Towards Data Augmentation for DRS-to-Text Generation

Muhammad Saad Amin [1], Alessandro Mazzei [1] and Luca Anselma [1]

[1] *University of Turin, Corso Svizzera 185, Turin, 10149, Italy*

Abstract

The data augmentation approach is becoming very popular in Natural Language Generation (NLG). Different approaches have been utilized in NLP and NLG to augment data and increase training examples for the neural model. Yet no studies have performed augmentation on logical input i.e., Discourse Representation Structures (DRS). We present data augmentation in DRS i.e., DRS taken from the PMB corpus, for the DRS-to-Text generation task. We conducted our experiments on a standard bi-LSTM-based sequence-to-sequence model thus creating an end-to-end neural approach for generating English sentences from DRS. We evaluated the output generated from word-level and character-level decoders with the help of reference-based evaluation metrics like BLEU, ROUGE, METEOR, NIST, and CIDEr. The practical implementation of augmented DRS succeeded in achieving better results compared to DRS without augmentation. To prove the significance of our model, we conducted statistical significance tests i.e., the *Shapiro-Wilk Test* (to check data normality) and the *Wilcoxon Test* (to test model significance). *Wilcoxon* results states that our model is significantly better with the p-value = 2.37e-05 for Char-level model and p-value = 7.78e-07 for Word-level model.

Keywords

Bi-LSTM, Data Augmentation, DRS-to-Text Generation, Neural Network, Parallel Meaning Bank (PMB), Statistical Significance Test, Shapiro-Wilk Test, Wilcoxon Test

1. Introduction

Data augmentation is an approach utilized to increase the number of examples for training a neural model without explicitly adding new data examples [1]. This approach is becoming very trendy in many NLP and NLG applications nowadays. This is due to the complex nature of tasks being addressed. Previously, most of the researchers working in the Computer Vision (CV) domain use different augmentation techniques i.e., cropping, flipping, color jittering, rotating, etc. [2]. This CV augmentation approach is very applicable to increase the number of examples as rotated, flipped or cropped versions of an image are also an image. But augmentation approach for NLP and NLG is not so easy to implement due to the discrete nature of sentences [3]. That means, if our sentence augmentation is not good, it will result in ungrammatical sentences and thus result in the bad performance of the model.

Discourse Representation Structure *(DRS)* is derived from Discourse Representation Theory *(DRT)* that is the formal representation of data as first order logic. Initial works in formal meaning representation focused on the generation of DRS from text, an approach referred to as parsing [4]. This work was directed toward mapping of words with their relevant logical representation and formulation. But very few works have been implemented in translation i.e., generating sentences from Discourse Representation Structures (DRS). Recently, different authors have implemented a *bi-LSTM-based* neural sequence-to-sequence model to generate sentences from DRS [5]. But till now to our knowledge, no work has been done to augment DRS i.e., formal logical representation and translation of the logical representation. Keeping in mind this research gap, we worked on DRS augmentation to check whether this approach will help in improving model performance as increased metrics scores.

[1]NL4AI 2022: Sixth Workshop on Natural Language for Artificial Intelligence, November 30-11, 2022, Udine, Italy [33]
EMAIL: muhammadsaad.amin@unito.it (A. 1); alessandro.mazzei@unito.it (A. 2); luca.anselma@unito.it (A. 3)
ORCID: 0000-0002-7002-9373 (A. 1); 0000-0003-3072-0108 (A. 2); 0000-0003-2292-6480 (A. 3)

The research questions that we addressed in these experiments are listed as follows:

1. Is it possible to augment *Formal Meaning Representation* based on logical inputs i.e., DRS?
2. How augmentation can be performed in DRS and the translation of DRS as both belong to two different directions?
3. Does augmentation in DRS result in increased model performance?
4. How to statistically justify the results with the help of *Significance Tests*?

So, in a nutshell, we can say that our main contribution is twofold. First, we have developed a way of augmenting logical inputs (DRS) and their respective translations. The initial format of DRS is the *Box Format*, and this version of DRS cannot be embedded into the neural network directly. To make DRS an input for the neural network we must flatten the *Box format* of DRS into *Clausal format* and then *Clausal format* is preprocessed into *Absolute DRS format* to be fed into a Neural Network (NN). Getting corpus data from PMB, we performed an augmentation approach on the *Clausal format* of DRS so that it can be preprocessed and passed to the neural model. A graphical depiction of the *Box* and *Clausal* format of DRS along with the translation is shown in Figure 1 below.

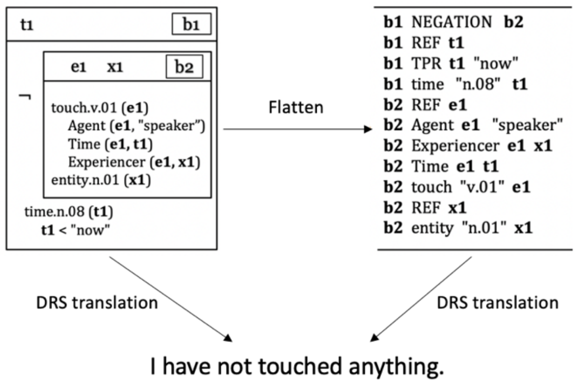

Figure 1: Box format of DRS (left-side) is flattened and converted into Clausal format of DRS (right-side) [5].

Both formats of DRS have the same meaning but to augment and embed DRS into NN, we must transform from *Box* format into *Clausal* format. So, we argued that the NN trained with augmented data produces better results. Secondly, we have applied statistical significance tests on the *DRS-to-Text* generation task to verify that better results are not achieved accidentally. For the implementation of statistical significance tests, the choice of the right test is another problem. Among a series of parametric and non-parametric tests, the choice of the right significance test is a tricky move. A detailed description of both contributions will be discussed in the latter sections.

The remaining paper is structured as follows: literature insights are described in Section 2. Section 3 describes the data and the approach used to augment logical input and respective translation of DRS. The methodology implemented to conduct the experiment is discussed in Section 4. Results are discussed in Section 5, and the conclusion and future work are described in Section 6.

2. Literature Insights

Literature insights into data augmentation in Natural Language Processing (NLP) and Generation (NLG) clearly state that this domain is still underexplored [6]. Many researchers in NLP have used different approaches to augment the data examples. Based on the text processing challenges, different *Rule-based* and *Model-based* approaches have been proposed by researchers in this domain [7]. Comparing the approaches, there exist some pros and cons of augmentation. *Rule-based* techniques are easily implementable but sometimes create more diverse data which is not required for data augmentation [8]. The data which is neither too similar nor too different from the original examples are considered good augmented data. Because similar or too different data moves towards overfitting of the model. Similarly, *model-based* approaches are considered good for augmentation, but it is very difficult to develop and utilize *model-based* augmentation approaches for increasing data every time [9].

Considering *Rule-based* techniques, different researchers proposed different approaches based on the nature of the task being executed. *Feature Space Data Augmentation* [10], *Easy Data Augmentation* based on *random insertion, deletion,* and *swap* [11], *Paraphrase Identification* [12], and *Dependency Tree Morphing* [13] are some of the *rule-based* approaches implemented in the literature. Similarly, *MixUp* (also referred as *Mixed Sample Data Augmentation Technique*, *MSDA*) [14], *CutMix* [15], *CutOut* [16], *Copy-Paste* [17], and *Seq2MixUp* [18] approaches are derived from *In-interpolation-based techniques*. Different *Model-based* techniques include *BackTranslation* [19], *SCPN* [20], *Semantic Text Exchange* (STE) [21], *ContextualAux* [22], *Lambada* [23], *XLDA* [24], *SeqMix* [25], *Slot-Sub-LM* [26], *UBT & TBT* [27], *Soft Con-textual DA* [28], *Data Diversification* [29], *DiPS* [30], and *Augmented SBERT* [31].

In our implementation, we have used a *Rule-based* approach to augment the data. We defined a *rule of verb change* with the help of *SpaCy NLP pipeline* to transform the data in present, past, and future tenses. Basically, in the *DRS-to-Text* generation system we have two formats as input to the Neural Network i.e., DRS and its respective translation as shown in fig. 1. Keeping in mind the aspect and nature of data used in our experimental implementation, we have to augment DRS and also the translation of the DRS. The nature of both types of data is totally different i.e., one is a logical input (*DRS*) and the other on is a linear text i.e., translation of DRS. By using a *Rule-based approach*, we successfully augment the DRS and the translation of DRS to increase the number of relevant examples, thus achieving higher results.

3. Data and Augmentation Approach

Originally, DRS is presented in Box format as it is easy to understand and analyze the structure. Box representation has unique labels i.e., b1, b2, b3… Each box has 2 layers stated as *top-layer* and the *bottom layer*. The *top layer* of DRS contains *Discourse Referents* i.e., x_1, t_1, and the *bottom layer* of DRS contains conditions over these *Discourse Referents*. Each referent or condition belongs to a unique box label. For example, b_2 person.n.01 x_1 contains three types of information i.e., b_2 as box label, x_1 as discourse referent, and person.n.01 as a predicate that is disambiguated with senses (senses are provided in *wordnet*, *synsets*) e.g., person.n.01, time.n.08.

The *box format* of DRS is not convenient for modeling purposes; therefore, we convert the Box format into the clausal format. The clausal format or the absolute format is easily readable by the neural network. In clausal format, the variables and the conditions of the box format are converted into clauses. For example, top box layer variables are converted into clauses by a special condition called "*REF*" i.e., b_2 REF x_1 which states that discourse variable x_1 is bound in box b_2.

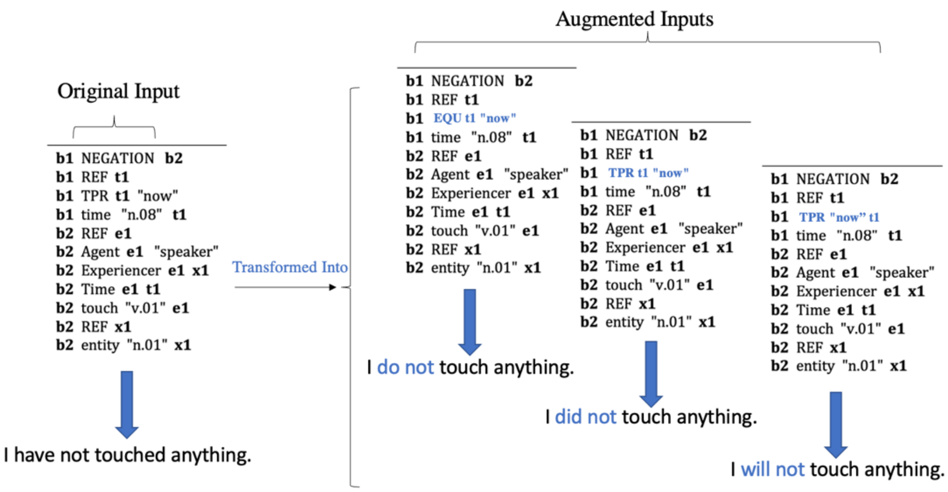

Figure 2: Graphical representation of data augmentation in DRS. On left there is original example of DRS with respective translation which is transformed into present, past, and future tense in both DRS and translation version.

DRS is also referred as the logical representation of components like semantic relations (*Agent, Patient, Theme*), operators (*REF, NOT*), the concepts (*touch.v.01*), variable indices (b_1, x_1), and deictic constants (*now, speaker, hearer*). By altering the values of these components, one can augment the DRS. There are multiple ways of augmenting a DRS based on *tense-change, polarity-change, name-change, quantity-change, and by changing numbers*. Among all these possible formats of DRS augmentation, we worked on *tense-change* approach. In *tense-change*, the tense of original *DRS* is converted into the present, past, and future tense as shown in Figure 2.

Tense-change augmentation is also referred to as a *verb-based* (*word that describes the action in the sentence*) augmentation approach because we are transforming verbs i.e., *present* → *past* and *future, past* → *present* and *future*, and *future* → *present* and *past*. By default, the tense change variants are taken as a *present, past, and future indefinite tenses.*

3.1. Left side: DRS Augmentation

DRS is a logical combination of events, and entities, and the relationships between these entities. Certain semantic phenomena are also covered in DRS including pronouns, presuppositions, quantification, negation, discourse relations, etc. Among different variants of DRS available on The *Parallel Meaning Bank (PMB) corpus*, we have used fully interpretable version of DRS. The reason behind this choice is the representation of information in DRS. In this version of DRS, we have *WordNet synset*-based *verbs, adverbs, nouns, and adjectives*. And *Verbnet* based semantic relations.

For augmenting DRS, we worked on a *verb-based* augmentation approach. To change the relation between entities of DRS, we adopted a simple *string-replacement* approach to replace one string with another string as shown in Fig.2. While iterating through each DRS, we first identified the time in which a verb is presented e.g., *EQU t1 "now"*, *TPR t1 "now", and TPR "now" t1*. These three formats represent verbs in any format of the present, past, or future tense. After, the identification of DRS in one format, we performed string replacement to convert a verb happening only in one type of tense into multiple types of different tenses e.g., *have not* → *does not, did not, will not* etc. This is how to augment the DRS which is the logical section of our input data. But during the augmentation of DRS, we kept track of the relevant translations of respective DRS as well. But just like DRS, augmentation of its translation is not just a string replacement approach. For the augmentation of linear text into different sentences, we used a *Rule-based* approach to convert sentences discussed in section 3.2 below.

3.2. Right side: Text Augmentation

Text augmentation as *tense change* is a very challenging task in NLP. For our implementation, we have used *SpaCy pipeline* to transform English sentences from one type of tense into another type based on the transformation performed in DRS. For implementation, we used *SQLite* database to keep track of the sentences with a max length of 1000 characters. We applied this pipeline to process the initial sentence and worked on sentence patterns to learn the structure of the sentence (*conjugates, singular, plural, past, present, and future*).

In tense transformation e.g., *tense change*, there are also other factors that must be kept in mind while reconstructing the sentence. Some major points of consideration include *active and passive, imperative, negation, singular and plural, subject and object, nouns, progressive and perfect, infinitive, first person, ambiguous, POS, and perfect participles sentences.* We have not worked only on simple and positive sentences but based on the translation of DRS, we have to deal with all types of tenses mentioned above. Table 1 elaborates on the examples associated with each type of tense form to identify the complexity of the task addressed.

If a sentence is presented as present perfect, present perfect continuous, or present continuous than it is converted into present indefinite as the default mode of tense change is the indefinite mode. The same strategy is also applied to other types of continuous, perfect and perfect continuous forms of past and future sentences.

Table 1
All cases of tense change encountered in our implementation

Conversion Type	Original Sentence	Converted Sentence
Present to Past & Future	I catch you	I caught you I will catch you
Past to Present & Future	He cheated on me	He cheats on me He will cheat on me
Future to Present & Past	I will love you	I love you I loved you
First person	I said no He said no	I say no He says no
Infinitive	I love to love	I will love to love
Ambiguous-POS	It was a thought	It will be a thought
Plural	The rabbits ran The rabbit ran	The rabbits run The rabbit runs
Third person singular	It will work	It works
Taking *will* as noun	The will says otherwise	The will said otherwise The will will say otherwise
Perfect tense	He had walked to the store	He walks to the store He will walk to the store
Continuous tense	I was going to the store	I am going to the store I will be going to the store
Double tense change	I win because I have five cookies	I won because I had five cookies
Negation	I did not go	I do not go I will not go
Future perfect	I will have been alive	I am alive I was alive I will be alive
Passive tenses	I am filled	I will be filled

4. Experimental Implementation

For the implementation of the experiment, a series of experimental steps are executed to perform the task under observation. For implementing augmentation in *DRS-to-Text* generation, we performed *Rule-based* and *string replacement* based on operations on DRS data. After performing data augmentation, we must put the augmented data into a *bi-LSTM-based neural network* to analyze the performance of our approach. For *Neural Machine Translation* (NMT) tasks, *LSTM* has been considered as the best model due to its ability to remember the connection between long-term input sequences [4]. Depending on literature-based suggestions, we also used *bi-LSTM-based sequence-to-sequence model* to translate DRS into English sentences.

DRS-to-Text is a particular logic to language generation task where input is the first-order logic and output is the corresponding linear text. This is not a generalized text generation task from graphs, tables, or images. Therefore, we must use a sequence-to-sequence model capable of remembering long sequences, and bi-LSTM is proven successful in remembering long logical input sequences [5]. Different pre-trained language models like BERT, ELMo, and ROBERTa have been used previously for parsing e.g., Text-to-AMR and Text-to-DRS. Still, for translation and generation, most of the researchers have focused only on bi-LSTM-based architectures [4]. Dealing with a very specific task, we have not tried other Transformer-based i.e., BERT, GPT, and BART architectures for logic-to-language implementation. But this can be a very interesting future direction to explore further architectures that can beat bi-LSTM for logic to language-based text generation task.

Neural Architecture. For the implementation of the experiment, we have used the encoder-decoder architecture of the NMT module. Bi-directional LSTM operates input sequences in both directions. The encoder part of the model encodes DRS representation, and the decoder module decodes DRS into its respective English sentences. To conduct this experiment, we have used *GPUs* with *CUDA* based *parallel computing platform* to speed up the experimental performance. The hyperparameter setting for our experiment is shown in Table 2 mentioning the parameters and their corresponding values.

Table 2
Hyperparameters of neural architecture for this experiment

Parameters	Values
Dimensions Embedding & RNN	300
Enc/Dec Cell	LSTM
Enc/Dec Depth	2
Mini-batch	48
Normalization Rate	0.9
lr-decay	0.5
lr-decay-strategy	Epoch
Optimizer	Adam
Validation Metric	Cross-Entropy
Cost-Type	ce-mean
Beam Size	10
Learning Rate	0.002

Dataset. We have used the English version of the *Parallel Meaning Bank (PMB) 3.0.0* dataset for our experiment, having gold standard (fully annotated corpus) 6620, 885, and 898 *training, validation*, and *testing* examples. Based on the nature of our implementation, we have used *Gold-PMB* dataset in both formats i.e., with augmentation and without augmentation, to check the increase in the evaluation scores. Then we expanded the training examples by adding *Silver-PMB* (partially manually annotated data) 97,598 training examples with *Gold-PMB* training examples. Collectively, to train our model without data augmentation, we have 104,218 *training*, 885 *validation*, and 898 *testing* examples. In the second experiment i.e., DRS-to-Text generation with augmentation, we only performed data augmentation on training examples. We did not augment, validation, or testing examples of the dataset. After train augmentation, we were having 26,480 training examples in the case of augmentation in *Gold-PMB*, and 4,16,872 training examples in the case of augmentation in *Gold-Silver-PMB*. Validation and testing files of PMB data are not augmented in our experiment. We also added only training examples of *Silver-PMB* with *Gold-PMB* to increase the number of training examples for our neural model. All dataset examples with and without augmentation are mentioned in Table 3 below.

Table 3
Dataset training, validation, and testing examples with and without data augmentation

Without Augmentation		With Augmentation	
Training (*Gold-PMB*)	6620	Training (*Gold-PMB*)	26480
Training (*Gold+Silver-PMB*)	104218	Training (*Gold+Silver-PMB*)	416872
Validation	885	Validation	885
Testing	898	Testing	898

Implementation Pipeline. The implementation pipeline includes all the steps involved in English text generation from DRS. Our main focus of this experiment is to perform data augmentation in DRS and analyze the accuracy improvement. So, we choose the *clausal format* of augmented DRS and preprocess it to make meaningful entities as atomic entities. This representation of DRS is meaningful for a neural network to understand the input pattern and perform well. The complete implementation pipeline is shown in Figure 3 below.

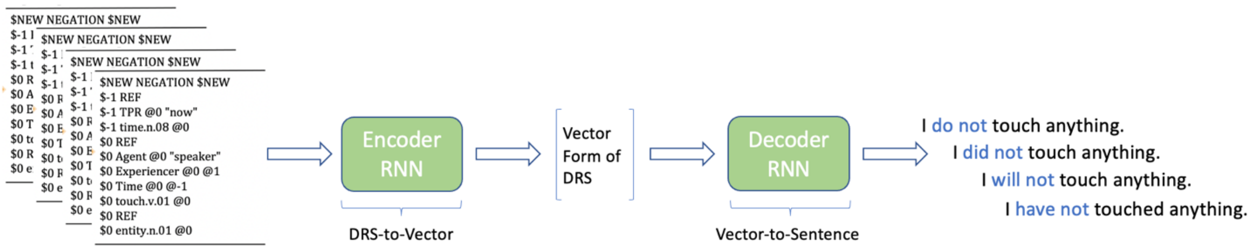

Figure 3: Complete pipeline of DRS to text generation. Encoder part encodes DRS to its respective vectorized form and then vectorized form is converted into English sentence with the help of decoder.

The encoder part of bi-LSTM encodes the DRS and converts it into vector form. This vector form is then embedded into the decoder part to be converted into respective English sentences. The neural model-generated English sentences are then compared with the reference English sentences to calculate the evaluation scores. For the evaluation of generated sentences, we are using 5 different automatic evaluation metrics like *BLEU, ROUGE, NIST, METEOR,* and *CIDEr* to check the syntax, semantics, relevance, and grammatical structure of the generated text. We have compared our results with *state-of-the-art DRS-to-Text* results of authors in [5] and proved that augmentation is helpful in getting better results as compared to results generated without augmentation.

5. Results

Results are the outcomes received after the implementation of the proposed methodology. Here we discuss our findings and try to prove the research questions addressed previously. In the implementation of DRS-to-Text generation, we conducted two experiments based on the types of PMB datasets. Our first experiment is also referred to as the baseline experiment conducted on the Gold-PMB dataset. We performed two different experiments on the gold dataset i.e., an experiment without augmentation on the PMB-Gold dataset, and an experiment with augmentation on the PMB-Gold dataset. We analyzed character-level and word-level results of the model and achieved high evaluation scores in all formats of evaluation metrics. Baseline results are mentioned in Table 4 with all descriptions of the dataset and evaluation metrics.

Table 4
Comparison of evaluation scores with and without augmentation

Dataset Type	Result Type	BLEU	NIST	METEOR	ROUGE_L	CIDEr
Gold-PMB (*Without augmentation*)	Char Level	47.72	7.68	39.42	72.59	4.84
	Word Level	32.91	5.80	29.99	61.39	3.49
Gold-PMB (*With augmentation*)	Char Level	**52.30**	**7.94**	**41.53**	**74.63**	**5.09**
	Word Level	**41.89**	**6.84**	**35.79**	**68.37**	**4.25**
Gold-Silver-PMB (*Wang et al.*)	Char Level	69.30	---	51.80	84.90	---
	Word Level	64.70	---	47.80	81.10	---
Gold-Silver-PMB (*Without augmentation*)	Char Level	70.18	9.44	52.20	85.74	6.85
	Word Level	64.11	8.93	47.59	81.31	6.11
Gold-Silver-PMB (*With augmentation*)	Char Level	**72.38**	**10.49**	**53.18**	**86.40**	**7.01**
	Word Level	**65.58**	**9.37**	**47.83**	**82.26**	**6.25**

Our second experiment is based on certain findings: first, if we add training examples of Silver-PMB data (not fully manually annotated corpus) with Gold-PMB data (fully annotated corpus), will it also go for an increase in evaluation scores? Secondly, can we achieve higher evaluation scores as compared to the Gold-PMB augmentation? Finally, we also must compare our augmentation-based results with literature models. So, to prove our hypothesis, we augmented the Gold and Silver PMB training examples and conducted the experiment. We succeeded in achieving high evaluation scores of all metrics but this time the score was not as high as we achieved in the Gold-PMB experiment. This is possibly due to the addition of certain DRS examples which were not fully manually annotated by the experts. A noise in SILVER-PMB data propagated through all the variants of dataset with and without augmentation. This causes into less increase in evaluation scores. Just like the augmentation results of Gold-PMB, we also analyzed character-level and word-level results of the neural model. We also compared the results with the literature and our implementation of the model with and without augmentation. All results are mentioned in Table 4.

The table reflects the successful implementation of our proposed hypothesis. In the literature, to the best of our knowledge, there is no implementation of augmentation in DRS but there are other implementations of DRS for language translations. To strengthen our hypothesis, we conducted a baseline experiment on a fully manually annotated gold corpus. Our baseline experiment strengthens our claim and then we further embedded Silver data into Gold and performed augmentation tasks. The first 2 experimental findings are of baseline experiments with and without augmentation. It is clearly shown in a bold format that we achieved efficient results for the augmented version of the DRS-to-Text implementation. The remaining 3 experiments are listed as the literature-based implementation of the author in 3rd row of Table 4. The 4th and 5th rows are our implementations on the gold and silver datasets with and without augmentation. And the 5th row (in bold) also highlights our augmentation-based results as the high scorer in its regard.

Statistical Significance Tests. To prove our model's achievement statistically, we conducted certain statistical significance tests as well [32]. Significance tests are becoming a new trend in the NLG domain nowadays. Significance tests are applied when two different models are applied to the same data, or the same model is applied to two different datasets. In our case, we applied the same bi-LSTM-based sequence-to-sequence model on two different data samples i.e., dataset without augmentation and dataset with augmentation. The purpose of doing these tests is to verify that the good results of one model are not achieved accidentally. Therefore, among a series of parametric and non-parametric tests, we choose the right test for our experiment based on two findings. First, we determined whether our data is normally distributed or not.

To check the normality of the data, we conducted Shapiro-Wilk Test. We choose this test because it is highly effective as compared to other tests used to check data normality. In our case, our data were not normally distributed and therefore we have to move towards non-parametric tests. If our data was normally distributed, then only a t-test would be enough to check model significance [32]. Among a list of non-parametric tests, we choose Wilcoxon Test due to two reasons. First, we choose the Wilcoxon test because it is highly suitable for the data which is coming from automatic evaluation metrics e.g., *BLEU, ROUGE, METEOR*, etc. Secondly, we choose this because it has the highest statistical significance as compared to other non-parametric tests working on scores coming from automatic evaluation metrics.

For the implementation of significance tests, we calculated the sentence-wise score of BLEU for model-generated test data and Gold reference data having approximately 1K examples. We conducted character level and word level significance tests and found that our augmentation models are significantly better with p-value = 2.37e-05 for the Char-level model and p-value = 7.78e-07 for the Word-level model.

6. Conclusion and Future Work

Data augmentation is a very challenging task in NLP and NLG. The main goal of augmentation is to increase training examples for the neural model without explicitly adding new data for training. In this contrast, we have implemented a data augmentation approach in DRS for text generation tasks. We conducted two experiments on PMB gold and gold-silver datasets. We achieved high evaluation scores of BLEU, ROUGE, METEOR, NIST, and CIDEr in the case of a model trained on augmented data. Furthermore, we conducted statistical significance tests to prove model performance on both character-level and word-level translations. We found that our augmentation models are significantly better with p-value = 2.37e-05 for Char-level model and p-value = 7.78e-07 for Word-level model.

In future, we will extend this experiment by applying other data augmentation approaches on logical forms (DRS) with respect to polarity change, number change, quantity change, and name change in the same DRS. We are also focusing on applying augmentation on low-resource languages like *ITALIAN, FRENCH, and DUTCH.*

7. References

[1] Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[2] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1):60.

[3] Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020b. Data boost: Text data augmentation through reinforcement learning guided conditional generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9031–9041, Online. Association for Computational Linguistics.

[4] Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018b. Exploring neural methods for parsing discourse representation structures. Transactions of the Association for Computational Linguistics, 6:619–633.

[5] Wang, C., van Noord, R., Bisazza, A., & Bos, J. (2021). Evaluating Text Generation from Discourse Representation Structures. In A. Bosselut, E. Durmus, V. Prashant Gangal, S. Gehrmann, Y. Jernite, L. Perez-Beltrachini, S. Shaikh, & W. Xu (Eds.), *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)* (pp. 73-83). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2021.gem-1.8.

[6] Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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), pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

[7] Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. *arXiv preprint arXiv:2105.03075*.

[8] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-Level Convolutional Networks for Text Classification. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

[9] Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019a. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3609–3619, Minneapolis, Minnesota. Association for Computational Linguistics.

[10] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex M Bronstein. 2018. δencoder: an effective sample synthesis method for few-shot object recognition. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages 2850–2860.

[11] Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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), pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

[12] Hannah Chen, Yangfeng Ji, and David Evans. 2020b. Finding friends and flipping frenemies: Automatic paraphrase dataset augmentation using graph theory. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4741–4751, Online. Association for Computational Linguistics.

[13] Gözde Gül ¸Sahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for lowresource languages. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.
[14] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. Proceedings of ICLR.
[15] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6023–6032.
[16] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. arXiv preprint.
[17] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. 2020. Simple copy-paste is a strong data augmentation method for instance segmentation. arXiv preprint.
[18] Demi Guo, Yoon Kim, and Alexander Rush. 2020. Sequence-level mixed sample data augmentation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5547–5552, Online. Association for Computational Linguistics.
[19] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96, Berlin, Germany. Association for Computational Linguistics.
[20] John Wieting and Kevin Gimpel. 2017. Revisiting Recurrent Networks for Paraphrastic Sentence Embeddings. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2078–2088, Vancouver, Canada. Association for Computational Linguistics.
[21] Steven Y. Feng, Aaron W. Li, and Jesse Hoey. 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), pages 2701–2711, Hong Kong, China. Association for Computational Linguistics.
[22] Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.
[23] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? Deep learning to the rescue! In Proceedings of AAAI, pages 7383–7390.
[24] Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. Xlda: Cross-lingual data augmentation for natural language inference and question answering. arXiv preprint arXiv:1905.11471.
[25] Rongzhi Zhang, Yue Yu, and Chao Zhang. 2020. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8566–8579, Online. Association for Computational Linguistics.
[26] Samuel Louvan and Bernardo Magnini. 2020. Simple is better! lightweight data augmentation for low resource slot filling and intent classification. In Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation, pages 167– 177, Hanoi, Vietnam. Association for Computational Linguistics.
[27] Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving Robustness of Machine Translation with Synthetic Noise. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.
[28] Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In Proceedings of the

57th Annual Meeting of the Association for Computational Linguistics, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.
[29] Xuan-Phi Nguyen, Shafiq Joty, Kui Wu, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. In Advances in Neural Information Processing Systems, volume 33, pages 10018–10029. Curran Associates, Inc.
[30] Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019a. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3609–3619, Minneapolis, Minnesota. Association for Computational Linguistics.
[31] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. Proceedings of NAACL.
[32] Dror, R., Baumer, G., Shlomov, S., & Reichart, R. (2018, July). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1383-1392).
[33] D. Nozza, L. Passaro, M. Polignano, Preface to the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), November 30, 2022, CEUR-WS.org, 2022.