Text Characterization Toolkit (TCT)

Daniel Simig[*], **Tianlu Wang**[*], **Verna Dankers**[*+], **Peter Henderson**[‡*],
Khuyagbaatar Batsuren[†], **Dieuwke Hupkes**[*], **Mona Diab**[*]
[*]Meta AI, [+]University of Edinburgh, [†]National University of Mongolia, [‡]Stanford University
{danielsimig,dieuwkehupkes,mdiab}@meta.com

Abstract

We present a tool, *Text Characterization Toolkit (TCT)*, that researchers can use to study characteristics of large datasets. Furthermore, such properties can lead to understanding the influence of such attributes on models' behaviour. Traditionally, in most NLP research, models are usually evaluated by reporting single-number performance scores on a number of readily available benchmarks, without much deeper analysis. Here, we argue that – especially given the well-known fact that benchmarks often contain biases, artefacts, and spurious correlations – deeper results analysis should become the de-facto standard when presenting new models or benchmarks. TCT aims at filling this gap by facilitating such deeper analysis for datasets at scale, where datasets can be for training/development/evaluation. TCT includes both an easy-to-use tool, as well as off-the-shelf scripts that can be used for specific analyses. We also present use-cases from several different domains. TCT is used to predict difficult examples for given well-known trained models; TCT is also used to identify (potentially harmful) biases present in a dataset.

1 Introduction

NLP technology has progressed tremendously over the recent decades with significant advances in algorithms and modeling. Yet, by comparison, our understanding lags behind significantly for datasets (including all datasets types in the model life cycle: training, validation, evaluation) that contribute to model performance. This is mostly due to the lack of frameworks, methods, and tools to draw insights into datasets, especially at scale.

Most NLP models, to date, are evaluated using a relatively small number of readily available evaluation benchmarks, that are often created automatically, or via crowd-sourcing (e.g. Bowman et al., 2015; Wang et al., 2018; Williams et al., 2018; Zellers et al., 2018). It is well-known that most popular (evaluation) datasets are rife with biases, dataset artefacts and spurious correlations, and are prone to be solved with shortcuts (Gardner et al., 2021; Kiela et al., 2021). Presenting models with *adversarial examples* for which those biases or correlations do not hold, often results in stark performance drops (e.g. Linzen, 2020; McCoy et al., 2019; Jia and Liang, 2017; Chen et al., 2016; Tsuchiya, 2018; Belinkov et al., 2019). At best, using datasets with such known issues might result in overestimation of a models' capability on the task in question, which may not be reflective of how well they can execute this task in more realistic scenarios. More worrying, however, is that training or finetuning on datasets that contain biases and artefacts may result in models implementing undesired, biased behaviour (e.g. Rudinger et al., 2018; Blodgett et al., 2016).

Additionally, datasets are usually treated as homogeneous collections of text, performance for which is captured in a single number – even though there is often a substantial difference between the difficulty/complexity of different examples in a dataset (e.g. Sugawara et al., 2022). Research papers rarely report thorough analysis of performance broken down by characteristics of the data set examples ignoring underlying patterns performance numbers may reflect. The problem is exacerbated by the pervasiveness of benchmarks coupled with a leaderboard competitive culture, where what counts most is system rank.

In part, this may be due to the fact that deeper analysis of results – especially when a number of different datasets is involved – is complex and time-consuming, and there are no standard frameworks or protocols that practitioners can resort to. The problem is even more pervasive, where we curate datasets for development and evaluation. How we curate, create, select data plays a critical role in understanding our models. Many NLP models (even beyond text) require up/down sampling of specific

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations, pages 72–87
November 20–23, 2022.

types of data. These processes should rely on principled characterization of data for any given model.

To this end, we believe that the existence of a standard toolkit that provides an easy to use set of tools and metrics allowing researchers to analyze and systematically characterize datasets, *at scale*, involved in the model life cycle, while gaining insights into the relationship between model performance and data properties could become more common place.

In this paper, we introduce the *Text Characterization Toolkit*[1] (TCT), which aims to enable researchers to gain a detailed understanding of the datasets and models they create – with minimal effort. TCT is inspired by the Coh-Metrix toolkit (Graesser et al., 2004), a collection of over 100 diverse text characteristics intended for use for text analysis in various applications. TCT offers these capabilities at scale by design. While TCT can process a dataset of 20000 paragraphs in less than a minute on a MacBook Pro laptop, the very same library, for instance, can also be used as part of a distributed PySpark pipeline to compute text characteristics for a full snapshot of Common Crawl (3.1B web pages) in a matter of hours. While text characteristics in TCT are currently implemented for the English language only, the framework is designed to scale to any new language via configuration of new resource files and SpaCy backends.

In this paper we present:

1. A repository of text metrics that can help reveal (latent) patterns in datasets coupled with model performance on these datasets;
2. A set of off-the-shelf analysis tools that researchers can use in a simple notebook to study properties of the dataset and the influence of those properties on model behaviour;
3. A framework that enables the community to share, reuse and standardize metrics and analyses methods/tools;
4. Use cases that demonstrate the efficacy of TCT in practice covering Language Model prompting, Translation, and Bias Detection.

With these contributions, we aspire to help the NLP community in particular and the AI community at large improve how we assess NLP models, and get closer to a scenario where providing detailed results' analyses becomes the standard for NLP research. Equally important, we believe that

[1] https://github.com/facebookresearch/text_characterization_toolkit

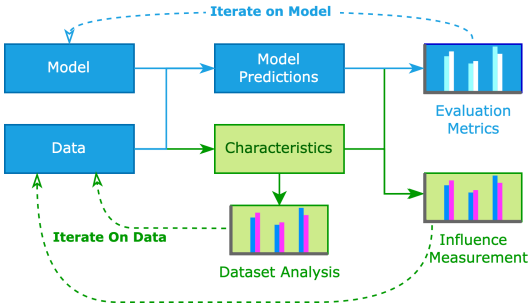


Figure 1: Text Characterization Toolkit extends model evaluation to provide insights about the role of data.

TCT could be an effective tool for data selection for both training and evaluation, in particular at scale.

2 [TCT] Text Characterization Toolkit

TCT consists of two main components:

- A framework for defining and computing text characteristics.
- A collection of analysis tools that help users interpret text characteristics and evaluate results with respect to these characteristics.

As illustrated by Figure 1, the workflow of extending a standard evaluation process with TCT is typically the following:

1. Given a dataset, extract text fragments from each data point into a file. For instance, a QA dataset comprising text fragments could be individual questions, whereas in document summarization, the text fragments would be the documents themselves.
2. Using the TCT command line tool, compute the characteristics of the text fragments and dump the results in a file. Furthermore, one might use the default characteristics already included in the framework or define their own specific metric in a new configuration file.
3. Create a Python notebook and include the TCT library. Using tools from this library load the computed characteristics and other evaluation specific data, then use some of the analysis tools provided by the framework: one might analyze the dataset itself (e.g. to identify spurious correlations or biases) or jointly analyze model evaluation metrics and text characteristics (e.g. through correlation analysis between TCT features and models' evaluation set accuracy).
4. Use the results of the analysis to improve the dataset, the model, or the evaluation protocol – for example by extending evaluation data

Category	Example Metrics
Descriptive	Word Count Sentence Length
Lexical Diversity	Type-Token Ratio MTLD
Complexity	Left Embeddedness # of NP modifiers
Incidence Scores	Different POS tags Types of connectives
Word Property	Age of Acquisition Concreteness

Table 1: Categories of characteristics currently implemented. See Appendix A for an exhaustive list.

with examples where a model is expected to perform poorly or focusing evaluation on a challenging subset of the test data.

Concrete examples of the workflow above are described in §3 and in Appendix B. The rest of this section provides more details on the two important components of the framework.

2.1 Text Characteristics

While the majority of the characteristics found in TCT is motivated by metric classes in Coh-Metrix (Graesser et al., 2004), we have included new data bases for existing metrics and added entirely new metrics. At the time of writing, there are 61 characteristics implemented in TCT. An overview of the main categories of currently implemented characteristics can be found in Table 1. The toolkit provides a standardized framework to implement, configure, and compute these metrics. Adding a new metric is as simple as implementing two Python functions: one that loads any required resource (such as a word database) and initializes computation, and one that computes the metric given these resources and an input text.

2.2 Analysis tools

To further decrease the effort required to carry out text characteristics based analysis, we provide an initial set of analysis tools that users can use out of the box. We encourage users to contribute their own implementations of TCT-based analyses to the toolkit, to allow for re-use in the future development of datasets and models. The current functionality of the toolkit includes:

1. Visualising distributions of different characteristics;
2. Visualising a pairwise correlation matrix for the characteristics, as illustrated in Figure 2;

Figure 2: An illustration of a correlation plot from TCT. Users might find these plots valuable to find unexpected correlations or to interpret results of multi-variable regression models.

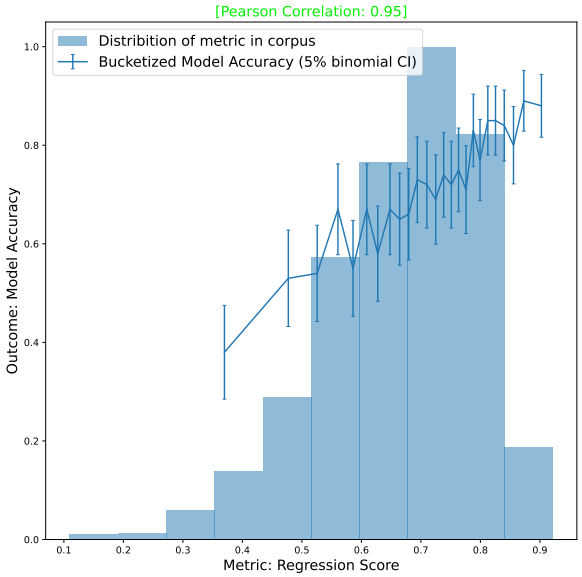
3. Visualising correlations between individual characteristics and outcomes (e.g., accuracy), as illustrated in Figure 4;
4. Fitting a model on all characteristics to outcomes (logistic regression and random forests are supported currently) and analyzing a model's predictive power and coefficients. This is illustrated in Figure 3.

3 Example Use Cases

In order to demonstrate the ability of TCT to produce meaningful and actionable insights, we provide 3 examples of its use on real world data. For each one of these use cases, a thorough description of the experimental setup and results is included in Appendix B and reference notebooks are provided in the `examples` directory of the TCT repository.

Predicting Accuracy of OPT Baselines Open Pre-trained Transformer Language Models (OPT, Zhang et al. 2022) are a sequence of publicly released LLMs that span model scales from 125M to 175B parameters. While scaling thees models lead to significant gains on numerous benchmarks, they still occasionally produce results that would seem trivially wrong to any human.

With the help of TCT, we attempted to better understand the robustness of these models quantitatively. We use the multi-variable logistic regression



(a) Model accuracy versus the regression output variable. Using the multi-variable regression tool can surface larger variations in model performance.

(b) TCT displays the most important feature coefficients to help users understand what characteristics contributed to the results shown in Figure 3a

Figure 3: Results from the TCT multi-variable regression analysis tool. Screenshots taken from the OPT analysis described in §3.

analysis tool to fit a model that predicts the accuracy of the 6.7B parameter OPT model on the HellaSwag common-sense inference task (Zellers et al., 2019) based on simple characteristics such as mean word length and concreteness. Using this model we can identify subsets of the test data with model accuracy as low as 40% and as high as 90% – shown on Figure 3.

Fluctuations in Translation Performance Using TCT we show how translation performance of the NLLB model (Costa-jussà et al., 2022) using the HuggingFace pipeline (Wolf et al., 2019) fluctuates as a function of sample characteristics, like the number of sentences - Figure 4 shows the output of TCT in this analysis. This performance

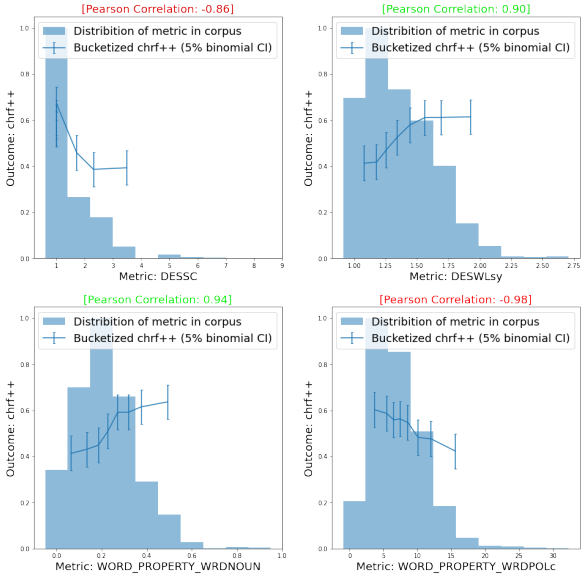


Figure 4: Model performance of a translation model in function of various characteristics of the text translated, produced by the TCT single-variable analysis tool. Appendix B.3 describes the experimental setup in detail.

heterogeneity can be fixed by segmenting sentences before using the pipeline, showing that TCT can help debug model pipelines even with many layers of abstraction.

Gender Bias in Co-reference Resolution By computing genderedness metrics on co-reference labels and using these metrics as inputs to the analysis tools, we reproduce the results of Zhao et al. (2018) showing that models perform much worse when the stereotypically associated gender of an occupation does not match the gender of the pronominal reference.

4 Related Work

There exist several tools that offer similar functionality to TCT in some specific respects. DataLab (Xiao et al., 2022) is a tool for detailed data analysis that, among other things, allows users to inspect datasets through the lens of a few text characteristics such as text length, lexical diversity and gender-related features. The *Know Your Data*[2] tool allows for inspection of image data, it surfaces spurious correlations, biases and imbalances in datasets. However, both tools do not connect model behavior to properties of datasets.

Collins et al. (2018) predicts overall hardness of classification datasets based on label statistics and a few text characteristics such as readability

[2] https://knowyourdata.withgoogle.com/

and lexical diversity. ExplainaBoard (Liu et al., 2021) focuses on model performance analysis and provides a model performance breakdown by simple attributes of data points such as text length, providing a functionality most similar to our work.

Our toolkit distinguishes itself by including a much wider variety of text characteristics. As a concrete example, ExplainaBoard does not provide features similar to WRDPRP2 (personal pronoun incidence), WRDPOLc (mean word polysemy) or DESSC (number of sentences per text), each playing an important role in our chosen use cases analyses described in §3. TCT also features a multivariable analysis tool that can identify larger variations in model performance compared to simpler, single-variable analyses.

By packaging our toolkit as a simple Python library used in notebooks – in contrast to the previously described feature-rich systems – we also intend to minimize the effort needed to both use it as well as contribute to it (eg. crowd sourcing more functionality).

The Coh-Metrix tool (Graesser et al., 2004) collected the most diverse set of text characteristics to our knowledge, designed for various use cases in linguistics and pedagogy. However, the tool, developed in 2004, is slow as it is designed to process a single document at a time, relatively difficult to access, and the underlying word databases are outdated. Our toolkit aims to make a subset of the Coh-Metrix metrics easily accessible to the NLP community, while simultaneousy addressing the scale impediment noted in the original Coh-Metrix tool.

5 Future Work

As illustrated in §2 we envision TCT to be a framework and an associated tool that allows for community contributions, crowdsourcing even more functionality and use cases. Future work involves usage of the tool:

Firstly, we encourage creators of new datasets to use TCT as a data annotation tool, to extract a wide range of dataset statistics in a straightforward manner, and report about them in academic publications for transparency. Such statistics could be included in datasheets and data cards (Gebru et al., 2021), and they can aid in outlier detection during data (pre-)processing and cleaning.

We also prompt dataset creators to perform statistical analyses capturing which features are predictive of the gold targets *before* further training computational models, to ensure one is aware about potential shortcut learning opportunities due to biases in the dataset. Naturally, not all correlations are bad or avoidable – e.g. consider sentences containing the word 'fantastic' that are likely to have a positive label in sentiment analysis – but others are good to be aware of when working with a dataset – e.g. consider a natural language inference task where all sentences with the label 'entailed' have an atypical average word length. Such analyses could be included in a 'cautions' section with a dataset's release.

A third type of usage would be by owners of new models, that, on the one hand, use TCT to measure whether some dataset characteristics are predictive of success and failure by their model, and, on the other hand, provide performance on subclasses of samples. One may already know that model performance is lower for longer sentences, but what about performance on different readability classes, classes with varying amounts of causal connectives or different ratings for syntactic complexity (e.g. SYNLE)? TCT will help answer those questions. Understanding how the model performance fluctuates for different data subsets provides further understanding in model robustness, and can, in turn, improve datasets' quality if model owners report back on biases identified in datasets.

Limitations

Text Characteristics in our framework have varying levels of coverage depending on their type. Word property based characteristics, for example, are limited by the coverage of the word databases that back them – this can be limited even for English.

Despite covering a sizeable number of metrics in Coh-Metrix's original toolkit, we still don't cover all possible relevant metrics for text processing, especially at different levels of granularity (eg. document/paragraph/character levels).

We are mostly limited to the text modality which could cater to speech data however TCT lacks analysis tools and metrics for more speech oriented datasets such as prosody and syllables for instance.

While we plan to extend the framework to multiple languages in the near future, language coverage of backend word databases and NLP pipelines such as WordNet (Miller, 1995) or SpaCy (Honnibal et al., 2020) will affect the ability to scale the number of languages supported.

References

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Marc Brysbaert, Paweł Mandera, Samantha F Mccormick, and Emmanuel Keuleers. Word prevalence norms for 62,000 English lemmas.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv e-prints*, pages arXiv–2204.

Edward Collins, Nikolai Rozanov, and Bingbing Zhang. 2018. Evolutionary data measures: Understanding the difficulty of text classification tasks. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 380–391, Brussels, Belgium. Association for Computational Linguistics.

Max Coltheart. 2018. The MRC Psycholinguistic Database:. *https://doi.org/10.1080/14640748108400805*, 33(4):497–505.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv e-prints*, pages arXiv–2207.

Christiane Fellbaum. 2010. WordNet. *Theory and Applications of Ontology: Computer Applications*, pages 231–243.

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers 2004 36:2*, 36(2):193–202.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. Age-of-acquisition ratings for 30,000 English words.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*,

pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021. ExplainaBoard: An Explainable Leaderboard for NLP. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the System Demonstrations*, pages 280–289.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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.

Kincaid Peter, Fishburne Jr Robert, L Rogers Richard, S Chissom Brad, and P J. Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. 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 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Saku Sugawara, Nikita Nangia, Alex Warstadt, and Samuel Bowman. 2022. What makes reading comprehension questions difficult? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6951–6971, Dublin, Ireland. Association for Computational Linguistics.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yang Xiao, Jinlan Fu, Weizhe Yuan, Vijay Viswanathan, Zhoumianze Liu, Yixin Liu, Graham Neubig, and Pengfei Liu. 2022. DataLab: A Platform for Data Analysis and Intervention. pages 182–195.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 4791–4800.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Singh Koura, Anjali Sridhar, Tianlu Wang, Luke Zettlemoyer, and Meta Ai. 2022. OPT: Open Pre-trained Transformer Language Models.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. 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 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Category	Metric Key	Description
Descriptive	DESPC	Number of paragraphs
	DESSC	Number of sentences
	DESWC	Number of words
	DESPL	Average number of sentences per paragraph
	DESPLd	Standard deviation of paragraph lengths (in sentences)
	DESPLw	Average number of words per paragraph
	DESSL	Average number of words per sentence)
	DESSLd	Standard deviation of sentence lengths (in words)
	DESWLsy	Average word length (syllables)
	DESWLsyd	Standard Deviation of word lengths (in syllables)
	DESWLlt	Average word length (letters)
	DESWLltd	Standard Deviation of word lengths (in letters)
Lexical Diversity	LDTTRc	Type-Token Ratio (TTR) computed over content words
	LDTTRa	Type-Token Ratio (TTR) computed over all words
	LDMTLD	Measure of Textual Lexical Diversity (MTLD)
	LDHDD	HD-D lexical diversity index
Syntactic Complexity	SYNLE	Left embeddedness: average words before main verb
	SYNNP	Number of modifiers per noun phrase, mean
	SYNMEDpos	Average edit distance between POS tags of consecutive sentences
	SYNMEDwrd	Average edit distance between consecutive sentences
	SYNMEDlem	Average edit distance between consecutive sentences (lemmatized)
	SYNSTRUTa	Sentence syntax similarity, adjacent sentences, mean
	SYNSTRUTt	Sentence syntax similarity, all combinations, mean
Readability	RDFRE	Flesch Reading Ease (Peter et al.)
	READFKGL	Flesch-Kincaid Grade Level (Peter et al.)

Table 2: List of paragraph-level metrics currently supported by TCT

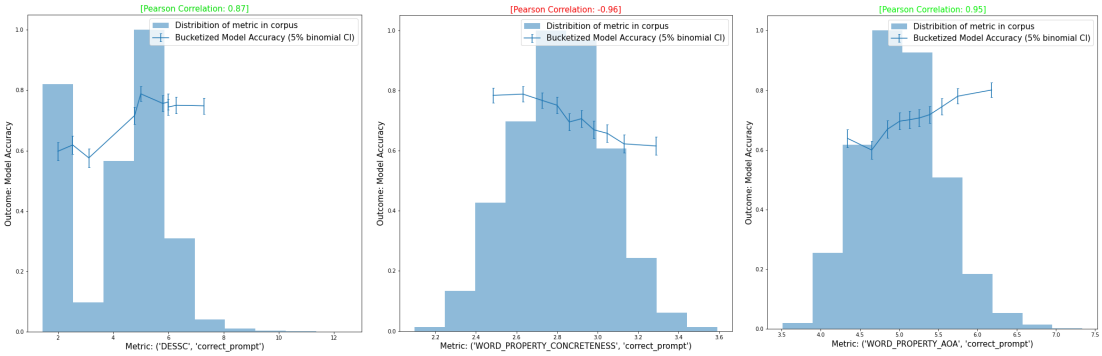
A List of Text Characteristics

Tables 2 and 3 list all currently implemented metrics along with a short description. For a large number of metrics the Coh-Metrix website[3] provides further details. For word property metrics we also document the source database in Table 3.

[3] http://cohmetrix.memphis.edu/cohmetrixhome

Category	Metric Key	Description
Incidence Scores	TOKEN_ATTRIBUTE_RATIO_ALHPA	Alphanumerical tokens
	TOKEN_ATTRIBUTE_RATIO_DIGIT	Tokens consisting of digits
	TOKEN_ATTRIBUTE_RATIO_PUNCT	Punctuation tokens
	TOKEN_ATTRIBUTE_RATIO_URL	URLs
	TOKEN_ATTRIBUTE_RATIO_EMAIL	E-mail addresses
	WORD_SET_INCIDENCE_WRDPRP1s	First person singular pronouns
	WORD_SET_INCIDENCE_WRDPRP1p	First person plural pronouns
	WORD_SET_INCIDENCE_WRDPRP2	Second person pronouns
	WORD_SET_INCIDENCE_WRDPRP3s	Third person singular pronouns
	WORD_SET_INCIDENCE_WRDPRP3p	Third person plural pronouns
	WORD_SET_INCIDENCE_CNCCaus	Causal connectives
	WORD_SET_INCIDENCE_CNCLogic	Logical connectives
	WORD_SET_INCIDENCE_CNCTemp	Temporal connectives
	WORD_SET_INCIDENCE_CNCAdd	Additive connectives
	WORD_SET_INCIDENCE_CNCPos	Positive connectives
	WORD_SET_INCIDENCE_CNCNeg	Negative connectives
	WORD_PROPERTY_WRDNOUN	Incidence score for POS tag 'PROPN', 'NOUN'
	WORD_PROPERTY_WRDVERB	Incidence score for POS tag 'VERB'
	WORD_PROPERTY_WRDADJ	Incidence score for POS tag 'ADJ'
	WORD_PROPERTY_WRDADV	Incidence score for POS tag 'ADV'
Word Property	WORD_PROPERTY_WRDFRQc	Mean Word frequency*, content words
	WORD_PROPERTY_WRDFRQa	Mean Word frequency*, all words
	WORD_PROPERTY_WRDFRQmc	Min Word frequency*
	WORD_PROPERTY_WRDFAMc	Mean Familiarity+, content words only
	WORD_PROPERTY_WRDCNCc	Mean Concreteness+, content words only
	WORD_PROPERTY_WRDIMGc	Mean Imagability+, content words only
	WORD_PROPERTY_WRDMEAc	Mean Meaningfulness+
	WORD_PROPERTY_WRDPOLc	Mean Polysemy†
	WORD_PROPERTY_WRDHYPn	Mean Hypernymy† (nouns)
	WORD_PROPERTY_WRDHYPv	Mean Hypernymy† (verbs)
	WORD_PROPERTY_WRDHYPnv	Mean Hypernymy† (verbs and nouns)
	WORD_PROPERTY_AOA	Mean Age of Acqusition (Kuperman et al.)
	WORD_PROPERTY_AOA_MAX	Max Age of Acqusition (Kuperman et al.)
	WORD_PROPERTY_CONCRETENESS	Mean Concreteness (Brysbaert et al., 2014)
	WORD_PROPERTY_PREVALENCE	Mean Prevalence (Brysbaert et al.)
	WORD_PROPERTY_PREVALENCE_MIN	Minimum Prevalence (Brysbaert et al.)

Table 3: List of word-level metrics currently supported by TCT. Common underlying word databases: *SpaCy (Honnibal et al., 2020) +MRC (Coltheart, 2018) † WordNet (Fellbaum, 2010)



(a) Sentence Count (b) Concreteness (c) Age of Acquisition

Figure 5: Correlations between text characteristics and accuracy for OPT experiments,

B Sample Use Cases

In this section we demonstrate how users can gain actionable insights on existing evaluation data using TCT, with minimal amount of additional work. The examples provided below can be reproduced by Python notebooks we included in the examples folder of the TCT repository.

B.1 Predicting Accuracy of OPT Baselines

Despite the recent success of large pre-trained language models (LLM), there are still ongoing debates regarding how good they really are, and how to evaluate that. After all, LLMS such as PaLM (Chowdhery et al., 2022) or DeBERTa (He et al., 2020) have saturated the performance on benchmarks, even outperforming human scores at times, but, at the same time, there still exist a myriad of seemingly trivial scenarios in which they fail.

Experimental setup In this demonstration we offer an alternative approach: We take existing data from the evaluation of the 6.7B OPT baseline (Zhang et al., 2022) then attempt to use simple data characteristics to identify interpretable subsets of the dataset on which OPT's performance substantially differs from its overall high accuracy. We use the HellaSwag task (Zellers et al., 2019),[4] a common-sense inference task that is trivial to solve for humans but challenging for LLMs.

To evaluate OPT models on this task, prompts corresponding to different choices were scored with the LLMs and the answer with the lowest perplexity was considered to be the choice of the model. For each data point in the test set, we consider two text fragments: the prompt corresponding to the correct answer and the concatenation of all the prompts corresponding to incorrect answers (see Table 6 for an example). With a single command using the `command_line_tool.py` we compute characteristics for the extracted texts and load the results into a notebook. We also load the result of the model evaluation, which is a single binary variable per data point describing whether the model predicted the right answer.

Results First, we inspect correlations between individual metrics and model performance. This analysis tool orders data points with respect to a particular TCT metric, groups them into buckets of 100 data points and computes model accuracy for each bucket. We find several different data characteristics that show high correlation with model performance, for example number of sentences per prompt or average concreteness (Brysbaert et al., 2014) of words. A visualisation of these results is shown in Figure 5.

Secondly, we employ the TCT class named `PredictFromCharacteristicsAnalysis` to fit a logistic regression model using all characteristics to predict whether the model will yield a correct answer for a particular data point. The tool computes the regression scores on a held out part of the dataset and visualizes model accuracy with respect to this score per data bucket, as shown in Figure 3a. We find more variance between the best and worst performing buckets compared to the single variable analysis. On the bucket with the highest predicted score the OPT baselines yield a 0.9 accuracy, but in the lowest scoring

[4]We chose the HellaSwag task for this demo as it had sufficiently many examples in the test set and showed the most interesting correlations out of all tasks the model was evaluated on prior.

Correct Prompt	Roof shingle removal: A man is sitting on a roof. He starts pulling up roofing on a roof.
Incorrect Prompts	Roof shingle removal: A man is sitting on a roof. He is using wrap to wrap a pair of skis. Roof shingle removal: A man is sitting on a roof. He is ripping level tiles off. Roof shingle removal: A man is sitting on a roof. He is holding a rubik's cube.

Figure 6: Example of text features extracted from the HellaSwag evaluation of the OPT model

bucket the accuracy is below 0.4, which approaches the random baseline of 0.25. To interpret the fitted regression model, we inspect its coefficients,[5] illustrated in Figure 3b. Interestingly, coefficients for given characteristics often yield opposite signs associated with the correct and incorrect answers, indicating that they are in fact, on their own, predictive of the correctness of an answer. For instance, the *DESWLlt* metric (mean number of letters per word) has coefficients of -0.44 and 0.62 for the `correct_prompt` and `incorrect_prompts` features, respectively.

We argue that such analyses are useful from two perspectives: i) Analyses that uncover patterns in what characteristics make examples difficult help us improve our understanding of how well a model has in fact learned the task we intended it to. This, in turn, provides a better estimate of the wider applicability of a model. ii) If one knows which text characteristics lead to poor performance from LLMs, one could improve the dataset's coverage for characteristics associated with low model performance – e.g. one could curate data points including tokens with low concreteness scores.

Table 6, Figure 5 and Figure 3b illustrate the model analysis process described previously in this section.

B.2 Gender Bias in Co-reference Resolution Models

Second, we would like to illustrate how TCT can aid in identifying biases in NLP systems, by revealing gender bias in coreference resolution systems.

Experimental Setup We use a coreference resolution model proposed by Lee et al. (2017) and the WinoBias dataset (Zhao et al., 2018). The model is evaluated using exact match to compute accuracy. To capture gender statistics, we configure a new Word Property metric "genderedness" based on Labor Force Statistics[6] and compute it on two text fragments (the two spans of the ground truth co-reference). A higher genderedness score represents that the occupation is associated with a female stereotype and vice versa. For pronominal references, we assign 100 to female ones (e.g. "she", "her") and 0 to male ones (e.g. "he", "his"). We add the difference between the two characteristics as an additional feature for analysis.

Results The analysis obtained by the TCT toolkit is illustrated in Figure 7. There is a negative correlation between model accuracy and the genderness difference between the occupation and the pronominal reference. In other words, if a female stereotypical occupation and a male pronoun co-occur in a test example (e.g. "nurse" and "he") or a male stereotypical occupation and a female pronoun (e.g. "constructor" and "she") co-occurs, the model is more likely to make a wrong prediction.

B.3 NLLB: Interpretable Fluctuations of Translation Performance

A third example of a task that could benefit from using TCT in analyses is *Neural Machine Translation* (NMT). We apply TCT to source sentences to identify patterns in translation success for an off-the-shelf NMT system.

Experimental Setup To investigate performance heterogeneity in translation models, we use the No Language Left Behind 1.3B distilled model and the English-Russian validation split of the multi-domain dataset from the same work (Costa-jussà et al., 2022). We use the HuggingFace transformers translation pipeline for easy inference (Wolf et al., 2019). We extract translations using the pipeline, and employ the `chrf++` metric to measure success per individual data point (Popović, 2017).[7] Using the toolkit we characterize the English source data with default settings.

[5] Since inputs to the regression were scaled to unit variance, direct comparison of coefficients is meaningful

[6] https://github.com/uclanlp/corefBias

[7] Note: we use this as it has better per data point properties than other corpus statistics like BLEU (Papineni et al., 2002).

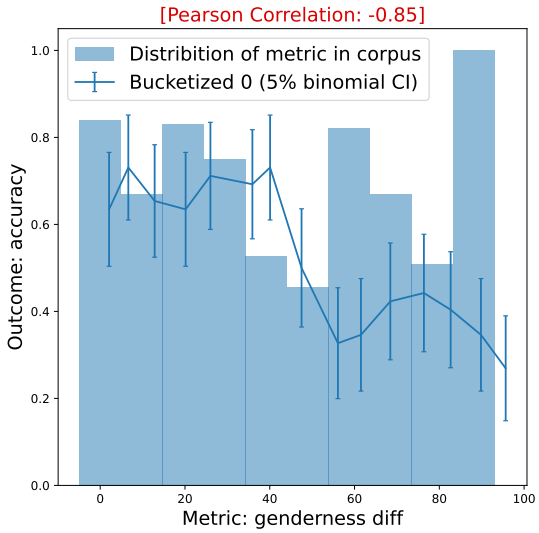


Figure 7: Genderness difference hurts the performance of a coreference resolution model.

Results Surprisingly, we find significant heterogeneity as seen in Figure 4. In particular, more sentences, more verbs, polysemy for content words, and chat-like messages lead to performance drops. Conversely, more nouns and words with more syllables correlate with better `chrf++` scores.

The driver of this heterogeneity may be deceptive. The HuggingFace translation pipeline does not keep track of the underlying model's training distribution. It would not know that the NLLB model was trained on sentence pairs and the evaluation data contains multi-sentence datapoints. An appropriate way to match the training distribution would instead be to split by sentences and translate individual sentences before re-concatenating. In fact, if we take this approach, we find that performance levels out with the biggest improvements coming from the largest sources of heterogeneity (Figure 10). This demonstration shows the power of TCT for debugging model workflows. With many layers of abstraction, it is easy to forget that underlying models are likely trained on a particular data distribution.

Additional Details Figure 8 shows the distribution of data-characterized performance for NLLB using the HuggingFace translate tool with no modification (other than increasing the maximum generated length to 512 tokens). Figure 9 shows the distribution of chrf++ scores for NLLB with sentence segmentation[8]. We pass each sentence in a batch to the segmentation pipeline before re-concatenating them by adding a space between sentences (since we only use English and Russian for this demonstration this is an appropriate concatenation method). Finally, Figure 10 shows the treatment effect. For each sentence we subtract the segmented NLLB chrf++ score from the unsegmented chrf++ score. Then we run the TCT toolkit on this outcome measure. We show that performance increases are such that they level out performance heterogeneity to some extent.

In Table 4 we demonstrate how the unsegmented NLLB model can drop out entire portions of the translation in multi-sentence validation datapoints. This is likely what leads to performance drops. The segmented version fixes this. As such, we suggest that TCT should be run at eval time even when using a known model that has been validated in the past. Different environmental setups can lead to failure modes such as this one that can be difficult to detect without data characterization.

[8]Sentence segmentation by SpaCy (Honnibal et al., 2020)

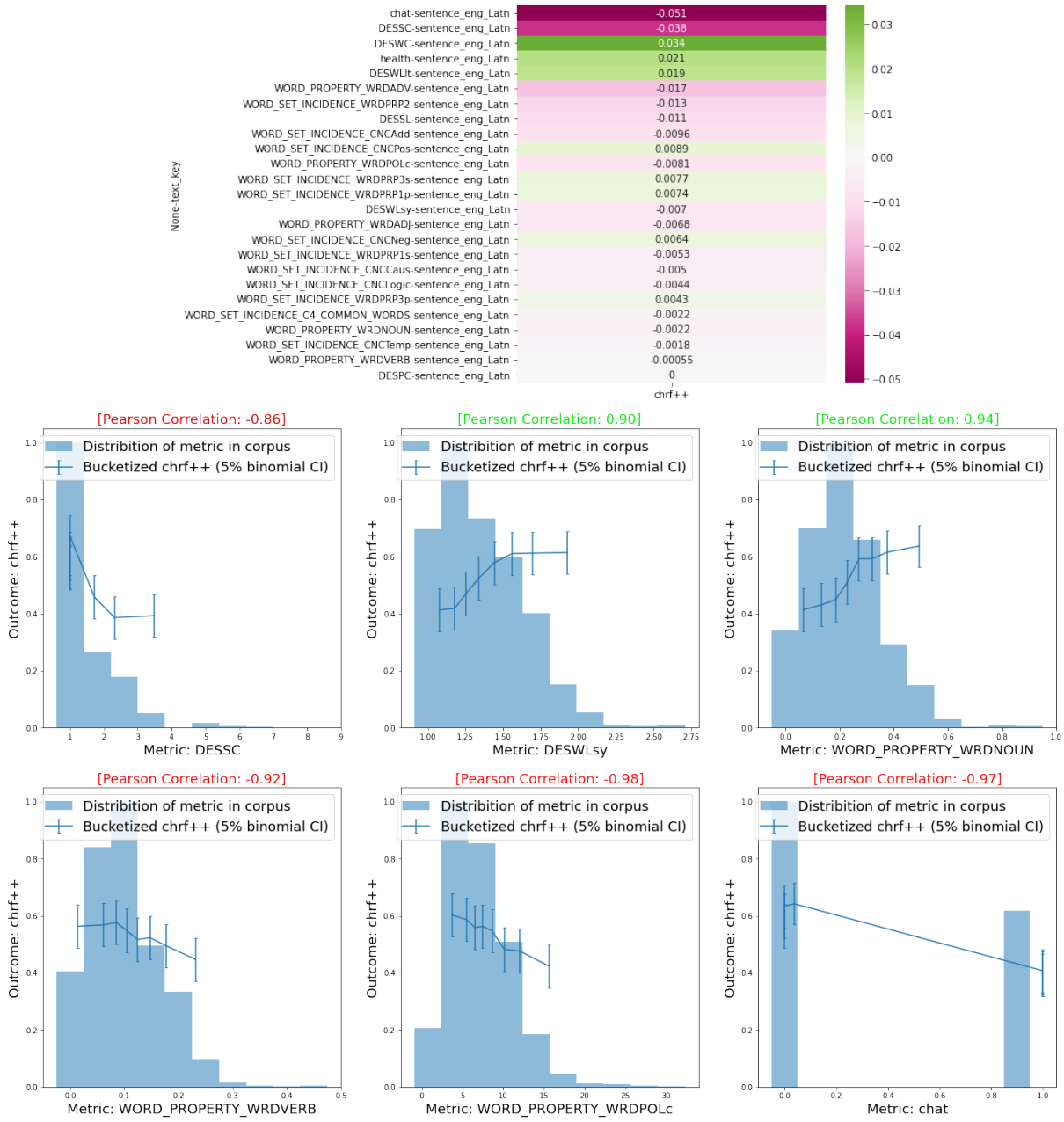


Figure 8: Top: A regression on the chrf++ score of a model pipeline using NLLB with no segmentation. Positive values indicate improved score, lower values indicate a negative correlation with score. Bottom: As can be seen there is even heterogeneity in treatment effects for several data characteristics.

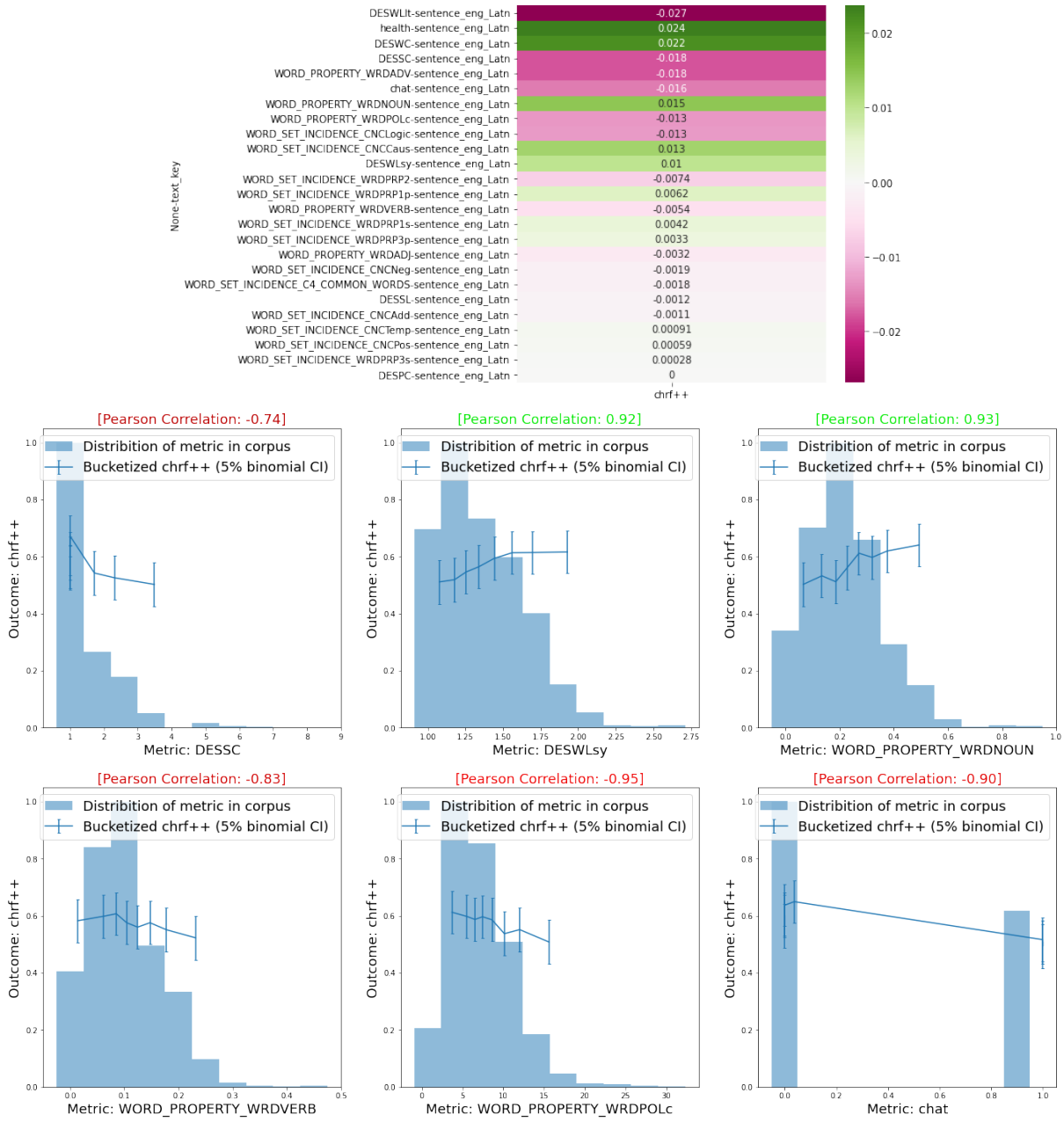


Figure 9: Top: A regression on the chrf++ score of a model pipeline using NLLB with segmentation. Positive values indicate improved score, lower values indicate a negative correlation with score. Bottom: As can be seen there is still some heterogeneity in treatment effects for several data characteristics but they have significantly flattened.

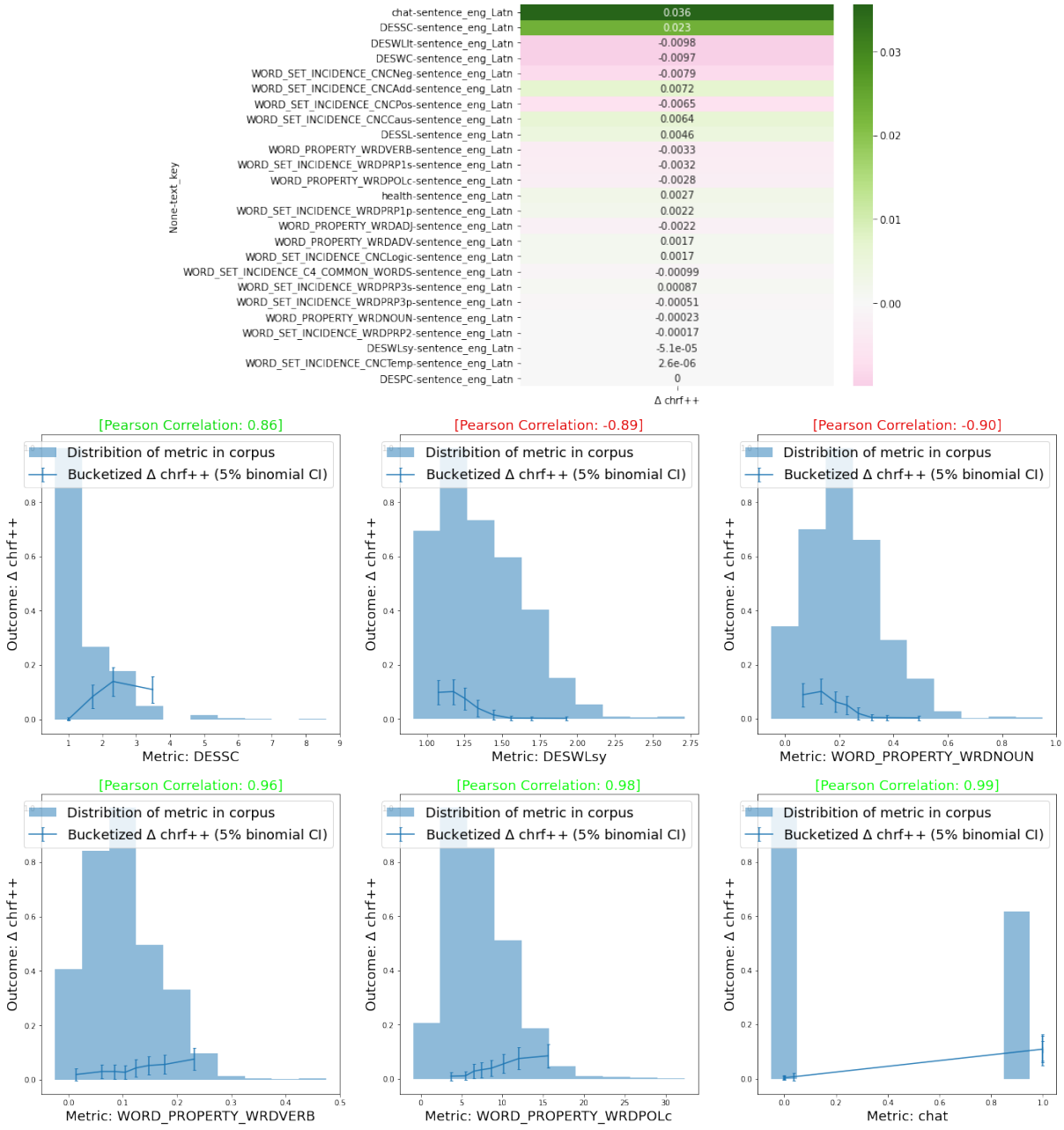


Figure 10: Top: A regression on the per-sentence treatment effect between a translation run through NLLB with and without sentence segmentation. Positive values indicate correlation with improved chrf++ from segmentation over the base model. Bottom: As can be seen there is even heterogeneity in treatment effects. Sentence splitting has the most positive effect correlation with the chat corpus and for the evaluation points with many sentences.

English	Yeah that would be so fun! It's really easy honestly, there's a bit of skill with steering but once you get the hang of it it feels super natural.
True	Да, было бы очень весело! Честно говоря, это очень просто, нужен небольшой навык руления, но как только вы поймете, ощущение будет очень естественным.
NLLB	Да, это было бы так весело! но как только ты научишься управлять, это будет очень естественно.
NLLB (seg)	Да, это было бы так весело! Это очень просто, честно говоря, требуется немного мастерства, но как только ты научишься управлять, это будет очень естественно.
English	That's right. How is your family? how many of you are there?
True	Точно. Как твоя семья? Сколько вас?
NLLB	Как ваша семья?
NLLB (seg)	- Да, это так. Как твоя семья? Сколько вас там?

Table 4: NLLB without proper segmentation misses entire portions of the context. For example, the red-highlighted portion is missing from the non-segmented NLLB text, but present elsewhere.