BLEU it All Away!
Refocussing SE ML on the Homo Sapience

Applis, L.H.

Publication date
2022

Document Version
Submitted manuscript

Citation (APA)
Applis, L. H. (2022). *BLEU it All Away! Refocussing SE ML on the Homo Sapience*. Abstract from International Summer School on Search- and Machine Learning-based Software Engineering, Cordoba, Spain.

BLEU it All Away!
Refocussing SE ML on the Homo Sapience

Leonhard Applis
TU Delft
L.H.Applis@tudelft.nl

***Abstract*—Many tasks in machine learning for software engineering rely on prominent NLP metrics, such as the BLEU or ROUGE score. The metrics are under heavy criticism themselves within the NLP community, but the SE community adapted them for lack of better alternatives. Within this paper, we summarize some of the problems with common metrics at the examples of code and look for alternatives. We argue that our only hope is the worst of all possible options: Humans.**

I. Introduction

In ancient Greece, Hephaistos was accompanied by servant automatons to help around his forge, freeing him to spent his time on true masterpieces. This hellenic ideal of automation lives up to this day and has its renaissance with software engineers: Tedious tasks such as writing tests[1] or documentation [2] are shifted towards automation to give room for the developers creativity. The narrative is great — the results are often humbling. Presentations for Githubs CoPilot [3] pick cherries, but thorough investigations usually lead to disturbing or amusing results. How did we end up here ?

One issue are the metrics. For this paper we focus on Documentation Generation [2], which is lately often interpreted as a translation task from source code to human language (i.e. english) and draws a lot from NLP research, such as sequence-to-sequence models [4] but also the most common metric BLEU [5]. In recent work, Gehrmann et al. [6] criticised the metric driven approaches and publications in NLP (specifically generation tasks). Among their primary findings are that ⓐ people *blindly* use existing datasets without manual inspection, sampling, etc. ⓑ people rarely inspect output manually or involve *end-users* ⓒ all publications use BLEU for lack of better options or for acceptance at a venue. Gehrmann et al. proposition is as compelling as it is easy: Instead of using *big data* and arguably weak metrics, center the evaluation around a group of expert users.

The remainder of this paper first highlights some flaws with BLEU in documentation generation in Section II and elaborates on the proposed solution in Section III. While we cover only one domain briefly, we consider this to be a general critique applicable for most domains. We close in Section IV by arguing that we need to change the course of SE-ML-Metrics, and while the proposal might not be perfect, it is one we haven't tried in a long time.

II. The Flaws

BLEU [5] is a metric to evaluate quality of translation and text-generation techniques. It compares the overlap of n-grams in a produced text compared to one or more reference texts, where commonly a four-gram is used, as it correlates closest to human acceptance[7]. There is wide criticism on BLEU[8],[9], but we highlight issues specific to the domain of software engineering:

① While BLEU takes n-grams into account, many pieces of programming language and documentation will produce a solid score despite sometimes contrary meaning. With common tokenization, *return (a + b) == (b - a);* and *return (a - b) == (b + a);* scores near perfect in BLEU. Some publications opt for one-gram BLEU, for which the above example gives an optimal match.

② Eghbali et al. [**?**] investigated the BLEU-Scores of randomly chosen samples from within different corpora. In a corpus of english literature there was a BLEU of ≈20%, comparing two random elements from Javas scores ≈40%. This is stunning insofar as these numbers form the expected *baseline* if we could produce random elements that follow the same distribution. In their initial publication, CodeBERT produces a BLEU-Score of 17.65% [2], which is 2.4% worse than drawing random elements.

③ Unlike natural language, programming languages (and their documentation) invent new words frequently. This is known as the open vocabulary problem [10] and is addressed in SE mostly by encodings. Prominent are BytePairEncoding[10] and Subword-Splitting [11]. Both increase the number of tokens - hitherto they benefit the BLEU score. It poses two primary issues: ③a it is harder to evaluate and compare the metric if evermore strings become attached. ③b the research field itself becomes overburdened in experiment-complexity only for the sake of metrics.

III. The Options

One approach to address the issues is to blame the metric; *The BLEU is dead, long live the BLEU*. One can easily stitch together *"MetaBLEU"* that combines normal BLEU for language-representation and stopword-cleaned BLEU for content-coverage. Similar *fixes* for BLEU have been proposed [12][**?**][13], mostly ductaping the underlying problems. These are not done in bad faith, on the contrary they fit perfectly in the current paradigm of ML publications — more data, more features and better tuned models can be used with the same benchmark and promise a safe academic voyage. But as a research field, we will hit dead ends by the need for ever

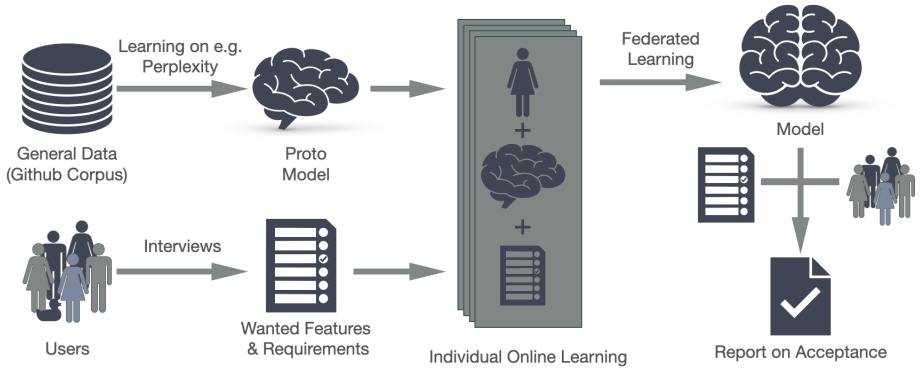

Fig. 1. Proposed Pipeline for SE Model Training

more data, a cacophony of metrics and incredible computation times.

Apart from these issues, another question remains: Is a model with good BLEU score useful? The only way to answer this is to ask real humans, real users. Gehrmann et al. [6] come to a similar conclusion and argue for model-cards based on expert-based qualitative analysis. Theoretically there are few fields easier to change evaluations than Software Engineering; Software Engineers produce the data, ML-libraries, models, metrics and are the final users.

The concrete suggestion (shown in Figure 1) is to start models with metrics, and produce *proto models* that cover a basic understanding of vocabulary and distributions. The downstream-tasks should be tuned with humans in the loop, by rating various aspects of the specific task (content, quality of language, feedback time, inter-prediction quality, etc.). Rating-Criteria should be derived from and with the final users, in a fashion like requirements engineering. This pipeline is similar to e.g. CodeBERT [14], which learns general perplexity on Code and then is fine-tuned for the specific task and language. The BERT-Core and the Code-Addition would form the *proto model* and the downstream-task of documentation generation would be done in active learning with experts rating samples, instead of *blind* metrics. Pieces for this novel pipeline are available and tested [15][16], and could themselves make great use-cases for reinforcement learning and federated learning.

IV. Conclusion

Following metrics down the rabbit hole lead us into a ML wonderland of free publications — but for outsiders we are just kids in an asylum. If our goal is to make models that are useful to developers and help them in their business, the only metric we really have to maximize is their feedback. No developer tries to write documentation with a certain BLEU score, hence we should turn our back on these proxy-metrics. We should trust our users that they know what they want, and change our own research to accomodate for their needs.

References

[1] G. Fraser and A. Arcuri, "Evosuite: automatic test suite generation for object-oriented software," in *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, 2011, pp. 416–419.

[2] "Codexglue: A benchmark dataset and open challenge for code intelligence," 2020.

[3] Github. [Online]. Available: https://copilot.github.com/

[4] B. Li, M. Yan, X. Xia, X. Hu, G. Li, and D. Lo, *DeepCommenter: A Deep Code Comment Generation Tool with Hybrid Lexical and Syntactical Information*. New York, NY, USA: Association for Computing Machinery, 2020, p. 1571–1575. [Online]. Available: https://doi.org/10.1145/3368089.3417926

[5] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[6] S. Gehrmann, E. Clark, and T. Sellam, "Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text," *arXiv preprint arXiv:2202.06935*, 2022.

[7] D. Coughlin, "Correlating automated and human assessments of machine translation quality," in *Proceedings of Machine Translation Summit IX: Papers*, 2003.

[8] C. Callison-Burch, M. Osborne, and P. Koehn, "Re-evaluating the role of bleu in machine translation research," in *11th conference of the european chapter of the association for computational linguistics*, 2006, pp. 249–256.

[9] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proceedings of the second international conference on Human Language Technology Research*, 2002, pp. 138–145.

[10] R.-M. Karampatsis, H. Babii, R. Robbes, C. Sutton, and A. Janes, "Big code!= big vocabulary: Open-vocabulary models for source code," in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE, 2020, pp. 1073–1085.

[11] H. Babii, A. Janes, and R. Robbes, "Modeling vocabulary for big code machine learning," *arXiv preprint arXiv:1904.01873*, 2019.

[12] S. Ren, D. Guo, S. Lu, L. Zhou, S. Liu, D. Tang, N. Sundaresan, M. Zhou, A. Blanco, and S. Ma, "Codebleu: a method for automatic evaluation of code synthesis," *arXiv preprint arXiv:2009.10297*, 2020.

[13] T. Sellam, D. Das, and A. P. Parikh, "Bleurt: Learning robust metrics for text generation," 2020. [Online]. Available: https://arxiv.org/abs/2004.04696

[14] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang *et al.*, "Codebert: A pre-trained model for programming and natural languages," *arXiv preprint arXiv:2002.08155*, 2020.

[15] B. Settles, "Active learning literature survey," 2009.

[16] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[17] A. Shimorina and A. Belz, "The human evaluation datasheet 1.0: A template for recording details of human evaluation experiments in nlp," 2021. [Online]. Available: https://arxiv.org/abs/2103.09710