Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4221–4226
Marseille, 20-25 June 2022

MTLens: Debugging Machine Translation Systems Based on Their Output

Shreyas Sharma, Kareem Darwish, Lucas Aguiar Pavanelli, Thiago Castro Ferreira
Mohamed Al-Badrashiny, Kamer Ali Yuksel, Hassan Sawaf
aiXplain Inc.
California, USA
{shreyas.sharma, kareem.darwish, lucas.pavanelli, thiago, mohamed, kamer, hassan}@aixplain.com

Abstract

The performance of Machine Translation (MT) systems varies significantly with inputs of diverging features such as topics, genres, and surface properties. Though there are many MT evaluation metrics that generally correlate with human judgments, they are not directly useful in identifying specific shortcomings of MT systems. In this demo, we present a benchmarking interface that enables improved evaluation of specific MT systems in isolation or multiple MT systems collectively by quantitatively evaluating their performance on many tasks across multiple domains and evaluation metrics. Further, it facilitates effective debugging and error analysis of MT output via the use of dynamic filters that help users hone in on problem sentences with specific properties, such as genre, topic, sentence length, etc. The interface can be extended to include additional filters such as lexical, morphological, and syntactic features. Aside from helping debug MT output, it can also help in identifying problems in reference translations and evaluation metrics.

Keywords: Machine Translation, Evaluation, Error Analysis

1. Introduction

The performance of Machine Translation (MT) systems may vary significantly across genres, topics, sentence surface properties (ex. length, punctuation, POS tags, etc.), and styles. Though much work has been done on MT evaluation in terms of metrics and datasets, performing error analysis efficiently is cumbersome and time-consuming. In this demo, we present an intuitive MT benchmarking interface for quantitatively evaluating and debugging MT systems together or in isolation along different properties of the text, such as genre and topic, and using a variety of MT evaluation metrics and evaluation test sets. Our proposed system helps researchers and practitioners identify problem areas for their MT model and whether these problems are specific to their models or shared by other models. Such can greatly simplify error analysis and help guide further research and development efforts. Additionally, our system has a modular design that can be extended to handle additional metrics and test set features. Though not directly shown in the demo, users can choose the test sets, metrics, and models of their choice, and the system would perform benchmarking per their preferences. When the benchmark results are ready and stored in a database, the user can interact with the evaluation results and perform system debugging. In the online demo (`https://bit.ly/3IdHzBv`), we show the capability of the system by evaluating 6 commercial MT systems[1] on 2 language pairs using 4 different test sets from OPUS[2] and 5 different evaluation metrics.

The contributions of this demo are as follows:

- We present an intuitive user interface that allows users to filter MT results by model, metric, test set, topic, or any other feature. Filtration allows users to see worst (or best) performing sentences that match specific criteria.
- The system allows users to compare an MT model against other models and to identify sentences where there is high or low variance between systems. In all such comparisons, users are allowed to apply any of the aforementioned filters.
- We show also that the interface can help identify mistakes in reference translations as well as shortcomings of evaluation metrics.

2. Related Work

Leaderboards and dashboards are becoming increasingly common for comparing and evaluating performance of various machine learning systems, including MT systems. ((Coleman et al., 2017; Olson et al., 2017; Mattson et al., 2020; Liu et al., 2021; Kiela et al., 2021)). For proper comparison of MT systems, much effort has been devoted to devising automatic metrics for properly evaluating MT output with and without a reference translation or translations. There are multiple metric types that measure similarity to a reference translation (ex. BLEU (Papineni et al., 2002)); measure post-editing effort (ex. TER (Snover et al., 2006)); or estimate human judgements with and without a reference translation (ex. COMET_DA and COMET_QE respectively (Rei et al., 2020)). Though such metrics quantitatively score MT in a manner that generally correlates with human evaluation, they do not elucidate why a particular MT output is better or worse than another output or which types of errors are most common.

[1] For legal reasons, the system names are anonymized.
[2] `https://opus.nlpl.eu/`

Llitjós et al. (2005) proposed a classification of the most common MT errors. Subsequent tools, such as BLAST (Stymne, 2011), attempted to aid manual annotation of MT errors. Such tools can be configured to handle a variety of error types. Kirchhoff et al. (2007) attempted to correlate between MT evaluation metric scores with input characteristics to ease error analysis. Popović and Ney (2011) proposed a method based on word error rate measures in an effort to automatically classify the error types. Further, other recent works also focused on reliability and bias analysis (Liu et al., 2021) as well as hardware and software performance (Mattson et al., 2020). Though automation is important, automatic classification is limited to a predefined set of error types. In this demo, we present an interface that easily shows the best/worst performing sentences given a set of filters. It is extensible to handle any property of the input or the output, allowing for fine-grained segmentation of test cases. Further, it allows users to quickly debug single MT systems or compare multiple MT systems against each other.

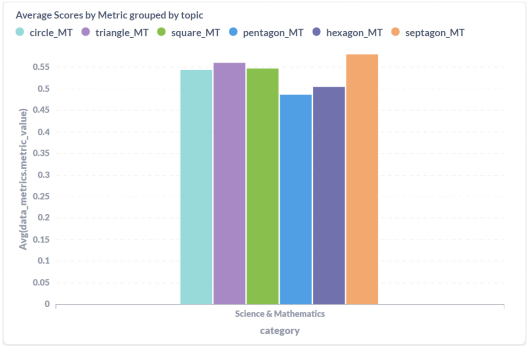
3. System Description

3.1. System Design

The demo interface is based on two primary components, namely a back-end database and a front-end business intelligence visualization toolkit. For the back-end database, we used mySQL[3], which is a popular open-source relational database management system. For a given test sentence, we stored the following information: sentence ID, test set name, topic, language pair (source and target language), source sentence, reference manual translation, translations from all the providers, values of all evaluation metrics for all providers along with mean and standard deviation for each metric, and sentence length. To speed up search, we designed an entity-relationship schema, where different tables store subsets of the information and tables are linked together using primary and foreign keys. For example, we have a table that contains general information about each sentence such as source sentence, reference translation, and topic, and another table that contains all the translations from all the different providers. The two tables are linked using sentence IDs. For the front-end visualization tool, we used Metabase[4], which is an open-source business intelligence tool, which connects to a back-end database and creates plots based on the results of SQL queries. The SQL queries are allowed to have custom "WHERE" statements, which allows for optional and dynamic filtration on different column values. Using such filtration, users of our interface can filter on a variety of features such as test set, topic, metric, translation provider, and sentence length to identify the sentences with the best/worst translations. Figure 1 shows a sample SQL statement with the corresponding output plot.

[3]https://www.mysql.com/

[4]https://github.com/metabase/metabase




```
SELECT Avg(data_metrics.metric_value), data_metrics.provider,
    data_main.category
FROM data_main
    JOIN data_metrics on data_metrics.sentence_id = data_main.sentence_id
WHERE 1 = 1
[[AND data_main.language_pair = {{Language_Pair}}]]
[[AND data_main.category = {{Topic}}]]
[[AND data_main.corpus = {{corpus}}]]
[[AND data_metrics.metric_name = {{metric}}]]
[[AND data_main.source_length < {{Max_length}}]]
[[AND data_main.source_length > {{Min_length}}]]
GROUP BY data_metrics.provider, data_main.category
```

Figure 1: Sample plot with associated SQL statement. Optional conditions are put between square brackets.

3.2. Metrics

We used three types of MT evaluation metrics, namely reference similarity metrics, human evaluation estimation metrics, and referenceless metrics.

Reference similarity metrics measure the similarity between a reference translation (or translations) and machine translation output, with higher similarity leading to higher scores. They range from ones that strictly use the surface forms, such as BLEU (Papineni et al., 2002) and character n-gram F-Score (chrF) (Popović, 2015), to ones that use semantic similarity, such as BERTScore (Zhang et al., 2019). In the demo, we utilized BLEU and chrF.

Human evaluation estimation metrics attempt to learn the scores that a human would have provided to machine translation outputs. These are generally considered among the most robust measures of machine translation quality. For this type of metrics, we used COMET_DA (Rei et al., 2020), which reportedly correlates better with human scores compared to BLEU, chrF, and BERTScore.

Referenceless metrics attempt to compare/rank machine translation outputs in the absence of a ground-truth reference translation. They rely on multilingual embeddings to compute the similarity between the source sentence and machine translation outputs. They are considered less reliable than metrics that utilize reference translations. We used two such metrics namely COMET_QE (Rei et al., 2020) and MTQuality, which we developed internally and uses cosine similarity between source sentence and machine translation output using Language-Agnostic BERT Sentence Embedding (LaBSE) (Feng et al., 2020).

3.3. Test sets

For the demo, we used 4 test sets for two language pairs namely English→German (EN-DE: 12,500 sen-

tences) and German→English (DE-EN: 12,491 sentences). The test sets were sampled from 4 different test sets from OPUS[5], which is a large public database of translated texts. The 4 sets and sample sizes are as follows:

Test set	EN-DE	DE-EN
OpenSubtitles[6]	2,293	2,291
TED2013[7]	2,857	2,856
TED2020[8]	4,786	4,786
Europarl[9]	2,564	2,558

We translated all the sentences in the test sets using 6 different commercial translation providers. For legal purposes, we are anonymizing their names and using the following names instead: circle_MT, triangle_MT, square_MT, pentagon_MT, hexagon_MT, and septagon_MT. For topic classification, we used a publicly available BERT model that is fine tuned on the Yahoo! Answers dataset[10].

3.4. Interface

The interface of the demo is composed of 4 main sections as follows:

- Figure 3.a has available filters, which include language pair, test set (corpus), topic, evaluation metric, MT provider, and minimum and maximum source sentence lengths. Filter values can be provided manually or by clicking on the items in Figure 3.b as in Figure 2.
- Figure 3.b includes general information about test sets (size, language pair breakdown, test set corpora breakdown, and topic breakdown), evaluation metrics, and evaluation metric values (overall or for specific topics). Clicking on any of the items (ex. language pair, topic, or metric) would automatically update the filters and would update the number of test sentences that match the filter criteria and the sentences show in Figures 3.c and 3.d.
- Figure 3.c compares the different MT results based on differences in standard deviation between the different results (for a specific evaluation metric). Specifically, the tables show the sentences with the lowest standard deviation (ranked by lowest overall performance to show the sentence where all the systems are not producing good results) and by the highest standard deviation (to show the sentences where some providers are providing very good translations while others are producing very poor translations). Identifying sentences where all providers are failing can highlight errors in the reference translations and universal issues that plague MT systems in general.

[5] https://opus.nlpl.eu/

[10] https://huggingface.co/fabriceyhc/bert-base-uncased-yahoo_answers_topics

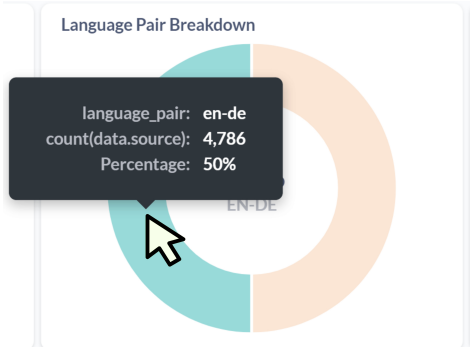


Figure 2: Highlighting and selecting a value to filter on.

- Figure 3.d shows the worst and best performing sentences for a specific provider as measured by a specific metric and that match any filter that was applied (test set, topic, or sentence length). The ranking is done by metric value first and then by metric standard deviation across all providers (in descending order). Showing the worst and best examples that match a user's criteria can help the user identify areas where their system is failing (or succeeding) particularly compared to other MT vendors.

4. System Debugging Using Our System

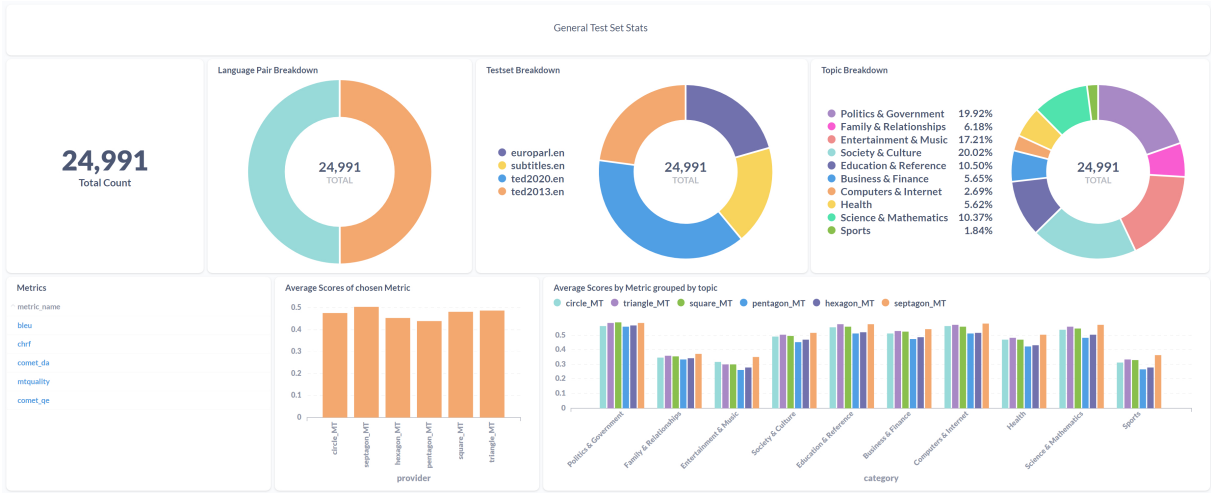
This section showcases sample scenarios with associated screenshots on how the system can used to debug MT output.

Scenario 1 (Figure 4): This scenario shows the subsetting of the test set to see the impact of filters (language pair: EN-DE; metric: COMET_DA; test set: TED2020; sentence length between 10 and 50 characters) on different topics while showing how many sentences match the filters. The Figure shows that "Computers & internet" topic is performing the worst while "Sports" where performing the best.

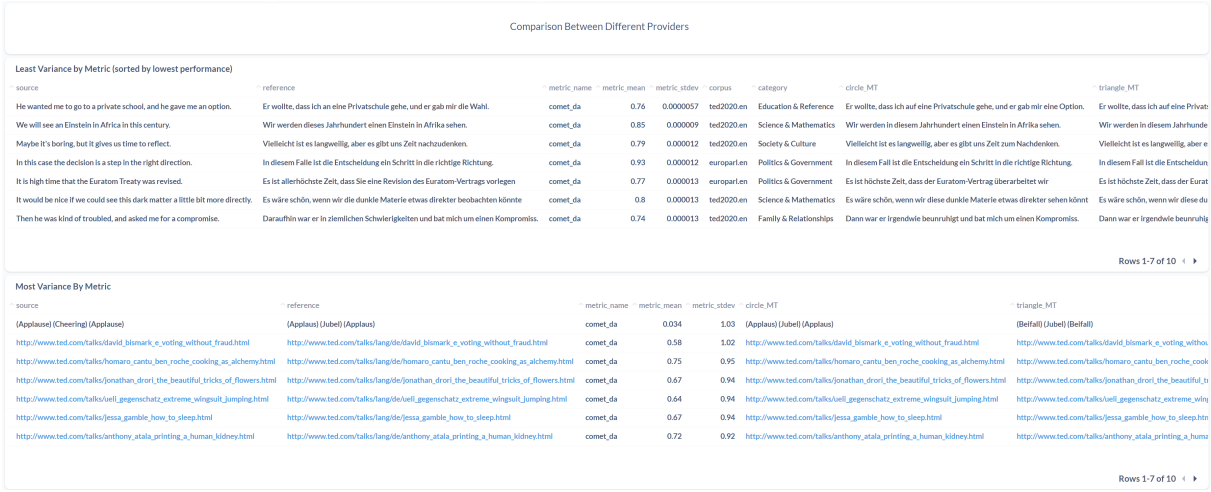
Scenario 2 (Figure 5): This scenario shows the worst performing sentences for all systems and the sentences where there is most variability between vendors (applied filters: language pair: EN-DE; topic: Society & Culture; metric: BLEU). Looking at the sentences where all the systems are performing poorly, there seems to be mistakes in some of the reference translations. For the sentences with the most variability, the output translations from different vendors are provided in the interface.

Scenario 3 (Figure 6): This scenario shows the worst and best performing sentences for a specific provider (applied filters: language pair: DE-EN; metric: COMET_QE; provider: square_MT). Looking at the sentences where the system is performing poorly, there seems to be some shortcomings of the evaluation metric for some of the sentences, and in other cases the machine translation system produced incorrect translations.

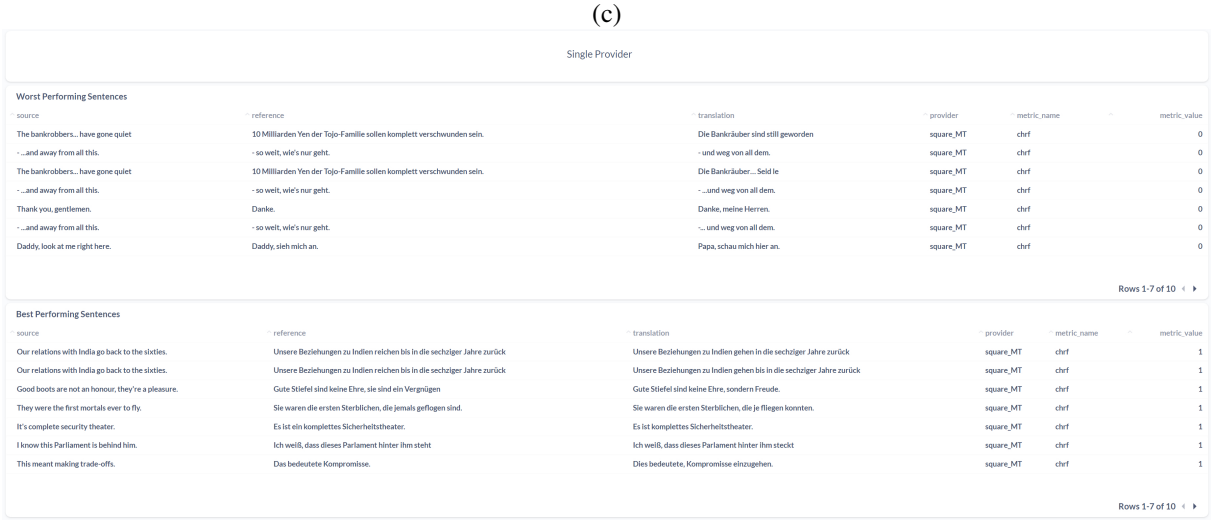
(a)



(b)



(c)



(d)

Figure 3: Screenshot of interface.

Figure 4: Subsetting the test set to see the impact of filters.

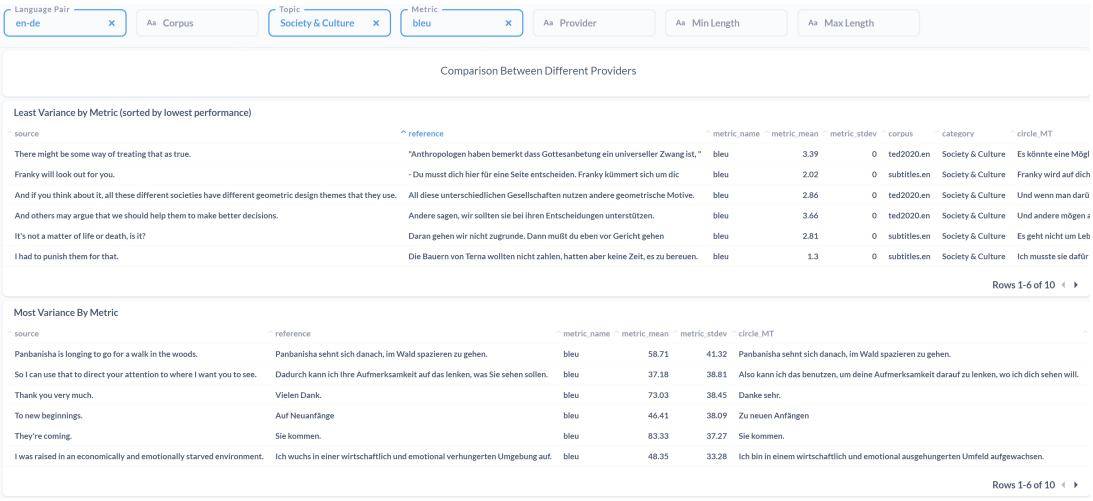


Figure 5: Identifying sentences where all providers do poorly, or there is a large variance between translations.

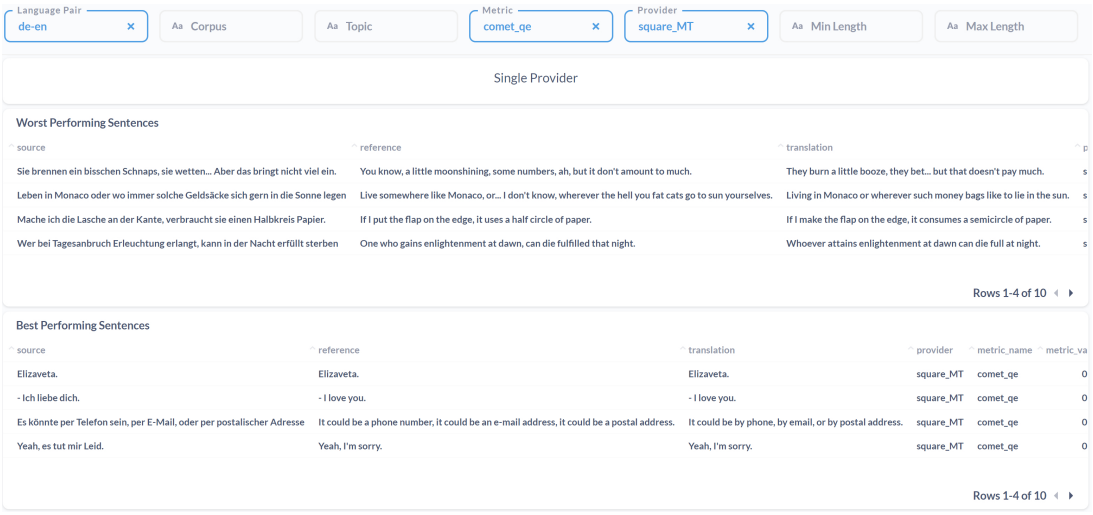


Figure 6: Identifying sentences where a single provider performs the best or the worst.

5. Conclusion

In this demo, we show an intuitive interface that allows user to debug and analyze MT systems in isolation or against each other through the use of a variety of filters such as test sets, metrics, input sentence properties, topic, etc. The interface can be extended to include additional filters such as lexical, morphological, and syntactic features. As the scenarios we presented suggest, the application of filters can help identify poorly performing topics, which my indicate gaps in training data, mistakes in reference translations, and even shortcomings in evaluation metrics. Other filtration scenarios may uncover other problems that may be intrinsic to a specific MT system or common across multiple systems. To access the complete platform and other benchmarking services, aiXplain[11] membership is required.

For future work, we plan to integrate correlations between features of input sentences and evaluation metrics. This would allow us to introduce more sophisticated filters and sentence ranking functions, such as showing the worst performing sentences across all MT systems given the morphological feature that correlates most with a given evaluation metric. Another interesting direction entails providing additional filters such as lexical, morphological, and syntactic features. We also plan to integrate human-in-the-loop evaluation in our platform to supplement our automatic metrics.

6. References

Coleman, C., Narayanan, D., Kang, D., Zhao, T., Zhang, J., Nardi, L., Bailis, P., Olukotun, K., Ré, C., and Zaharia, M. (2017). Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101):102.

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2020). Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., et al. (2021). Dynabench: Rethinking benchmarking in nlp. *arXiv preprint arXiv:2104.14337*.

Kirchhoff, K., Rambow, O., Habash, N., and Diab, M. (2007). Semi-automatic error analysis for large-scale statistical machine translation systems. *Proceedings of the machine translation summit (MT-Summit), Copenhagen*.

Liu, P., Fu, J., Xiao, Y., Yuan, W., Chang, S., Dai, J., Liu, Y., Ye, Z., Dou, Z.-Y., and Neubig, G. (2021). Explainaboard: An explainable leaderboard for nlp. *arXiv preprint arXiv:2104.06387*.

Llitjós, A. F., Carbonell, J. G., and Lavie, A. (2005). A framework for interactive and automatic refinement of transfer-based machine translation. In *Proceedings of the 10th EAMT Conference: Practical applications of machine translation*.

Mattson, P., Reddi, V. J., Cheng, C., Coleman, C., Diamos, G., Kanter, D., Micikevicius, P., Patterson, D., Schmuelling, G., Tang, H., et al. (2020). Mlperf: An industry standard benchmark suite for machine learning performance. *IEEE Micro*, 40(2):8–16.

Olson, R. S., La Cava, W., Orzechowski, P., Urbanowicz, R. J., and Moore, J. H. (2017). Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData mining*, 10(1):1–13.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Popović, M. and Ney, H. (2011). Towards automatic error analysis of machine translation output. *Computational Linguistics*, 37(4):657–688.

Popović, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. (2020). Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Stymne, S. (2011). Blast: A tool for error analysis of machine translation output. In *Proceedings of the ACL-HLT 2011 System Demonstrations*, pages 56–61.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

[11] https://aixplain.com/