Separating the Wheat from the Chaff with BREAD: An open-source benchmark and metrics to detect redundancy in text

Isaac Caswell
Google Research
icaswell@google.com

Lisa Wang
Google DeepMind
wanglisa@google.com

Isabel Papadimitriou
Computer Science Department
Stanford University
isabelvp@stanford.edu

Abstract

Data quality is a problem that perpetually resurfaces throughout the field of NLP, regardless of task, domain, or architecture, and remains especially severe for lower-resource languages. A typical and insidious issue, affecting both training data and model output, is data that is repetitive and dominated by linguistically uninteresting boilerplate, such as price catalogs or computer-generated log files. Though this problem permeates many web-scraped corpora, there has yet to be a benchmark to test against, or a systematic study to find simple metrics that generalize across languages and agree with human judgements of data quality. In the present work, we create and release BREAD, a human-labeled benchmark on repetitive boilerplate vs. plausible linguistic content, spanning 360 languages. We release several baseline CRED (Character REDundancy) scores along with it, and evaluate their effectiveness on BREAD. We hope that the community will use this resource to develop better filtering methods, and that our reference implementations of CRED scores can become standard corpus evaluation tools, driving the development of cleaner language modeling corpora, especially in low-resource languages. [1]

1 Introduction

In this paper, we introduce a benchmark and propose a suite of metrics to help identify a common facet of low-quality data: repetitive boilerplate that is not reflective of natural linguistic content. Large language corpora scraped from the internet are becoming invaluable tools as self-supervised language modeling has gained prominence as a driving force of advancements in NLP (Devlin et al., 2018; Chowdhery et al., 2022; Brown et al., 2020, inter alia). In the case of many low-resource languages, noisy in-language data often makes up a significant proportion of any scraped corpus (Kreutzer et al., 2022). Very often, this noise is in the form of repetitive boilerplate: uninteresting data without linguistic diversity, such as a long list of similar products from an e-commerce website. Automatically reducing repetitive boilerplate in low-resource language corpora remains an important problem to extend NLP to the thousands of languages currently underserved by language technology.

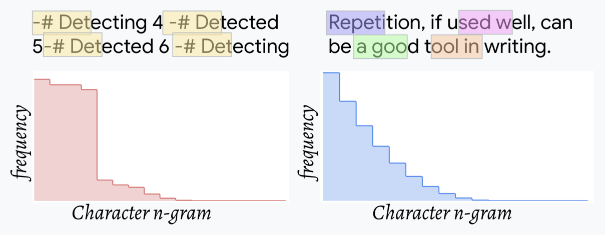


Figure 1: Character ngram based metrics compare the ngram frequency histogram between natural text and repetitive text, and assign a score of how repetitive it is. In this toy example, the character 6-gram histogram on the left is clearly distinguishable from the more natural distribution on the right. The CRED metrics rely on this intuition, applying simple metrics based on ngram frequency in order to detect repetitive boilerplate data in a language-agnostic manner.

To address the problem of redundant boilerplate, our contributions in the current work are two-fold:

1. We release BREAD (Boilerplate and Redundancy Evaluation on Assorted Documents), the first benchmark to measure redundancy and boilerplate in text;
2. We test and open-source CRED (Character REDundancy) scores, a suite of interpretable, fast, language-agnostic metrics for detecting repetition in documents.

Since data noise disproportionately affects low-resource languages, we only consider metrics that are language-agnostic (meaning their performance doesn't depend on any particular language). As

[1] Our data for the BREAD benchmark and code for the CRED scores suite is at https://github.com/toizzy/bread

Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 324–338
December 6, 2023 ©2023 Association for Computational Linguistics

such, we do not consider neural methods for the baselines released with BREAD: though they are more expressive than surface-level metrics, they rely on high-quality training data, and are therefore less reliable for low-resource languages where the training data is scarce, noisy, or highly overlapping with eval or model-training data. Similarly, neural metrics struggle with interpretability and reproducibility.

The difference between a paragraph of natural text and a long, repetitive list does not depend on the source language or the particular thing that is repeating. Therefore, it is possible to build language-agnostic metrics that ignore textual features entirely, and operate purely on the token-frequency distribution. Using this intuition, we explore three ngram-based metrics: **type-token ratio (TTR)**, measuring the percentage of unique ngrams; **ngram-moment**, measuring the peakiness of the frequency distribution; and **ngram-Zipfianness**, measuring the distance from the expected frequency distribution of natural language.

Our objective is to detect redundant language *within one document*. This is different from a commonly studied problem in data quality management, where redundancy refers to a dataset containing many redundant copies of similar natural documents. We open-source the BREAD benchmark and the CRED metrics, making it a replicable resource for the community.

2 Related Work

In the field of Data-Quality management, quality scores are used for *measurement* and *improvement*, and often incorporated into an iterative process (Wang, 1998). For NLP, there are many existing works highlighting the importance of cleaning data for training neural models (Khayrallah and Koehn, 2018; Junczys-Dowmunt, 2018a; Wang et al., 2018b). Many denoising approaches rely on classifiers (Chen and Huang, 2016; Chen et al., 2016; Wang et al., 2017) or cross-entropy distance between models (Moore and Lewis, 2010; Axelrod et al., 2011; van der Wees et al., 2017; Axelrod, 2017), an approach often applied to data weighting and curriculum training (Zhang et al., 2017; Wang et al., 2018a, 2019). There are neural diversity metrics, like Miranda et al. (2022), which uses the cosine distance between Task2Vec embeddings (Achille et al., 2019), and has been applied to measure LLM output diversity (Lee et al., 2023).

Although data noise has always been a recognized problem, it has become a more pressing issue in recent years, as models have become more and more expressive, therefore also more capable of memorizing noise. Statistical machine translation models were more robust to data noise and tended only to benefit from bigger data (Goutte et al., 2012) (with a few exceptions, like Taghipour et al. (2011)), and works on data filtering were usually focused on improving training efficiency (for instance, Johnson et al. (2007)). Despite their generally higher performance, neural models tended to be much more sensitive to data noise (Khayrallah and Koehn, 2018), possibly as a result of being able to memorize statistical outliers (Arpit et al., 2017; Feldman and Zhang, 2020). Even early versions of Paracrawl damaged MT performance (Junczys-Dowmunt, 2018b; Schamper et al., 2018), and the winners of the yearly WMT campaign tend to rely heavily on data filtering (Junczys-Dowmunt, 2018a; Chaudhary et al., 2019; Lu et al., 2020; Lo and Joanis, 2020). As a result, there have been several data filtering shared tasks in WMT (Koehn et al., 2018, 2019, 2020), and open-sourcing of various iterations of data cleaner BICLEANER (Esplà-Gomis et al., 2020; Ramírez-Sánchez et al., 2020; Zaragoza-Bernabeu et al., 2022), which use a variety of approaches, including bilingual dictionaries, random forests, and neural models.

While neural metrics or complex ensembles like BICLEANER are often effective, they 1) are harder to interpret; 2) may filter on artifacts like domain, rather than quality; 3) will tend only to work for languages they have explicitly been trained on; and 4) cannot be replicated between works unless a public implementation is released. For this reason, the baseline metrics released with BREAD are simple, interpretable, surface-level metrics, that work independent of language and domain.

A token-based metric to measure the diversity and redundancy of token ngrams *between* documents in a corpus (rather than within segments of one document) is SELF-BLEU (Zhu et al., 2018), which is based on the widely used BLEU score (Papineni et al., 2002). On a more granular level of character ngrams, the CHRF (Popović, 2015) and CHRF++ (Popović, 2017) metrics measure similarity between documents, correlating better with human judgement than token-level metrics like BLEU, especially for low-resource and highly-inflecting languages (Kocmi et al., 2021; Freitag et al., 2022;

Bapna et al., 2022; Caswell et al., 2020). We follow this intuition and use character-ngram metrics. The frequency moment score defined in the present work is similar to segment-level CHRF applied with itself as its own reference.

Perhaps the most similar approaches to those in the present work come from a separate field, namely detecting redundancy and diversity in relational or tabular data (Ehrlinger and Wöß, 2022). Batista and Salgado (2007) and Ehrlinger and Wöß (2019) define interpretable minimality scores to measure redundancy at a schema-level for tabular data, based on cluster density, which is equivalent to the TTR in the present work.

3 BREAD: Dataset and Annotation

We release BREAD (Boilerplate and Redundancy Evaluation on Assorted Documents), an expert-annotated dataset spanning 360 languages, to tune and benchmark methods for filtering repetitive boilerplate. BREAD consists of randomly-chosen documents from the multilingual, common-crawl-based MADLAD-400 dataset (Kudugunta et al., 2023), which are then annotated by expert NLP-practitioner annotators.

Our annotation schema consists of two high-confidence classes and two low-confidence classes. The high-confidence classes are 1) REP, repetitive boilerplate (N=449), and 2) OK, natural text (N=863). To keep the examples in REP and OK high-confidence, we also use two low-confidence codes: BOIL, for documents that are clearly non-linguistic boilerplate or noise, but are not necessarily repeating (N=499); and UNK for where the annotator was not sure (N=3339). Documents labeled as UNK were discarded. See Appendix Table 2 for examples of each class. The examples labeled OK cover 360 languages, with no individual language having more than 6 samples; the language distribution of the other three codes are harder to measure, since they are often nonlinguistic content or noisy ambiguous text. Examples are capped at 5000 character for ease of processing.

BREAD is split into a tune and a test set, each with 1000 documents. We propose two benchmarks, scored with F1 on the following binary prediction problems:

1. **BREAD-REPEAT**: positive class is OK; negative is REP.
2. **BREAD-NOISY**: positive class is OK; negative is union of REP and BOIL.

4 Methods

We explore three well-studied, straightforward metrics based on ngram frequency distributions and evaluate their effectiveness in the domain of measuring repetitive boilerplate. We explore both character ngrams and token ngrams, as well as combinations of the two. As with BLEU (Papineni et al., 2002), we consider using multiple n-gram lengths at once, and combining these scores by averaging them. By construction, all metrics assign a higher score to noisier text.

The input to all our metrics is the smoothed frequency distribution of ngrams within a document. Distributions of ngrams tend to be noisier for shorter texts, so we apply Laplace smoothing with parameter λ, and clip the distribution with ϵ-thresholding (keeping only ngrams with probability over some ϵ (Freitag et al., 2023)). Let $f_n^{(i)}$ be the raw frequency of the ith most common n-gram. We define our smoothed frequency distribution as:

$$\tilde{f}_n^{(i)} \propto \left(f_n^{(i)} + \lambda\right) \mathbb{1}\{f_n^{(i)} > \epsilon\} \qquad (1)$$

This said, the authors would like to foreshadow that this distribution clipping and smoothing end up not being very important parameters for well-performing metrics, so the reader may safely ignore this and imagine that the metrics are a function of raw frequency.

The metrics we explored are as follows:

TTR: As an intuitive and well-known baseline metric, we use the Token-Type Ratio (TTR) (Templin, 1957), which is the fraction of unique tokens in a document (types) over the total number of tokens. We use n-grams as tokens.

Frequency Moment score: The second score we consider is the generalized moment of the frequency probability distribution, the sum of all frequencies when applying a with a nonlinearity $g(x)$. For a character n-gram with length n, the momet score is defined as:

$$m_n = \sum_i g\left(\tilde{f}_n^{(i)}\right) \qquad (2)$$

The nonlinearity $g(x)$ is a parameter we can vary to best fit out benchmark. Intuitively, setting any superlinear $g(x)$, this metric measures redundancy, or peakiness, of the ngram counts, as the score is larger when there is more weight in the head of

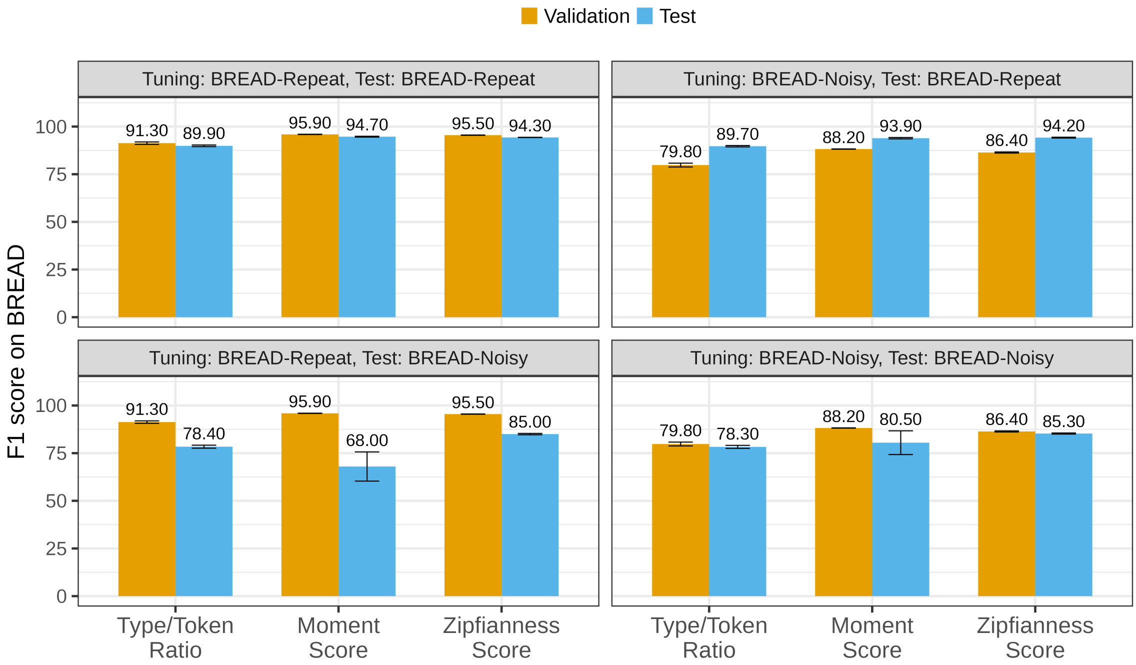


Figure 2: F1 scores for the three metrics proposed in this work, for all four combinations of tuning and testing on BREAD-REPEAT and BREAD-NOISY. The reported values are the average of the top ten parameter settings on the tuning set. Error bars represent 95% confidence intervals.

the distribution. When $g(x) = x^k$, the score corresponds to the kth moment of the distribution; when $g(x) = -x\log(x)$, it corresponds to the entropy.

Zipfianness: Human languages have a largely consistent word distribution: across languages, the empirical frequency of how often different words appear follows a Zipfian, or power-law distribution, where the word in frequency rank r has frequency roughly proportional to $\frac{1}{r}$ (Zipf, 1936; Piantadosi, 2014). For example, in English the most common word "the" occurs around double the rate of the second most common word "of". To test whether a document is distributed like natural text, we can check whether its ngram distribution matches the empirical ngram distribution of a human language, which we estimate as a function of the n-gram length with a slight modification of the classic $\frac{1}{r}$ value (details in Appendix D). [2] Therefore, we define the Zipfianness score as follows:

$$z_n = \sum_i d\left(\hat{f}_n^{(i)}, \tilde{f}_n^{(i)}\right) \qquad (3)$$

Where $\hat{f}_i$ is the estimated frequency of the ith most common token, and $d(x, y)$ is a distance metric. For $d(x, y)$, we consider $|x-y|^2, \log(|x-y|), \log^2(|x-y|)$ and $\mathrm{JSD}(x, y)$. We initially also considered KL divergence (in both directions) and absolute distance, but they proved less effective.

4.1 Compensating for Length Dependency

All three of these scores are dependent on the length of the document and are all minimized when the document consists only of unique n-grams (i.e. input distribution is uniform). Therefore, we normalize the score on a candidate document by what the score would be for a document of the same length with only unique n-grams. This leads to the interpretation of something like "How much more redundant is this document than a natural document of the same length?". However, since natural languages are drawn from a finite and non-uniform set of symbols, the uniform distribution becomes an increasingly bad approximation of a "natural" document as the document length increases, and leads to the reverse skew of what the length normalization was originally trying to address. To compensate for this, we introduce a simple asymptote for the number of tokens in a document, and normalize by the uniform distribution for a document with that length. This approach is chosen over the more typical approach of a fixed-width sliding window over characters, as is often done

[2] We also experiment with the empirical token distribution from a random sample of 10,000 English documents from MADLAD-400-clean, and find the results to be the same (but much more painful to calculate), so for simplicity we focus only on the analytic approximation in this paper.

with TTR (Kettunen, 2014), because BREAD has a significant range in document lengths, so we expect this approach to capture the variation in scores more cleanly. Details are in Appendix Section A.

4.2 Grid Search

Each metric is dependent on the parameters used to smooth and nonlinearize the frequency distribution, the length normalization asymptote, and the appropriate threshold when used as a classifier. Therefore, we split the dataset into a 50/50 validation/test split, and perform a grid search on the validation split, optimizing for F1 score. Variants of the scores optimized for different metrics are also open-sourced (§6); details in Appendix B.

5 Baseline Metric Results on BREAD

As shown in Fig. 2, all metrics have fairly good correlation with human judgement, even when they are trained on the out-of-domain split of BREAD (the off-diagonal entries). For detecting repetition alone (BREAD-REPEAT; top row), both the moment score and the Zipfianness score performed about 5% better on both tuning and test sets than TTR. When detecting both noise and boilerplate (BREAD-NOISY), the difference in scores is more pronounced, with Zipfianness outperforming TTR by 9% on the test split. The moment score, which like TTR is only able to detect redundancy but not other types of noise, barely outperforms TTR.

It is worth noting that for questions of data noise, there is a large difference between apparently close scores, if they are both close to 100. Caswell et al. (2020) note (§*Massive Class Imbalances: 99% Accuracy Is Not Enough*) that if a Language Identification model has a precision of 99.0, using it to generate a dataset for a typical low-resource language will yeild a dataset with just under *a tenth of a percent* of sentences in the target language. Increasing this precision to 99.9%, though under 1% better in additive terms, is a 10x improvement in dataset precision. Keeping this in mind, we see that although we have a ways to go with better data quality scores, the improvement in noise detection from 78 F1 to 85 F1 is quite substantial!

For a qualitative understanding of what scores on BREAD look like, one can refer to Figure 3, which shows the moment score as a function of length, along with the decision boundary. Details of the best hyperparameters per ngram length are given in Appendix Table 5.

5.1 Which Parameters Worked the Best?

Unsurprisingly, the most important parameter was the choice of n-gram length(s). Our initial grid search went over a deep grid of different values. However, since many of these factors ended up not being very important, they led to overfitting and poor test scores. Therefore, for the final values, we re-ran the grid search with a very limited set of parameters (§B). Findings from both rounds are summarized here:

- **n-grams:** For the purely repetition-based metrics (TTR, Moment), the most effective n-gram length seemed to be anything of length 6-grams and up. For Zipfianness, the peak was considerably earlier, at 4-grams and 5-grams. The best single n-gram value for across all approaches would therefore be a 5-gram or 6-gram, similar to the finding by Popović (2015) that 6-grams corresponded the best with human-judged quality for CHRF. Ensembles of different types of n-grams usually achieved slightly higher quality, but the improvements were minor.
- **Smoothing:** There was no obvious pattern to the best smoothing value λ.
- **Distribution truncation:** The optimal ϵ value for ϵ-clipping was almost always 0, and the optimal k for top-k clipping was almost always ∞. We conclude that using the full distribution is generally optimal, and omitted distribution truncation in the final grid search.
- **Nonlinearity:** The best nonlinearity for the moment score tended to be x^2, corresponding nicely with the variance, though the squared entropy $(x \log(x))^2$, $x^{1.5}$, and x^3 also frequently came out on top for different settings of the other parameters. The best distance function for Zipfianness was generally the squared distance, though $\log(|x - y|)$ also performed well.
- **lengthnorm asymptote:** The best asymptote for the document length (used when normalizing by length; §4.1) was usually 2000.

5.2 CRED as a Metric for Data Quality

To validate these metrics on existing datasets and to demonstrate how they can be used to assess data quality, we report their average scores on the MADLAD-400 dataset. This resource is an excellent testing ground because it has both *clean* and *noisy* splits, and furthermore covers many very low-

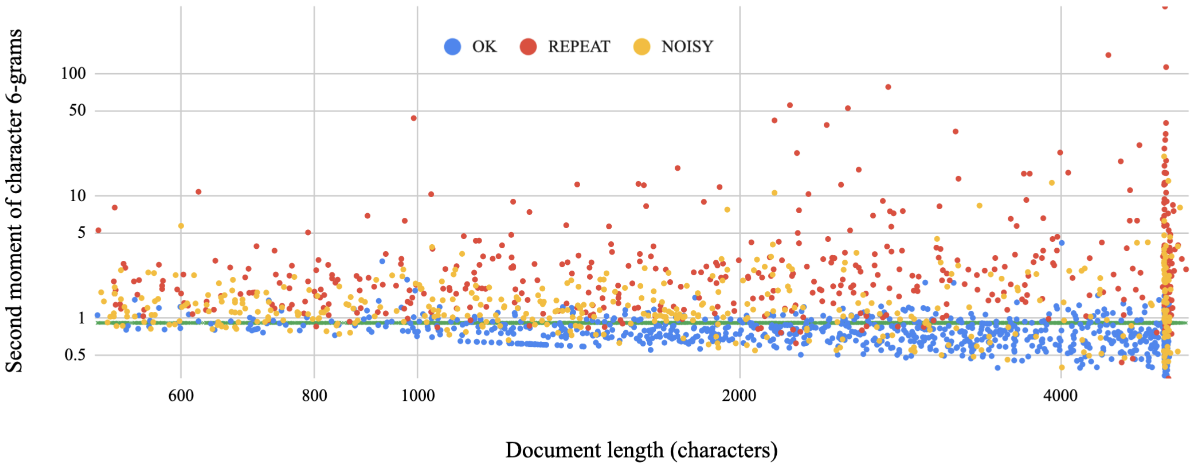


Figure 3: Moment scores on BREAD as a function of document length, with the learned decision boundary in green, demonstrating how moment scores effectively separate noisy data from clean data along the y-axis. Each point represents a document in BREAD, with the OK labels in blue, REP labels in red, and BOIL labels in orange. The cluster at the right reflects the truncation of BREAD documents at 5000 characters.

	TTR	moment	zipf.
MAD. CLEAN HRL	0.116	0.677	0.679
MAD. CLEAN LRL	0.175	0.972	0.688
MAD. NOISY HRL	0.136	0.802	1.064
MAD. NOISY LRL	0.189	1.473	2.063

Table 1: Scores on the **noisy** and **clean** splits of MADLAD-400, for 45 high-resource languages (HRL, >1M documents in the **clean** split) and 368 low-resource languages (LRL). All scores show more severe noise for low-resource languages, and for the *noisy* split.

resource languages, where we expect more noise in the data. Results are reported in Table 1. We make the following observations:

1. All three metrics agree that the *noisy* split indeed has more repetitive content. This offers more evidence that our metrics are effective at detecting noise and assessing data quality.
2. For low-resource languages (LRL), all metrics indicate that both clean and noisy splits are noisier compared to the respective splits in high-resource languages (HRL), which would align with intuition.
3. The relative scores also allow us to make the interesting inference that the *clean* split of the low-resource languages has a similar noisiness level to the *noisy* split of the high-resource languages.

6 Open-Sourcing

We open-source reference implementations of these metrics. Following the example of SACREBLEU (Post, 2018), each score has a unique signature reporting all relevant hyperparameters, so it is fully reproducible. In order to suit different levels of noise and different preferences of precision versus recall, we release versions of each classifier that have been tuned for F1 on a balanced version of BREAD, as well as a version that has been tuned on the P4 score (Sitarz, 2022) with BREAD upweighted so it is 75% clean data.

7 Conclusion

Data quality is an evergreen problem, and as NLP is widening to a growing set of low-resource languages, where noise is a more severe problem, the need for more interpretable metrics to asses noise becomes especially prominent. Recent approaches to highly multilingual technologies like NMT and LangID have reported severe noise issues for low-resource languages (Caswell et al., 2020; Bapna et al., 2022), and many publicly available datasets with low-resource languages in fact contain no in-language content (Kreutzer et al., 2022). Nonetheless, there was heretofore no public benchmark for boilerplate and noise detection.The present work introduces BREAD, a multilingual, expert-annotated benchmark for detecting noise. It also investigates several interpretable, language-agnostic baseline metrics based on character ngram frequency distributions, as well as their scores on the public dataset MADLAD-400. Finally, it open-sources reference implementations of several language-agnostic metrics for scoring and classifying data.

Limitations

While the BREAD and the metrics introduced in this paper are useful approximations, there are many forms of noise they can't detect. They can't detect poor grammar, scrambled text, translationese, toxicity, or other noise that follows a Zipfian-distribution. Furthermore they can't detect inter-example redundancy, for which a better-suited metric would be something like SELF-BLEU.

Furthermore, such a metric may not generalize well to all languages. Although the language-agnostic approach to the creation of the BREAD eval set is constructed to work for all languages, many languages, especially those with more distinct character sets like Chinese and Japanese, may exhibit unique forms of noise or token distributions.

Finally, these metrics will tend to be less useful for shorter texts, and practitioners are cautioned against using them on sentence-level data.

Ethics Statement

We introduce a benchmark dataset and scoring mechanisms for improving the quality of low-resource language corpora. Like any metrics based on surface-level features, our metrics are coarse and do not reflect the subtleties of different languages. We propose for our CRED scores to be used in a battery of data quality evaluation methods.

Acknowledgements

The authors would like to thank the wonderful Ithaka Co-Operative House, which enabled this collaboration. This research was funded in part by NSF award number IIS-2128145.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. *CoRR*, abs/1902.03545.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Amittai Axelrod. 2017. Cynical selection of language model training data. *CoRR*, abs/1709.02279.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, Theresa Breiner, Vera Axelrod, Jason Riesa, Yuan Cao, Mia Xu Chen, Klaus Macherey, Maxim Krikun, Pidong Wang, Alexander Gutkin, Apurva Shah, Yanping Huang, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2022. Building Machine Translation Systems for the Next Thousand Languages. *arXiv e-prints*, page arXiv:2205.03983.

Maria da Conceição Moraes Batista and Ana Carolina Salgado. 2007. Information quality measurement in data integration schemas. In *QDB*, pages 61–72.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Boxing Chen and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 314–323.

Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. 2016. Bilingual methods for adaptive training data selection for machine translation. In *AMTA*.

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 preprint arXiv:2204.02311*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lisa Ehrlinger and Wolfram Wöß. 2019. A novel data quality metric for minimality. In *Data Quality and Trust in Big Data: 5th International Workshop, QUAT 2018, Held in Conjunction with WISE 2018, Dubai, UAE, November 12–15, 2018, Revised Selected Papers 5*, pages 1–15. Springer.

Lisa Ehrlinger and Wolfram Wöß. 2022. A survey of data quality measurement and monitoring tools. *Frontiers in big data*, page 28.

Miquel Esplà-Gomis, Víctor M. Sánchez-Cartagena, Jaume Zaragoza-Bernabeu, and Felipe Sánchez-Martínez. 2020. Bicleaner at WMT 2020: Universitat d'alacant-prompsit's submission to the parallel corpus filtering shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 952–958, Online. Association for Computational Linguistics.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems*, volume 33, pages 2881–2891. Curran Associates, Inc.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum bayes risk decoding for machine translation.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André FT Martins. 2022. Results of wmt22 metrics shared task: Stop using bleu–neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68.

Cyril Goutte, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas: Research Papers*, San Diego, California, USA. Association for Machine Translation in the Americas.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018a. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018b. Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 425–430, Belgium, Brussels. Association for Computational Linguistics.

Kimmo Kettunen. 2014. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics*, 21(3):223–245.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta

Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, et al. 2023. MADLAD-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*.

Alycia Lee, Brando Miranda, and Sanmi Koyejo. 2023. Beyond scale: the diversity coefficient as a data quality metric demonstrates llms are pre-trained on formally diverse data.

Chi-kiu Lo and Eric Joanis. 2020. Improving parallel data identification using iteratively refined sentence alignments and bilingual mappings of pre-trained language models. In *Proceedings of the Fifth Conference on Machine Translation*, pages 972–978, Online. Association for Computational Linguistics.

Jun Lu, Xin Ge, Yangbin Shi, and Yuqi Zhang. 2020. Alibaba submission to the WMT20 parallel corpus filtering task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 979–984, Online. Association for Computational Linguistics.

Brando Miranda, Patrick Yu, Yu-Xiong Wang, and Sanmi Koyejo. 2022. The curse of low task diversity: On the failure of transfer learning to outperform maml and their empirical equivalence.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference*, pages 220–224.

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.

Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21:1112–1130.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Julian Schamper, Jan Rosendahl, Parnia Bahar, Yunsu Kim, Arne Nix, and Hermann Ney. 2018. The RWTH Aachen University supervised machine translation systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 496–503, Belgium, Brussels. Association for Computational Linguistics.

Mikolaj Sitarz. 2022. Extending f1 metric, probabilistic approach.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. In *Proceedings of Machine Translation Summit XIII: Papers*, Xiamen, China.

Mildred C Templin. 1957. Certain language skills in children; their development and interrelationships.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine transaltion. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410.

Richard Y Wang. 1998. A product perspective on total data quality management. *Communications of the ACM*, 41(2):58–65.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488. Association for Computational Linguistics.

Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2018a. Dynamic sentence sampling for efficient training of neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 298–304, Melbourne, Australia. Association for Computational Linguistics.

Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019. Dynamically composing domain-data selection with clean-data selection by "co-curricular learning" for neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1282–1292, Florence, Italy. Association for Computational Linguistics.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018b. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third*

Conference on Machine Translation: Research Papers, pages 133–143. Association for Computational Linguistics.

Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz Rojas. 2022. Bicleaner AI: Bicleaner goes neural. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 824–831, Marseille, France. European Language Resources Association.

Dakun Zhang, Jungi Kim, Josep Crego, and Jean Senellart. 2017. Boosting neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 271–276, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

George Kingsley Zipf. 1936. *The Psychobiology Of Language*. Routledge.

A Length normalization details

As mentioned in Section 4.1, these simple metrics have a dependency on the length of the document, which is undesirable. Therefore, we normalize them by dividing by their minimum possible value for a document of that length, which is achieved on the uniform distribution. (The maximizing value, achieved by the one-hot distribution, grows very quickly only seemed to add noise.)

A.1 Moment

The distribution of moment scores on a sample of filtered, web-mined text across a variety of languages[3] can be seen in Figure 4a. There is a clear lower bound on this distribution, which corresponds to a uniform token distribution. In Figure 4b, the distribution is plotted alongside the score on the uniform distribution (in red) and in Figure 4c, the moment scores are shown when normalized by the uniform distribution. It is clear that this is a poor fit for longer documents, where the uniform distribution is more unlikely, and indeed (in the case of a finite alphabet) impossible. Therefore, we introduce an asymptote on the length of the document. For a document with true length n, we instead calculate the uniform distribution moment on a scaled length $\tilde{n} = \frac{n*\alpha}{n+\alpha}$, for some asymptote α. The yellow line in Figure 4b shows the uniform distribution on $\tilde{n}$ with an asymptote of $\alpha = 5000$, and Figure 4d demonstrates that after dividing by this, the length dependency, at least when it comes to the lower bound, has nicely flattened out.

A.2 Zipfianness

We normalize the Zipfianness score in the same way as the moment score, namely by the score on the uniform distribution, with some asymptote parameter α.

B Grid Search

Each metric is dependent on the parameters used to smooth and nonlinearize the frequency distribution (Section 4). Furthermore, in order to use such a metric as a classifier for whether text is noisy or not, an appropriate threshold is needed as the decision boundary. Therefore, for each metric, we carry out a grid search over its possible hyperparameters. We split the BREAD dataset into a 50/50 tune/test split, and perform the grid search on the tune split. [4] The hyperparameter ranges initially explored were as follows:

Grid Search 1:

- **ngrams:** we explore every contiguous combination of ngrams from character 2-grams to character 10-grams. We also explore token 1-grams and 2-grams, and combinations of token 2-grams with character 5- and 6-grams, as in CHRF++.
- ϵ **values**: we cover the range of [0, 0.01]
- k **values**: we cover the range of [2, 1024], as well as no top-k filtering
- **smoothing**: we cover the range of [0, 2].
- **nonlinearities**: These vary by method and are described along with each method.

However, given the small size of the tuning metric, this led to severe overfitting. Based on analysis of which parameters were or were not very important, we re-did the final, simpler grid search:

Grid Search 2 (constrained):

- **ngrams:** we explored only sets of one to two ngram values at once, for instance a mixture of 4-grams and 5-grams, but not larger sets like in the first gridsearch. For multiple-ngram settings we looked at contiguous lengths as well as skip-2 lengths. We explored character 1-grams to character 10-grams.
- ϵ **values**: we did not do epsilon truncation.
- k **values**: we did not perform top-k filtering
- **smoothing**: we only explored 0 and 1.
- **nonlinearities**: We limited ourselves to $x^{1.5}, x^2, x^3$ for the moment, and $|x - y|^2, \log(|x-y|), \log^2(|x-y|)$, JSD(x, y) for Zipfianness.

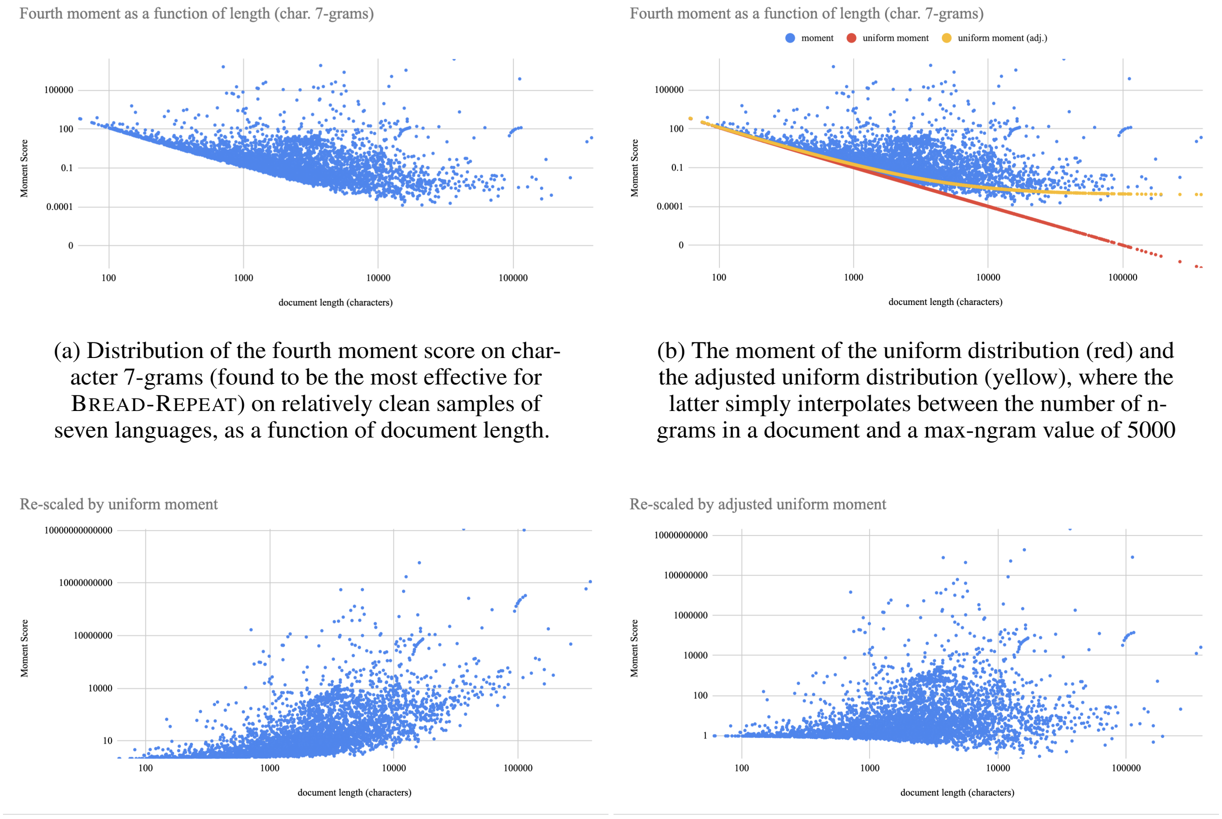
We optimized the grid search with the F1 metric. The choice of the optimization metric is inherently dependent on the data balance, and a one-size-fits-all solution is not possible; as such, though this is the metric explored in this paper, variants of the scores optimized for different metrics are open-sourced (See Section 6).

C Dataset classes and examples

Several examples of documents annotated with different classes from the BREAD dataset are given in Table 2.

[3] Equal mix of Arabic, English, Finnish, German, Russian, Swahili, and Turkish

[4] A train split per se is not necessary, as we are not training any models.



(a) Distribution of the fourth moment score on character 7-grams (found to be the most effective for BREAD-REPEAT) on relatively clean samples of seven languages, as a function of document length.

(b) The moment of the uniform distribution (red) and the adjusted uniform distribution (yellow), where the latter simply interpolates between the number of n-grams in a document and a max-ngram value of 5000

(c) Moment score normalized by the uniform moment. It is apparent that the score is still length-dependent.

(d) Moment score normalized by the adjusted uniform moment. The length dependency is much less.

Figure 4: Length normalization for the moment score

Class	Description	Example
OK	natural text	Alokba den Sangremer Sensaksem | Tir Yimyim By nungsang on November 28, 2017 Comments Off on Alokba den Sangremer Sensaksem Sangremer : Alokba, nenok ashiakang kijong tepenjem kibong, Okolai nabo tulura ta meteta lir, saka nü kinungtsü indangang junga memetet. La kechi inyaker-aka? Tenünga shiba aka? Alokba : Oko, Labo mapangshia polashia tzüwa awaba dak alaka kecha balaka meinyakerako, la tenünga Tzüwala, süra ner tantsüa kechi inyaker?
		Kipirde basylan türkmen migrantlar Türk polisiýasy we migrant. Kipriň demirgazyk böleginiň metbugatynda soňky wagtlarda türkmen migrantlary barada köp maglumat çykyp başlady. Diňe soňky birnäçe günüň dowamynda ol ýerde birnäçe türkmen zähmet migrantlarynyň ogurlykda aýyplanyp, suda çekilip, soňra-da wagtlaýynça tussag edilendigi habar berilýär. Belli bolşy ýaly Türkmenistan garaşsyzlygyny alandan soň Türkiýe türkmen zähmet migrantlarynyň esasy ýykgyn edýän ýurtlarynyň birine öwrüldi. Türkiýedäki türkmen zähmet migrantlary barasynda türk metbugatynda yzygiderli maglumatlar çap edilýär. Ýöne indi Türkiýeden Kipriň demirgazyk bölegine gidip işleýän türkmenistanly zähmet migrantlary barada hem metbugatda çap edilýän maglumatlar köpelýär.
REP	repetitive boilerplate	Shabir May 13, 2019 at 8:24 PM Shabir May 13, 2019 at 8:27 PM Shabir May 13, 2019 at 8:28 PM Do visit the site Eduassam jobs in Assam tridip May 31, 2019 at 8:24 PM golam June 12, 2019 at 10:48 PM
		3.6 miles 18° 2020-01-13 12:16:54 3.7 miles 181° 2020-01-18 14:04:11 3.7 miles 181° 2020-01-19 19:29:48 3.8 miles 235° 2020-01-20 19:43:23 Stations qui ont entendu WA1PLE-13 directement par radio – 2020-012019-122019-112019-102019-092019-082019-07 1 2020-01-14 03:19:07 2020-01-14 03:19:07 FN42JD > FN31ST 67.3 miles 250° 2020-01-14 03:19:07 54 2020-01-09 00:45:56 2020-01-19 06:52:58 FN42JD > FN42BF 32.4 miles 282° 2020-01-19 06:52:58 1 2020-01-15 01:20:00 2020-01-15 01:20:00 FN42JD > FN33TA 84.8 miles 317° 2020-01-15 01:20:00
BOIL	boilerplate but not repeating	jasa service rolling door murah: jasa service kunci rolling murah jakarta selatan,utara,pusat,slipi,sunter, tangerang. jasa service kunci rolling murah jakarta selatan,utara,pusat,slipi,sunter, tangerang. Diposting oleh ardicom di 18.53
		E5500/6500 68" Cabinet 4U Rack Mount Kit – Sun Parts from AnySystem.com. X9674A 595-5540 For pricing and availability, please call 201-445-3122 or email sales@anysystem.com . AnySystem - Home / X9674A 595-5540 E5500/6500 68" Cabinet 4U Rack Mount Kit – Sun Parts E5500/6500 68" Cabinet 4U Rack Mount Kit – Sun Parts from AnySystem.com.

Table 2: BREAD Dataset classes and corresponding examples. Note that some examples are excerpts from longer documents.

D Zipf Approximation via Random Gradient Descent

We initially calculated the empirical Zipf distribution from a linguistically diverse set of data. However, this was cumbersome to deal with, since we needed a value for every n-gram length and for every n-gram index, leading to a 20x10000 table. Although the approximation of $f_r \propto \frac{1}{r^b}$, for the 1-indexed rank of a token r and some exponent b, is an ok approximation, it is known to be fairly poor near the edges of the distribution. Therefore, we used the following algorithm to determine a better approximation, which we call Random Gradient Descent (RGD). The basic approach is to perturb a point randomly until the loss function improves, and then follow that direction in the parameter space until the loss stops decreasing, and alternate doing these two steps until convergence. In pseudocode, this algorithm looks like this:

```
def rgd(initial_args, loss_fn,
lr=0.01,
branch_n=10,
max_steps=10000,
max_attempts=10):
  total_steps = 0
  best_args = initial_args.copy()
  n_failed = 0
  cur_loss = loss_fn(best_args)
  initial_loss = cur_loss
  it = 0
  while True:
    it += 1
    if total_steps >= max_steps: break
    total_steps += branch_n
    branch, branch_grad, branch_loss =
    get_best_branch(best_args, loss_fn,
    lr, branch_n)
    if branch is None:
      # This means that no branch
      # improved on the best args.
      # As a result, there is no
      # gradient to follow.
      n_failed += 1
      if max_attempts
        and n_failed >= max_attempts:
        break
      continue
    cur_loss = branch_loss
    n_failed = 0
    best_args, follow_steps, follow_loss =
    follow_grad(branch, branch_grad,
    loss_fn)
    total_steps += follow_steps
    cur_loss = follow_loss
  return best_args, cur_loss, total_steps
```

```
def get_best_branch(args, loss_fn,
                    lr, branch_n):
  """Look at branch_n random points
  around args. Return the one with
  the lowest loss, and if none of them
  decreases the loss, return None's.
  """
  cur_loss = loss_fn(args)
  pool_args = [(args, lr, loss_fn)
  for _ in range(branch_n)]

  with Pool() as p:
    result =
    p.map(eval_branch, pool_args)
  branches, losses, grads = zip(*result)
  best_loss = min(losses)
  if best_loss >= cur_loss:
    return None, None, None
  i = losses.index(best_loss)
  return branches[i], grads[i], best_loss

def follow_grad(args, grad, loss_fn,
max_flat=20):
  """Follow the gradient grad
  until the loss stops improving.
  Guaranteed never to make the
  loss worse; might not change it.
  """
  cur_loss = loss_fn(args)
  initial_loss = cur_loss
  best_args = args.copy()
  n_flat = 0
  total_steps = 0
  while True:
    new_args = take_step(best_args, grad)
    new_loss = loss_fn(new_args)
    if new_loss > cur_loss: break
    elif new_loss == cur_loss:
      n_flat += 1
      if n_flat >= max_flat:
        break
    elif new_loss < cur_loss:
      total_steps += 1 + n_flat
      n_flat = 0
      best_args = new_args.copy()
      cur_loss = new_loss
  return best_args, total_steps, cur_loss
```

The literature is certainly rich with better and subtler ways to find a good approximation, but this method yielded an approximation that performed as well as the empirical Zipf distribution with our methods. The approximation we found with this method, and which we used in the main paper, is as follows, for the rth most common character n-grams of length n :

$$b = 6.809 * (r + 2.768)^{-1.487} + 0.527$$

$$s = 0.107 * (n + 12.0147)^{-12.654} + 0.0139$$

$$f_r^n = s\frac{1}{r^b}$$

score	n	Tune	Test	α	nl	λ
TTR	1	82.6	82.0	NA	NA	NA
TTR	2	82.8	82.4	NA	NA	NA
TTR	3	83.9	83.4	NA	NA	NA
TTR	4	86.3	85.1	NA	NA	NA
TTR	5	87.9	87.5	NA	NA	NA
TTR	6	89.4	89.2	NA	NA	NA
TTR	7	90.8	90.2	NA	NA	NA
TTR	8	91.8	90.4	NA	NA	NA
TTR	9	92.4	90.7	NA	NA	NA
TTR	10	92.5	90.1	NA	NA	NA
mmt.	1	83.3	82.9	∞	$x^{1.5}$	1
mmt.	2	84.4	84.6	2k	$x^{1.5}$	0
mmt.	3	87.4	88.0	2k	$x^{1.5}$	0
mmt.	4	92.3	91.7	2k	$x^{1.5}$	1
mmt.	5	95.2	93.7	2k	$x^{1.5}$	0
mmt.	6	95.8	94.9	2k	$x^{1.5}$	0
mmt.	7	95.8	94.3	2k	x^3	1
mmt.	8	95.8	94.6	2k	x^3	1
mmt.	9	95.4	94.7	2k	x^3	1
mmt.	10	95.1	94.5	5k	x^2	0
Zipf	1	83.1	82.2	2k	$\log(x)$	1
Zipf	2	84.7	84.0	∞	$\log(x)$	0
Zipf	3	90.1	89.7	2k	JSD	1
Zipf	4	94.7	93.7	2k	x^2	0
Zipf	5	95.5	94.3	5k	x^2	0
Zipf	6	94.5	93.4	∞	x^2	0
Zipf	7	93.5	92.8	2k	x^2	0
Zipf	8	92.7	92.3	2k	x^2	0
Zipf	9	91.8	91.3	2k	x^2	0
Zipf	10	91.2	90.7	2k	x^2	0

Table 3: Eval on BREAD-REPEAT

score	n	Tune	Test	α	nl	λ
TTR	1	70.7	70.3	NA	NA	NA
TTR	2	70.8	70.5	NA	NA	NA
TTR	3	71.6	71.3	NA	NA	NA
TTR	4	73.6	72.8	NA	NA	NA
TTR	5	75.2	74.8	NA	NA	NA
TTR	6	77.3	76.7	NA	NA	NA
TTR	7	79.0	77.7	NA	NA	NA
TTR	8	80.6	79.1	NA	NA	NA
TTR	9	81.6	79.7	NA	NA	NA
TTR	10	81.9	79.6	NA	NA	NA
mmt.	1	71.3	49.5	∞	$x^{1.5}$	1
mmt.	2	72.7	56.4	2k	$x^{1.5}$	0
mmt.	3	75.4	58.5	2k	$x^{1.5}$	0
mmt.	4	81.2	57.7	2k	$x^{1.5}$	1
mmt.	5	86.2	66.9	2k	$x^{1.5}$	1
mmt.	6	88.0	79.6	2k	x^2	1
mmt.	7	88.0	87.6	2k	x^2	0
mmt.	8	88.3	87.9	2k	x^2	0
mmt.	9	88.1	87.2	2k	x^2	0
mmt.	10	87.5	64.1	5k	x^3	1
Zipf	1	71.0	61.3	2k	$\log(x)$	1
Zipf	2	73.0	72.4	∞	$\log(x)$	0
Zipf	3	78.6	48.7	2k	JSD	1
Zipf	4	86.2	85.5	2k	x^2	0
Zipf	5	86.0	85.4	2k	x^2	0
Zipf	6	84.2	84.4	2k	x^2	0
Zipf	7	82.5	82.4	2k	x^2	0
Zipf	8	81.4	81.2	2k	x^2	0
Zipf	9	80.1	79.8	2k	x^2	0
Zipf	10	80.0	72.0	∞	$\log(x)$	1

Table 4: Eval on BREAD-NOISY

Table 5: F1 and Parameters of the scores that maximized the tune F1 on BREAD-REPEAT and BREAD-NOISY, for all combinations of character n-gram length and score type. The parameters in question are the length-normalization asymptote α, the nonlinearity `nl`, and the Laplace smoothing parameter λ. Perhaps the most interesting thing to note is when the tune/test F1 scores as a function of ngram size: for the two metrics that only detect repetition (TTR and Moment), larger ngrams are generally better, whereas for Zipfianness, utility peaks around 5.