/transfer-learning-and-data-augmentation-applied-to-the-simpsons-image-dataset-e292716fbd43.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), http://dx.doi.org/10.1186/s40537-019-0197-0, URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018). FakeNewsNet: A data repository with news content, social context and dynamic information for studying fake news on social media. arXiv preprint arXiv:1809.01286.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, *8*(3), 171–188. http://dx.doi.org/10.1089/big.2020.0062, URL: https://github.com/KaiDMML/FakeNewsNet.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. http://dx.doi.org/10.48550/ARXIV.1708.01967, URL: https://arxiv.org/abs/1708.01967.

Shu, K., Wang, S., & Liu, H. (2017). Exploiting tri-relationship for fake news detection. arXiv preprint arXiv:1712.07709.

Song, L., & Gildea, D. (2019). SemBleu: A robust metric for AMR parsing evaluation. http://dx.doi.org/10.48550/arXiv.1905.10726, arXiv arXiv:1905.10726.

Song, C., Ning, N., Zhang, Y., & Wu, B. (2021). Knowledge augmented transformer for adversarial multidomain multiclassification multimodal fake news detection. *Neurocomputing*, *462*, 88–100. http://dx.doi.org/10.1016/j.neucom.2021.07.077, URL: https://www.sciencedirect.com/science/article/abs/pii/S0925231221011590.

Spasic, I., & Nenadic, G. (2020). Clinical text data in machine learning: Systematic review. *JMIR Medical Informatics*, http://dx.doi.org/10.2196/17984.

Wang, C. (2018). Abstract meaning representation parsing. https://www.cs.brandeis.edu/~cwang24/files/thesis-wang.pdf.

Wang, W. Y., & Yang, D. (2015). That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 2557–2563). Lisbon, Portugal: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D15-1306, URL: https://aclanthology.org/D15-1306.

Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv:1901.11196.

Wiggers, K. (2022). 3 big problems with datasets in AI and machine learning. *VentureBeat*, URL: https://venturebeat.com/2021/12/17/3-big-problems-with-datasets-in-ai-and-machine-learning.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., & Le, Q. Unsupervised data augmentation for consistency training. URL: https://arxiv.org/pdf/1904.12848.pdf.

Xu, D., Li, J., Zhu, M., Zhang, M., & Zhou, G. (2020). Improving AMR parsing with sequence-to-sequence pre-training. http://dx.doi.org/10.48550/arXiv.2010.01771, arXiv arXiv:2010.01771.

Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., & Shah, N. (2020). Data augmentation for graph neural networks. URL: https://arxiv.org/abs/2006.06830.