Diverse Paraphrasing with Insertion Models for Few-Shot Intent Detection

Raphaël Chevasson, Charlotte Laclau, Christophe Gravier

Diverse Paraphrasing with Insertion Models for Few-Shot Intent Detection

Raphaël Chevasson[1], Charlotte Laclau[2], and Christophe Gravier[1]

[1] Université Jean Monnet Saint-Étienne, CNRS, Institut d Optique Graduate School
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ÉTIENNE, FRANCE
firstname.lastname@univ-st-etienne.fr
[2] Télécom Paris, Institut Polytechnique de Paris
charlotte.laclau@telecom-paris.fr

Abstract. In contrast to classic autoregressive generation, insertion-based models can predict in a order-free way multiple tokens at a time, which make their generation uniquely controllable: it can be constrained to strictly include an ordered list of tokens. We propose to exploit this feature in a new diverse paraphrasing framework: first, we extract important tokens or keywords in the source sentence; second, we augment them; third, we generate new samples around them by using insertion models. We show that the generated paraphrases are competitive with state of the art autoregressive paraphrasers, not only in diversity but also in quality. We further investigate their potential to create new pseudo-labelled samples for data augmentation, using a meta-learning classification framework, and find equally competitive result. In addition to proving non-autoregressive (NAR) viability for paraphrasing, we contribute our open-source framework as a starting point for further research into controllable NAR generation.

Keywords: Deep Learning · Natural language processing · Controllable text generation · Transformers · Non-autoregressive · Insertion models.

1 Introduction

A *good* paraphraser should, for each source sentence, generate a batch of paraphrases which 1. are fluent, 2. have a similar meaning with the original source and 3. are sufficiently diverse between themselves and also between the source sentence. Since a classic language model only optimize for fluency, the two last requirements are harder to satisfy and require a special loss, architecture, or decoding scheme. For automatic text generation, predicting the next token autoregressively (left-to-right, one at a time) using transformers neural networks is the most popular approach. Other emerging methods, such as insertion-based models, can predict in a order-free way multiple tokens at a time, attracting a lot of attention recently due to the potential gain to inference time. However, an understudied benefit of these models is their ability to constrain the generation to strictly include an ordered list of tokens, since they can build a sentence around them.

In this work, we investigate ways to leverage the generation constraints allowed by insertion-based text generation for diverse and slot-retaining generation for paraphrasing. We investigate the following scientific questions:
RQ1: Can insertion models be used as an efficient trade-off between fluency, semantic similarity and diversity in neural paraphrasing?
RQ2: What is the potential of such paraphraser as a data augmentation technique with respect to AR paraphrasers? Can this comparison inform us on the relative importance of fluency, similarity, and diversity for data augmentation using neural paraphrasing?

2 Related Work

As a text-to-text task, paraphrasing shares much similarity with translation and summarization. The most common approach is to use pretrained text-to-text models like BART [13] or T5 [21], fine-tuned on paraphrase corpus like MSRP [31], PAWS [28,30] or Quora [22][3]. A variation is to use off the shelf translation models to translate into many different languages (for diversity), then back-translating into the source language[4] [8,14,26], also known as round-trip translation (`RTT`). While `RTT` generates highly fluent sentences, it lacks diversity and does not guarantee that the meaning of the original sentence is preserved [4].

Other works opt to guaranty diversity, such as DivGAN [2] which forces the diverse sampling of a GAN latent via a diversity loss term. [20] also uses a GAN framework, but with several generators, and a compound loss with two discriminators ensuring paraphrases are distinguishable between themselves yet valid with respect to the source. ProtAugment [4] uses a variant of beam search with a diversity term [24] and randomly forbids unigrams from the source, forcing diversity at the expense of fluency. Rather than only using the paraphrases and their source labels as a fine-tuning corpus, it achieves semi-supervised learning with a compound loss that uses the paraphrases of labelled samples as positive examples, and paraphrases of unlabeled samples as negative examples to a prototypical learning objective. Blocking some particular unigrams to enhance diversity was also explored in [18] and coined as *dynamic blocking*.

Few works however focus on preserving meaning, which directly clashes with diversity. Diversity vs. fluency is the most important trade-off for textual data augmentation, and current generative models frequently loses key intent cues (such as important keywords or named entities), even when equipped with a copy mechanism. This is stressed in the Parrot framework [3], in which they add slots annotations to the training set. In [9], they extract those words at inference time, and they then average the logits from a reconstruction model trained on sets of words with the logits from a `RTT` model. Our work explores a different path: from extracted slots keywords, we leverage the possibility of insertion-based models to enforce hard constraints in the generated texts. Noting that we can also opt to expand the hard constraints using synonyms of the slots keywords

[3] The Huggingface library [27] mostly uses this approach. [4] The Fairseq library [19] also uses this approach.

(under a stochastic process), we ultimately propose an insertion-based diverse paraphrasing framework (Section 3) which leads to more fluent yet more diverse paraphrases , and ultimately with impact on the meta-learning intent detection task (Section 4.3).

For an in-depth review of existing paraphrasers, we refer the interested reader to the survey of [32]. Used for data augmentation, such paraphraser have proven very efficient to improve classification where few labelled examples, even with very low fluency are available [25,12].

3 Diverse Paraphrases Generation

Insertion models generate a sentence by expanding an ordered list of words. Contrary to standard left-to-right autoregressive models, the input words are hard constraints – they are guaranteed to be included, ordered, and they are attended by all generated tokens (attention is bidirectional even at inference time). Relying on these properties, we propose a paraphrase generation scheme that promote diversity using an insertion model (see Figure 1). We described each of the followed steps below.

3.1 Notations

Let $\mathcal{X}$ be a set of n source sentences, such that $\mathcal{X} = \{x^{(1)}, \cdots, x^{(n)}\}$. Let us consider $x^{(i)} \in \mathcal{X}$ one source sentence. For ease of reading, we use the simplified notation $x^{(i)} = x$ in the following. Let $x = (x_1, \cdots, x_\ell)$ be the sequence of tokens representing a source sentence, omitting the starting `[CLS]` and ending `[SEP]` tokens.

Our goal is to produce $Y = \{y_1, \cdots, y_m\}$, the set of m generated paraphrases for a given source sentence x. Note that each paraphrase y_j is potentially differing in length. We will note $\mathcal{Y}$ the complete paraphrase dataset, in contrast to Y, the batch of paraphrases from a particular source sentence x. Hence we have $Y^{(i)} \subseteq \mathcal{Y}, \forall x^{(i)} \in \mathcal{X}$.

3.2 Keywords Extraction

We first identify the k most important words from the source sentence in order to drive the downstream generation process. We call them keywords and note them $w = (w_1, \cdots, w_k)$. The notion of *most important* is defined as the greatest contribution to the sentence semantic. This is measured using the entropy of the word w_i in the sentence x, which can be approximated using a language model or a faster method like *tf-idf* and excluding stopwords – to name a few. Note that in our work, words are a list of tokens that should not be broken apart, like the tokens that form a word ("everyday") or an expression ("living room"). To stay close to the pre-training procedure of our non-autoregressive model, we follow [29] method, which consists in: (**1**) Splitting the sentence into words using

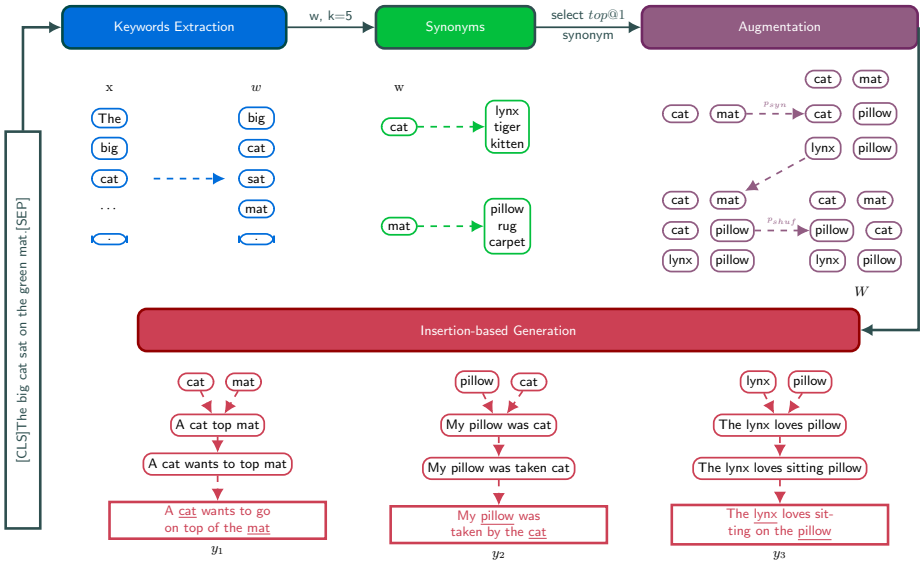

Fig. 1: Our method is two-fold. We first extract keywords and augment them via random shuffling and synonym replacement. Then, we generate paraphrases using them as an input. Those paraphrases are finally used as a data augmentation in a meta-learning framework.

an English, Regex-based word tokenizer[5]. (**2**) Applying the keyword extractor from [11], called YAKE, which was pre-trained to extract keywords, in an unsupervised way by leveraging text statistical features[6]. (**3**) Removing stopwords and duplicates, and keep k keywords, randomly chosen. We take care of treating them atomically throughout all our subsequent procedures. In addition, we empirically found that keeping the final punctuation sign, that would otherwise be removed as a stopword and preventing generation to its right[7], contributed to preserve the source sentence semantic. When our keywords extraction process fails to extract a sufficient number k of keywords, e.g. for very short sentences, we retain the k longest words. We found this strategy to be a surprisingly strong baseline, despite being very simple.

3.3 Keywords Augmentations

We augment the keyword list with two operations, namely shuffling and synonym replacement. For each keyword list w, we generate m augmentations, noted $W = (w_{i,j})_{1\leq j\leq k}^{1\leq i\leq m}$.

First, we generate a permutation for each line of W by swapping each keyword with another random word in the keyword sequence with a probability p_{shuf}.

[5] segtok: github [6] YAKE: github [7] More precisely, keeping "![end]", "?[end]", and ".[end]" as keywords

This step is meant to encourage the model to generate different alignments for better diversity. While this comes with the cost of a lower semantic alignment with the source sentence, it has been proven efficient for data augmentation [6].

Second, we replace each word with a random, but contextually relevant, synonym with a probability p_{syn}. We benchmarked several publicly available[8] synonymers using human evaluation and found `EWISER` [1], a transformer-based method that leverage the context around a word to ranks synonyms from its lemmatized WordNet synset, to be the most efficient one for our problem. A well-known downside of WordNet-based methods is the loss of semantic information resulting from the lemmatization; in practice, we found it to be acceptable for nouns or adverbs, but too significant for verbs, where the conjugation carry rich information. We reconjugate the verb synonyms to the most likely conjugation of the source verb, which qualitatively helps preserving the source semantic.

3.4 Constrained Paraphrases Generation

While we could have used a seq2seq NAR model like the Levenshtein Transformers [7], Insertion Transformers [23], or "Encode, Tag, Realize" [15], we test a different approach in this work, that is to drive our generation from a set of keywords to build around, rather than with cross-attention to the source sentence. This means we needed a language model rather than a seq2seq model. We use `POINTER` [29], which is to our knowledge, the only publicly available, large pre-trained NAR language model. `POINTER` is an insertion-based transformer model that build a sentence around an ordered list of given words. It is trained to predict the token to insert after each source token, doubling the sentence length in one iteration, and predicting [NoInsertion] everywhere when the sentence is fully built. The model heavily relies on a BERT backbone, and was fine-tuned for this progressive generation task from a pre-trained BERT checkpoint. We use this model for the generation of our diverse paraphrases Y, based on augmented keywords sequences W as an input.

We fine-tune the pre-trained `POINTER` model from [29] for each of the datasets used in our experiments (an unsupervised process). Unlike fine-tuning BERT on domain corpora, which not always provides improvements for the downstream tasks [17], this is a crucial step in our case. The paraphrase style depends on the domain at hand: for instance, the tone and turn of sentences in a dataset based on Wikipedia is different than the ones from a dataset made of user-generated queries to a chatbot. Otherwise, the model cannot use a long context to adapt the style as we do not provide the source sentence but only give it a few keywords as input.

[8] We tested the lesker from nltk (→ wordnet synset), bablify (→ bablenet synset), getalp/disambiguate, and ewiser (→ wordnet synset) wich vastly outperformed others.

4 Experiments

4.1 Datasets, Baselines and Metrics

There is no standard benchmark for NLP data augmentation, and comparing to different methods requires reproducing their results on a common dataset. Bearing this in mind, we choose to match our most strong and complex baseline settings. ProtAugment [4] was evaluated on the intent detection datasets of the DialoGLUE benchmark [16]. Summary statistics can be found in Table 1.

dataset	classes	samples	#tokens
Banking77	77	13,083	$11.7_{7.6}$
HWU64	64	11,036	$6.6_{2.9}$
Clinic150	150	22,500	$8.5_{3.3}$
Liu	54	25,478	$7.5_{3.4}$

Table 1: Summary statistics of each dataset.

Other baselines are as follows. `EDA`[25] is a paraphrasing method that use random synonyms, additions, deletions and shuffles. We selected it for its widespread usage in NLP data augmentation and simplicity. `AEDA`[12] is a simpler variant of `EDA` that random inserts semicolons as its only augmentation. It surprisingly often achieves higher results. `RTT` is a Round-Trip Translation scheme, which consists in translating from English to another language, then back to English. We construct 5 paraphrases using French, Spanish, Italian, German and Deutch intermediate languages, using public translation models from the Helsinki-NLP team for each language pair. `Bart-uni`[4] is finally our most challenging baseline. It is based on BART-base, a denoising transformer[13], and fine-tuned for paraphrasing on question-answering datasets in ProtAugment [4], which is the state-of-the-art intent detection framework, and which heavily relies on textual augmentation. We use their strongest configuration with unigram masking: At decoding times, we forbid source unigrams with a probability, which forces the beam search to diverge and create diverse generations.

We propose to automatically evaluate the quality of the generated paraphrases along two dimensions: the fluency and the diversity. We approximate the standalone fluency of the generated paraphrases with GPT2 language model perplexity (`ppl`). We use public checkpoints `distylgpt2` from Hugging Face, and compute average and standard deviation in logarithmic space (meaning we use geometric mean and standard deviation). We estimate the diversity of a set of paraphrases that share the same source using the metric proposed by [10], coined as `dist-2`, which represents the number of distinct bigrams divided by the number of distinct tokens among all those paraphrases. As a more general and softer metric than encompass both fluency and semantic conservation, we log `BLEURT` and `BERTScore` between the paraphrase and it source reference.

4.2 Experimental Settings

Our code and paraphrases are publicly available[9]

[9] https://github.com/RaphaelChevasson/DPIM

Reproduction and Hyperparameters. We use publicly available paraphrases for `RTT` and `Bart-uni` baselines[10][11] (5 per source), and run `EDA` and `AEDA` from their public repository[12][13] using the default parameters. There was no default for `AEDA` number of augmentations, so we picked 9 to align with `EDA`. For our paraphrase generation, we chose to extract $k = 3$ keywords per sentence. In order to reduce the already long generation length, we picked the minimal number which still kept reasonable information. We generate $m = 5$ augmentations per sentence to match our best baseline. For p_{syn} and p_{shuf} we search the $\{0, 0.25, 0.5, 0.75, 1\}^2$ domain. To alleviate the computational requirements, we ran the complete search only on truncated datasets with the $1,000$ first sentences (which led to $5,000$ generated paraphrases), and reran the 4 more promising on the full dataset. The grid search maximizes the validation-set classification accuracy of the ProtAugment framework with our generated paraphrases over 5 cross-validation runs. We find $p_{syn} = 0.75$, and $p_{shuf} = 0.00$ (on `BANKING77` and `HWU64`) or $p_{shuf} = 0.25$ (on `Liu` and `Clinic150`).

Insertion-Based Generation. Following `Bart-uni`, we fine-tuned our model with a single run on each of the 4 unlabeled training set. Our base model is `POINTER` wiki pretrained model[14], which is itself based on HuggingFace `bert-large-uncased` pretrained model for masked language modeling. We save and evaluate a checkpoint every 2^n and $100,000$ 8-batch training iteration, and find that training only on 500k samples (9 hours on an Nvidia Titan RTX) is sufficient, our criterion being shorter sentence length, and no fluency/diversity degradation.

Application to Meta-Learning Intent Detection. To evaluate the data augmentation potential of the paraphrases, we ran the ProtAugment meta-learning framework, using the paraphrases from each method as pseudo-labelled samples. We match [5,4] most challenging setup, with only 10 labelled samples per class and disjoint classes between the training, test and validations sets.

4.3 Analysis

[RQ1] Paraphrase Quality. Starting with a global and quantitative analysis (Figure 2), our method reach a comparable fluency (ppl) while achieving a consistently higher inter-Y diversity (dist-2), which place us in the favorable bottom-right corner of the tradeoff over every dataset. By focusing on the distribution over one dataset (Figure 3), we see that for both methods, fluency and diversity are strikingly uncorrelated. Our method have a tighter range of perplexity, which we attribute to the model being able to bidirectional attend keywords from the beginning of the generation rather than being forced out of its comfort zone mid-generation at decode time like `Bart-uni`. We also have a wider range

[10] `RTT`: paraphrases [11] `Bart-uni`: paraphrases [12] `EDA`: official code [13] `AEDA`: official code [14] `https://github.com/dreasysnail/POINTER`

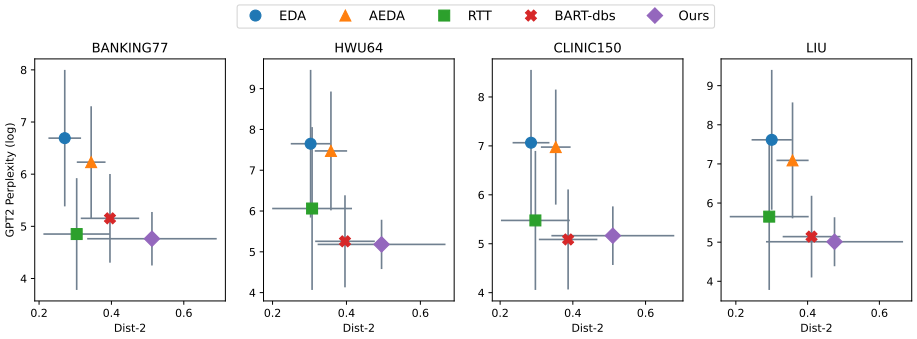

Fig. 2: Evaluation of the paraphrase quality. Lower-right corner is best as lower perplexity denotes better fluency and higher dist-2 value denotes more diversity.

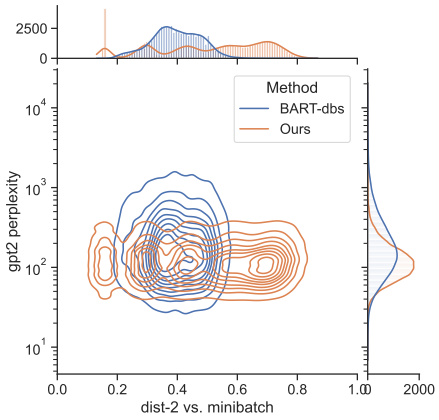

Fig. 3: Comparison of the distribution of the ppl vs. dist2 metric between `Bart-uni` and our approach for all paraphrases generated on Banking77.

Table 2: Average number of characters of the paraphrases $mean_{std}$.

	Banking77	HWU64	Liu	Clinic150
`EDA`	58_{40}	35_{15}	41_{16}	35_{6}
`AEDA`	63_{42}	36_{16}	44_{17}	38_{17}
`RTT`	55_{37}	34_{26}	40_{29}	34_{23}
`Bart-uni`	89_{31}	47_{12}	53_{14}	53_{16}
`Ours`	154_{67}	133_{46}	128_{44}	117_{41}

	Banking77		HWU64		Liu		Clinic150	
	`BLEURT`	`BERTScore`	`BLEURT`	`BERTScore`	`BLEURT`	`BERTScore`	`BLEURT`	`BERTScore`
`RTT`	$76.0_{13.1}$	$97.2_{2.4}$	$72.2_{15.2}$	$95.5_{3.5}$	$75.2_{15.7}$	$96.1_{3.4}$	$72.2_{13.5}$	$95.9_{2.9}$
`Bart-uni`	$36.7_{9.0}$	$85.3_{1.9}$	$33.4_{10.9}$	$84.5_{2.4}$	$32.3_{10.6}$	$84.2_{2.2}$	$34.7_{10.0}$	$85.2_{2.2}$
`Ours`	$43.3_{6.9}$	$86.7_{1.8}$	$41.1_{7.4}$	$85.2_{2.3}$	$38.2_{8.7}$	$85.2_{2.4}$	$42.8_{7.4}$	$85.8_{2.4}$

Table 3: Additional metrics, written with the $mean_{std}$ compact notation.

Source	I'm interested in what your exchange rates are based on.
`RTT`	**1.** I'm interested in what your exchange rates are. **2.** I'm interested in what exchange rates are based. **3.** I'm interested in what exchange rates are based on. **4.** I'm interested in your exchange rates. **5.** I'm interested in what your exchange rates are based on.
`Bart-uni`	**1.** Interesting note: How much rate should we expect when comparing with other currency markets.? **2.** How much should we expect when considering Exchange rate variations between different currency zones.? **3.** How much should we expect when considering Exchange rate variations between different currency countries.? **4.** How much should we expect when considering Exchange rate options.? **5.** How much should we expect when considering Exchange rate variations between different currency regions/yearly ranges.?
`Ours`	**1.** hi, i am extremely interested in getting an open interchange of exchange rates, but there have not been any established exchanges with exchange y. **2.** hi, i am extremely interested in purchasing, holding, exchanging and interchange. i noticed your website offer exchange rates, which are usually based on fair play. **3.** please, i am extremely interested in both in your exchange and exchange rates. i can make an interchange. is no common ground for free play. **4.** hi, i am extremely interested in finding a way to interchange funds with your services. the charges on my account are usually a few per month from the foreign accounts unit, and i have not found what seems fair. **5.** hi, i am aware and extremely interested in some in changing my currency exchange policy. i have not established a balance but there are some strange charges. i am being charged a fee per sq pound at the correct unit exchange rate.

Table 4: A batch of diverse paraphrases from Banking77 where all 3 methods perform well.

of diversity and distinguish about 5 emerging groups, which we attribute to the number of realizations of $p_{syn} + p_{shuf}$. `BLEURT` and `BERTScore` measurements shows that this advantage over `Bart-uni` extends to semantic conservation (Table 3), although still far from low-diversity methods like `RTT`.

We can further characterize this diversity by taking a qualitative look at a (x, Y) sample (Table 4). We note a strong bias in generation length w.r.t. the source sentence, which we quantify in Table 2. Our methods avoid a pattern often exhibited by `Bart-uni`: when the unigram blocking acts at the end of the beam search, it often leaves big chunks of `Bart-uni` sentences identical. From `RTT` to `Bart-uni` to `Ours`, the generation takes more liberty in interpreting and sometime adding information, sacrifice more semantic similarity, while opening more diversity. Without stronger constraints, our method is thus not suited for tasks where semantic conservation is key, but has a lot of potential for open-ended tasks like assisting creative writing or data augmentation.

[**RQ2**] **Potential for Data Augmentation.** For the latter, it seems hard to surmise if the fluency-diversity-conservation tradeoffs exhibited by our method will contribute sufficient but not excessive noise to the classifier and translate to a definitive improvement. By testing this empirically (Table 5), we observe a staggering improvement over most baseline, but not enough to surpass `Bart-uni`. Either our tradeoff advantage over this method does not carry to the classifier training signal, or the length bias negatively outweighs it.

By taking a closer look at the accuracy and tradeoff variations over our hyperparameters (Figure 4), it seems that diversity really helps while fluency

Table 5: Classification accuracy $mean_{std}$. Best result(s) are in bold.

	Banking77	HWU64	Liu	Clinic150
EDA	$84.0_{1.3}$	$77.8_{2.3}$	$80.9_{1.9}$	$93.3_{0.7}$
AEDA	$82.4_{1.2}$	$78.0_{1.6}$	$80.3_{2.2}$	$93.1_{0.4}$
RTT	$83.4_{1.5}$	$78._{1.2}$	$80.5_{2.0}$	$93.1_{0.7}$
Bart-uni	$\mathbf{87.4_{0.6}}$	$\mathbf{83.2_{1.4}}$	$\mathbf{84.9_{1.8}}$	$\mathbf{95.8_{0.3}}$
Ours	$86.3_{0.8}$	$\mathbf{83.6_{1.6}}$	$\mathbf{84.4_{1.2}}$	$\mathbf{95.4_{0.5}}$

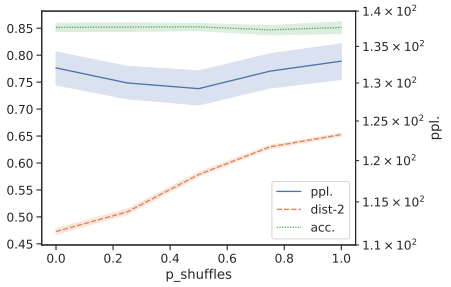

Fig. 4: Metrics ppl, dist-2 and accuracy as a function of $p_{shuffle}$ for Banking77.

have a mixed impact; we cannot however make a definitive conclusion given the low variation compared to the variance over our 5 cross-validation runs.

Considering the statistical significance of accuracies in Table 5, we nevertheless attain state-of-the-art results in 3 out of the 4 datasets, which validate the potential of NAR data augmentation, and we largely exceed every other baseline. Considering our work is the first to tackle NAR data augmentation whereas the AR methods were explored and optimized in much more details by the community, this is very encouraging.

5 Limitations

Moving Parts. While our pipeline is simple to understand, the number of moving parts (python environments, implementation-wise) make it more difficult to setup than an end-to-end method.

Length of the Paraphrases. Our current generation model systematically provide sentences longer than the sources (see Table 2), only marginally reduced by our fine-tuning on unlabeled data. While it is possible to favors [no insertion] tokens to reduce sentence length, it just displaces the problem from being out-of-domain of the classifier training to being out-of-domain of the expander training. To solve the root problem without full cross-attention, we think providing the (augmented) source length as an input and using a model trained with oracle length, but still allowed to deviate from it, is the most promising direction.

6 Conclusion

In this work, we proposed an approach to replace AutoRegressive models by more flexible and controllable Non-AutoRegressive ones in the paraphrase generation task, with a deep dive in state-of-the-art (meta-learning) data augmentation for low-resource fine classification. Capitalizing on the open-endedness of these

tasks, we proposed an extensible pipeline that achieve satisfactory results, while not even requiring cross-attention. We compared its behaviour against a number of diverse baselines, including two strong autoregressive ones, and found that the non-autoregressive model have a definite advantage for handling constrained diversity. We discussed the strengths and weaknesses of our method and open-source it to foster future research, as we ultimately think there is still untapped potential in the control offered by NAR generation methods.

Acknowledgments

This work was supported by the Futur & Ruptures program at Institut Mines-Télécom [2019-2022]. It was granted access to the HPC resources of IDRIS under the allocation 2022-AD011013590 made by GENCI.

References

1. Bevilacqua, M., Navigli, R.: Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In: Proceedings of ACL. pp. 2854–2864. ACL (2020)
2. Cao, Y., Wan, X.: DivGAN: Towards diverse paraphrase generation via diversified generative adversarial network. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2411–2421. ACL, Online (Nov 2020)
3. Damodaran, P.: Parrot: Paraphrase generation for nlu. (2021)
4. Dopierre, T., Gravier, C., Logerais, W.: PROTAUGMENT: Unsupervised diverse short-texts paraphrasing for intent detection meta-learning. In: Proceedings of ACL-IJCNLP. pp. 2454–2466 (2021)
5. Geng, R., Li, B., Li, Y., Zhu, X., Jian, P., Sun, J.: Induction networks for few-shot text classification. In: Proceedings of EMNLP-IJCNLP. pp. 3904–3913 (2019)
6. Goyal, T., Durrett, G.: Neural syntactic preordering for controlled paraphrase generation. In: Proceedings of ACL. pp. 238–252. ACL (2020)
7. Gu, J., Wang, C., Zhao, J.: Levenshtein transformer. In: NeurIPS. vol. 32 (2019)
8. Guo, Y., Liao, Y., Jiang, X., Zhang, Q., Zhang, Y., Liu, Q.: Zero-Shot Paraphrase Generation with Multilingual Language Models. arXiv:1911.03597 (2019)
9. Guo, Z., Huang, Z., Zhu, K.Q., Chen, G., Zhang, K., Chen, B., Huang, F.: Automatically paraphrasing via sentence reconstruction and round-trip translation. In: Zhou, Z.H. (ed.) Proceedings of IJCAI. pp. 3815–3821 (2021)
10. Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M., Callison-Burch, C.: Comparison of diverse decoding methods from conditional language models. In: Proceedings of ACL. pp. 3752–3762 (2019)
11. Jorge, A., Campos, R., Jatowt, A., Nunes, S., Rocha, C., Cordeiro, J.P., Pasquali, A., Mangaravite, V.: Text2story workshop - narrative extraction from texts. SIGIR Forum pp. 150–152 (2018)
12. Karimi, A., Rossi, L., Prati, A.: AEDA: An easier data augmentation technique for text classification. In: Findings of ACL: EMNLP. pp. 2748–2754. ACL (2021)
13. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of ACL (2020)

14. Mallinson, J., Sennrich, R., Lapata, M.: Paraphrasing revisited with neural machine translation. In: Proceedings of EACL. pp. 881–893 (2017)
15. Malmi, E., Krause, S., Rothe, S., Mirylenka, D., Severyn, A.: Encode, tag, realize: High-precision text editing. In: Proceedings EMNLP-IJCNLP. pp. 5054–5065
16. Mehri, S., Eric, M., Hakkani-Tür, D.: Dialoglue: A natural language understanding benchmark for task-oriented dialogue (2020)
17. Merchant, A., Rahimtoroghi, E., Pavlick, E., Tenney, I.: What happens to BERT embeddings during fine-tuning? In: Proceedings of the BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP. pp. 33–44. ACL (2020)
18. Niu, T., Yavuz, S., Zhou, Y., Keskar, N.S., Wang, H., Xiong, C.: Unsupervised paraphrasing with pretrained language models. In: Proceedings of EMNLP. pp. 5136–5150. ACL (Nov 2021)
19. Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., Auli, M.: fairseq: A fast, extensible toolkit for sequence modeling. In: Proceedings of NAACL-HLT: Demonstrations (2019)
20. Qian, L., Qiu, L., Zhang, W., Jiang, X., Yu, Y.: Exploring diverse expressions for paraphrase generation. In: Proceedings of EMNLP-IJCNLP. pp. 3173–3182 (2019)
21. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020)
22. Sharma, L., Graesser, L., Nangia, N., Evci, U.: Natural language understanding with the quora question pairs dataset (2019)
23. Stern, M., Chan, W., Kiros, J., Uszkoreit, J.: Insertion transformer: Flexible sequence generation via insertion operations. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of ICML. vol. 97, pp. 5976–5985. PMLR (2019)
24. Vijayakumar, A., Cogswell, M., Selvaraju, R., Sun, Q., Lee, S., Crandall, D., Batra, D.: Diverse Beam Search for Improved Description of Complex Scenes. Proceedings of AAAI **32**(1) (2018)
25. Wei, J., Zou, K.: EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of EMNLP-IJCNLP. pp. 6382–6388 (2019)
26. Wieting, J., Mallinson, J., Gimpel, K.: Learning paraphrastic sentence embeddings from back-translated bitext. In: Proceedings of EMNLP. pp. 274–285. ACL, Copenhagen, Denmark (Sep 2017)
27. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. In: Proceedings of EMNLP: System Demonstrations. pp. 38–45. ACL (2020)
28. Yang, Y., Zhang, Y., Tar, C., Baldridge, J.: PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In: Proceedings of EMNLP-IJCNLP. pp. 3687–3692. ACL, Hong Kong, China (2019)
29. Zhang, Y., Wang, G., Li, C., Gan, Z., Brockett, C., Dolan, B.: POINTER: Constrained progressive text generation via insertion-based generative pre-training. In: Proceedings of EMNLP. pp. 8649–8670. ACL (2020)
30. Zhang, Y., Baldridge, J., He, L.: PAWS: Paraphrase adversaries from word scrambling. In: Proceedings of NAACL/HLT,. pp. 1298–1308. ACL (2019)
31. Zhao, S., Wang, H.: Paraphrases and applications. In: Coling : Paraphrases and Applications–Tutorial notes. pp. 1–87 (2010)
32. Zhou, J., Bhat, S.: Paraphrase generation: A survey of the state of the art. In: Proceedings of EMNLP. pp. 5075–5086. ACL (2021)