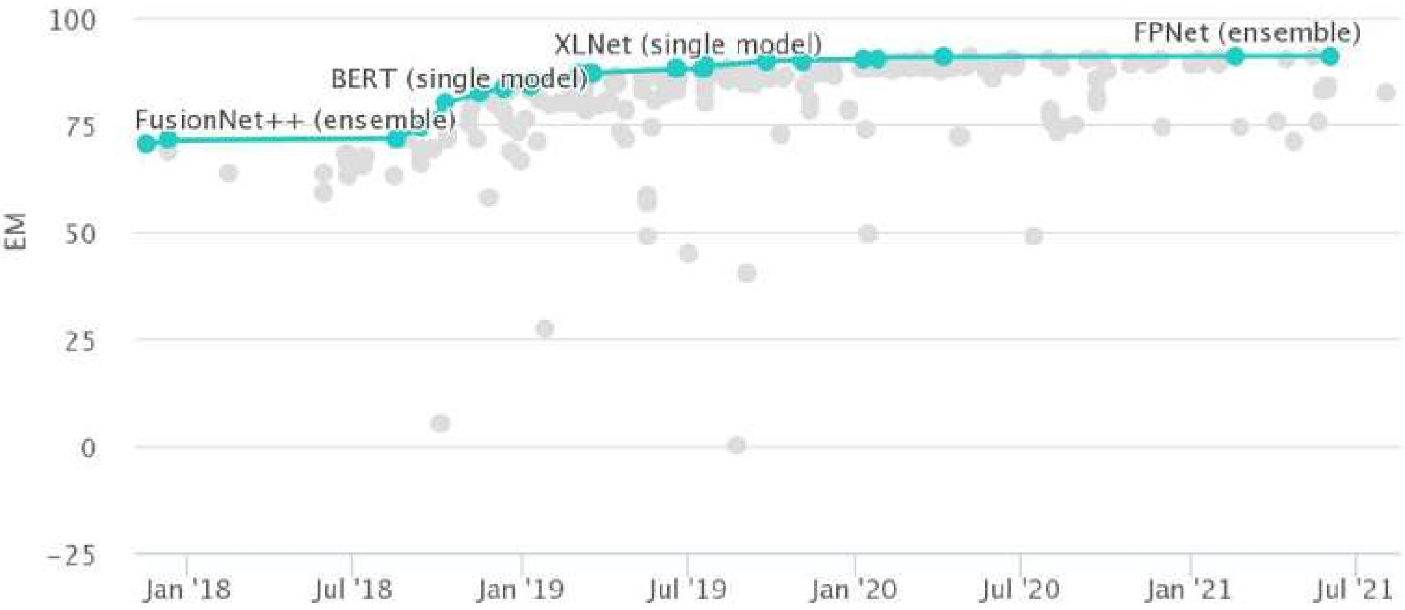
natural language processing

Challenges and Opportunities in NLP Benchmarking

Sebastian Ruder
23 Aug 2021 • 16 min read



Over the last years, models in NLP have become much more powerful, driven by advances in transfer learning. A consequence of this drastic increase in performance is that existing benchmarks have been left behind. Recent models "have outpaced the benchmarks to test for them" (AI Index Report 2021), quickly reaching super-human performance on standard benchmarks such as SuperGLUE and SQuAD. Does this mean that we have solved natural language processing? Far from it.

However, the traditional practices for evaluating performance of NLP models, using a single metric such as accuracy or BLEU, relying on static benchmarks and abstract task formulations no longer work as well in light of models' surprisingly robust *superficial* natural language understanding ability. We thus need to rethink how we design our benchmarks and evaluate our models so that they can still serve as useful indicators of progress going forward.

This post aims to give an overview of challenges and opportunities in benchmarking in NLP, together with some general recommendations. I tried to cover perspectives from recent papers, talks at ACL 2021 as well as at the ACL 2021 Workshop on Benchmarking: Past, Present and Future, in addition to some of my own thoughts.

Header image: Performance on SQuAD 2.0 over time (Credit: Papers with Code)

Table of contents:

What is a benchmark?

> *"Datasets are the telescopes of our field."—Aravind Joshi*

The original use of the term refers to horizontal marks made by surveyors in stone structures, into which an angle-iron could be placed to form a "bench" for a leveling rod. Figuratively, a benchmark refers to a standard point of reference against which things can be compared. A benchmark as it is used in ML or NLP typically has several components: it consists of one or multiple datasets, one or multiple associated metrics, and a way to aggregate performance.

A benchmark sets a standard for assessing the performance of different systems that is agreed upon by the community. To ensure that a benchmark is accepted by the community, many recent benchmarks either select a representative set of standard tasks, such as GLUE or XTREME or actively solicit task proposals from the community, such as SuperGLUE, GEM, or BIG-Bench.

For people in the field, benchmarks are crucial tools to track progress. Aravind Joshi said that without benchmarks to assess the performance of our models, we are just like "astronomers wanting to see the stars but refusing to build telescopes".

For practitioners and outsiders, benchmarks provide an objective lens into a field that enables them to identify useful models and keep track of a field's progress. For instance, the AI Index Report 2021 uses SuperGLUE and SQuAD as a proxy for overall progress in natural language processing.

Reaching human performance on influential benchmarks is often seen as a key milestone for a field. AlphaFold 2 reaching performance competitive with experimental methods on the CASP 14 competition marked a major scientific advance in the field of structural biology.

A brief history of benchmarking

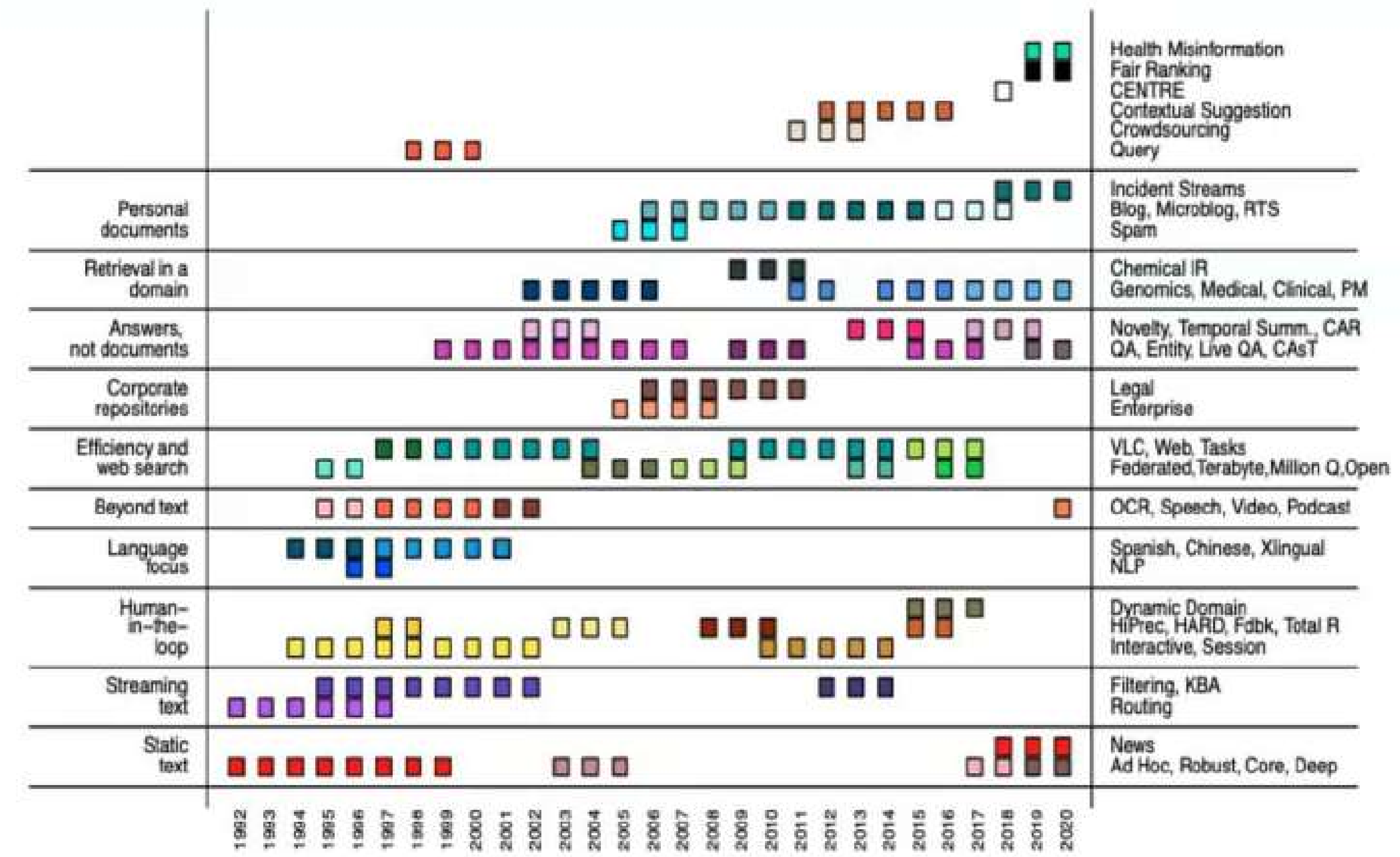
> *"Creating good benchmarks is harder than most imagine."—John R. Mashey; foreword to Systems Benchmarking (2020)*

Benchmarks have a long history of being used to assess the performance of computational systems. The Standard Performance Evaluation Corporation (SPEC), established in 1988 is one of the oldest organisations dedicated to benchmarking the performance of computer hardware. Crucially, SPEC had support from most important companies in the field. Every year, it would release different benchmark sets, each composed of multiple programs, with performance measured as the geometric mean of millions of instructions per second (MIPS).

A recent ML-specific analogue to SPEC is MLCommons, which organises the MLPerf series of performance benchmarks focusing on model training and inference. Similar to SPEC, MLPerf has a broad base of support from academia and industry, building on previous individual efforts for measuring performance such as DeepBench by Baidu or DAWNBench by Stanford.

For US agencies such as DARPA and NIST, benchmarks played a crucial role in measuring and tracking scientific progress. Early benchmarks for automatic speech recognition (ASR) such as TIMIT and Switchboard were funded by DARPA and coordinated by NIST starting in 1986. Later influential benchmarks in other areas of ML such as MNIST were also based on NIST data.

For language technology and information retrieval (IR), NIST ran the DARPA-funded TREC series of workshops covering a wide array of tracks and topics, which can be seen below. TREC organised competitions built on an evaluation paradigm pioneered by Cranfield in the 1960s where models are evaluated based on a set of test collections, consisting of documents, questions, and human relevance judgements. As the variance in performance across topics is large, scores are averaged over many topics. TREC's "standard, widely available, and carefully constructed set of data laid the groundwork for further innovation" (Varian, 2008) in IR.



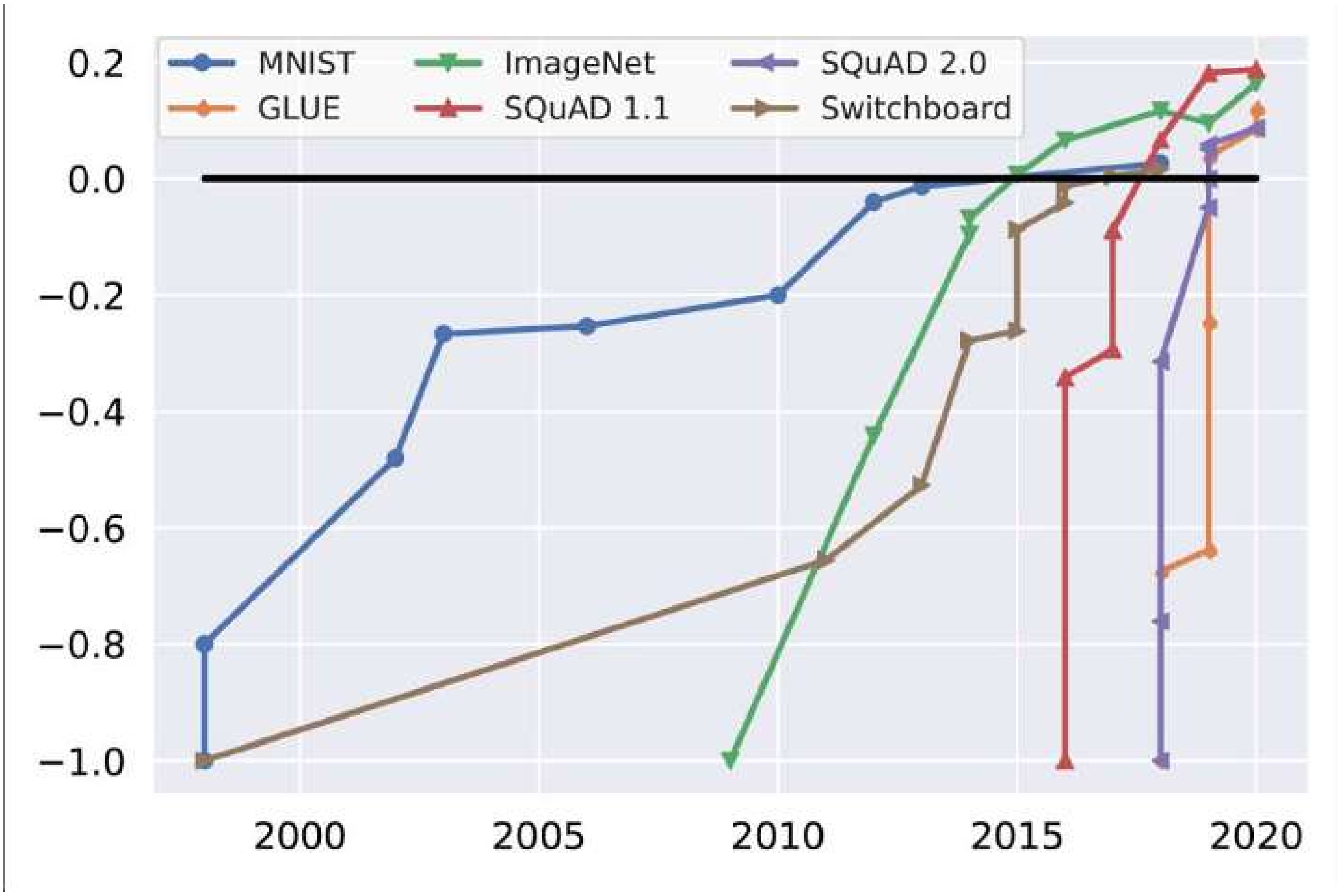
Tasks and topics in the TREC workshops from 1992–2020 (Credit: Ellen Voorhees)

Many recent influential benchmarks such as ImageNet, SQuAD, or SNLI are large in scale, consisting of hundreds of thousands of examples and were developed by academic groups at well-funded universities. In the era of deep learning, such large-scale datasets have been credited as one of the pillars driving progress in research, with fields such as NLP or biology witnessing their 'ImageNet moment'.

As models have become more powerful and general-purpose, benchmarks have become more application-oriented and increasingly moved from single-task to multi-task and single-domain to multi-domain benchmarks. Key examples of these trends are a

transition from a focus on core linguistic tasks such as part-of-speech tagging and transition from a focus on core linguistic tasks such as part-of-speech tagging and dependency parsing to tasks that are closer to the real-world such as goal-oriented dialogue and open-domain question answering (Kwiatkowski et al., 2019); the emergence of multi-task datasets such as GLUE; and multi-modality datasets such as WILDS.

However, while it took more than 15 years to achieve superhuman performance on classic benchmarks such as MNIST or Switchboard, models have achieved superhuman performance on more recent benchmarks such as GLUE and SQuAD 2.0 within about a year of their release, as can be seen in the figure below. At the same time, we know that the capabilities that these benchmarks aim to test, such as general question answering are far from being solved.



Benchmark saturation over time for popular benchmarks. Initial performance and human performance are normalised to -1 and 0 respectively (Kiela et al., 2021).

Another factor that has contributed to the saturation of these benchmarks is that limitations and annotation artefacts of recent datasets have been identified much more quickly compared to earlier benchmarks. In SNLI, annotators have been shown to rely on heuristics, which allow models to make the correct prediction in many cases using the hypothesis alone (Gururangan et al., 2018) while models trained on SQuAD are subject to adversarially inserted sentences (Jia and Liang, 2017).

to adversarially inserted sentences (Jia and Liang, 2017).

A recent trend is the development of adversarial datasets such as Adversarial NLI (Nie et al., 2020), Beat the AI (Bartolo et al., 2020), and others where examples are created to be difficult for current models. Dynabench (Kiela et al., 2021), a recent open-source platform has been designed to facilitate the creation of such datasets. A benefit of such benchmarks is that they can be dynamically updated to be challenging as new models emerge and consequently do not saturate as easily as static benchmarks.

Metrics matter

> *"When you can measure what you are speaking of and express it in numbers, you know that on which you are discussing. But when you cannot measure it and express it in numbers, your knowledge is of a very meagre and unsatisfactory kind."—Lord Kelvin*

When it comes to measuring performance, metrics play an important and often under-appreciated role. For classification tasks, accuracy or F-score metrics may seem like the obvious choice but—depending on the application—different types of errors incur different costs. For fine-grained sentiment analysis, confusing between *positive* and *very positive* may not be problematic while mixing up *very positive* and *very negative* is. Chris Potts highlights an array of practical examples where metrics like F-score fall short, many in scenarios where errors are much more costly.

Designing a good metric requires domain expertise. MLPerf measures the wallclock time required to train a model to a dataset-specific performance target, a measure informed by both end use cases and the difficulty to compare other efficiency metrics such as FLOPS across models. In ASR, only the percentage of correctly transcribed words (akin to accuracy) was initially used as the metric. The community later settled on word error rate, i.e. $\frac{\text{substitutions+deletions+insertions}}{\text{number of words in reference}}$ as it directly reflects the cost of correcting transcription errors.

There is a large difference between metrics designed for decades-long research and metrics designed for near-term development of practical applications, as highlighted by Mark Liberman. For developing decade-scale technology, we need efficient metrics that can be crude as long as they point in the general direction of our distant goal. Examples of such metrics are the word error rate in ASR (which assumes that all words are equally important) and BLEU in machine translation (which assumes that word order is not important). In contrast, for the evaluation of practical technology we need metrics that are designed with the requirements of specific applications in mind and that can consider different types of errors.

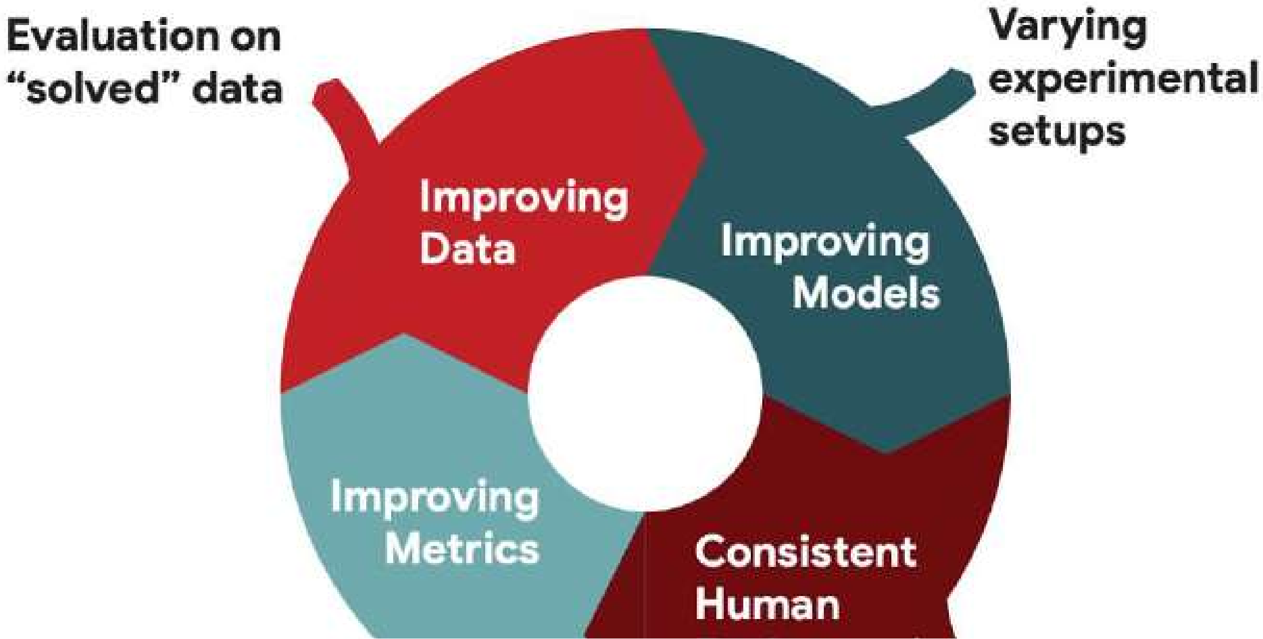
consider different types of error classes.

The rapid increase in model performance in recent years has catapulted us from the decade-long to the near-term regime for many applications. However, even in this more application-oriented setting we are still relying on the same metrics that we have used to measure long-term research progress thus far. In a recent meta-analysis, Marie et al. (2021) found that 82% papers of machine translation (MT) papers between 2019–2020 only evaluate on BLEU—despite 108 alternative metrics having been proposed for MT evaluation in the last decade, many of which correlate better with human judgements. As models become stronger, metrics like BLEU are no longer able to accurately identify and compare the best-performing models.

While evaluation of natural language generation (NLG) models is notoriously difficult, standard n-gram overlap-based metrics such as ROUGE or BLEU are furthermore less suited to languages with rich morphology, which will be assigned relatively lower scores.

A recent trend in NLG is towards the development of automatic metrics such as BERTScore (Zhang et al., 2020) that leverage the power of large pre-trained models. A recent modification of this method makes it more suitable for near-term MT evaluation by assigning larger weights to more difficult tokens, i.e. tokens that are translated correctly only by few MT systems (Zhan et al., 2021).

In order to continue to make progress, we need to be able to update and refine our metrics, to replace efficient simplified metrics with application-specific ones. The recent GEM benchmark, for instance, explicitly includes metrics as a component that should be improved over time, as can be seen below.



Evaluation with gameable metrics
Eval
Non-repeatable human evaluation

Opportunities (circle) and challenges of benchmark evaluation (Gehrmann et al., 2021).

Recommendations:

1. Consider metrics that are better suited to the downstream task and language.
2. Consider metrics that highlight the trade-offs of the downstream setting.
3. Update and refine metrics over time.

Consider the downstream use case

> *"[...] benchmarks shape a field, for better or worse. Good benchmarks are in alignment with real applications, but bad benchmarks are not, forcing engineers to choose between making changes that help end users or making changes that only help with marketing."—David A. Patterson; foreword to Systems Benchmarking (2020)*

NLP technology is increasingly used in many real-world application areas, from creative self-expression to fraud detection and recommendation. It is thus key to design benchmarks with such real-world settings in mind.

A benchmark's data composition and evaluation protocol should reflect the real-world use case, as highlighted by Ido Dagan. For relation classification, for instance, the FewRel dataset lacks some important realistic properties, which Few-shot TACRED addresses. For binary sentiment classification on the IMDb dataset, only highly polarised positive and negative reviews are considered and labels are exactly balanced. In information retrieval, retrieving relevant before non-relevant documents is necessary *but not sufficient* for real-world usage.

As a first rule of social responsibility for NLP, Chris Potts proposes "Do exactly what you said you would do". As researchers in the field, we should communicate clearly what performance on a benchmark reflects and how this corresponds to real-world settings. In a similar vein, Bowman and Dahl (2021) argue that good performance on a benchmark should imply robust in-domain performance on the task.

However, the real-world application of a task may confront the model with data different from its training distribution. It is thus key to assess the robustness of the model and how well it generalises to such out-of-distribution data, including data with a temporal shift and data from other language varieties.

In light of the limited linguistic diversity in NLP research (Joshi et al., 2020), it is furthermore crucial not to treat English as the singular language for evaluation. When designing a benchmark, collecting—at a minimum—test data in other languages may help to highlight new challenges and promote language inclusion. Similarly, when evaluating models, leveraging the increasing number of non-English language datasets in tasks such as question answering and summarisation (Hasan et al., 2021) can provide additional evidence of a model's versatility.

Ultimately, considering the challenges of current and future real-world applications of language technology may provide inspiration for many new evaluations and benchmarks. Benchmarks are among the most impactful artefacts of our field and often lead to entirely new research directions, so it is crucial for them to reflect real-world and potentially ambitious use cases of our technology.

Recommendations:

1. Design the benchmark and its evaluation so that it reflects the real-world use case.
2. Evaluate in-domain and out-of-domain generalisation.
3. Collect data and evaluate models on other languages.
4. Take inspiration from real-world applications of language technology.

Fine-grained evaluation

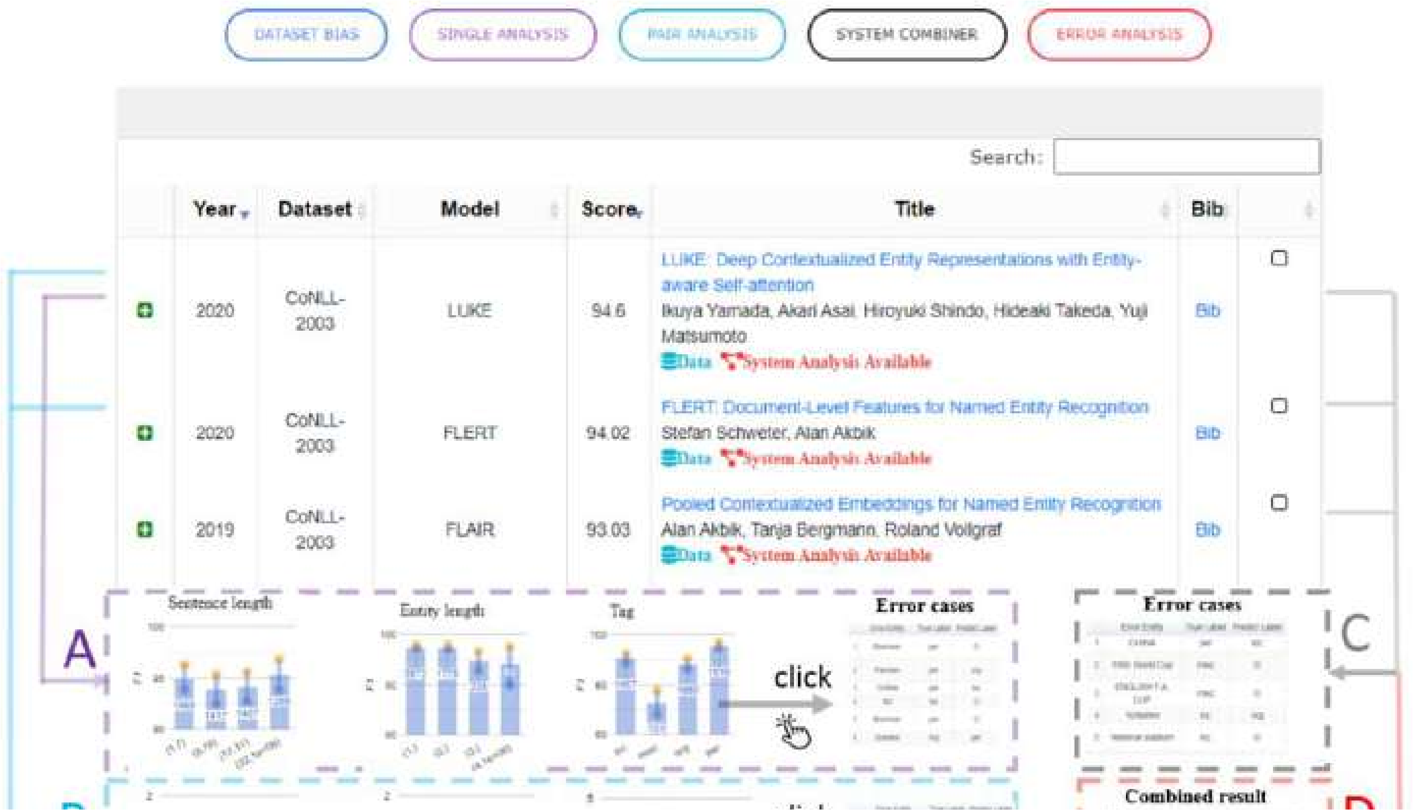
> *"No matter how much people want performance to be a single number, even the **right** mean with no distribution can be misleading, and the **wrong** mean certainly is no better."—John R. Mashey*

The downstream use case of technology should also inform the metrics we use for evaluation. In particular, for downstream applications often not a single metric but an array of constraints need to be considered. Rada Mihalcea calls for moving away from just focusing on accuracy and to focus on other important aspects of real-world scenarios. What is important in a particular setting, in other words, the utility of an NLP system, ultimately depends on the requirements of each individual user (Ethayarajh and Jurafsky, 2020).

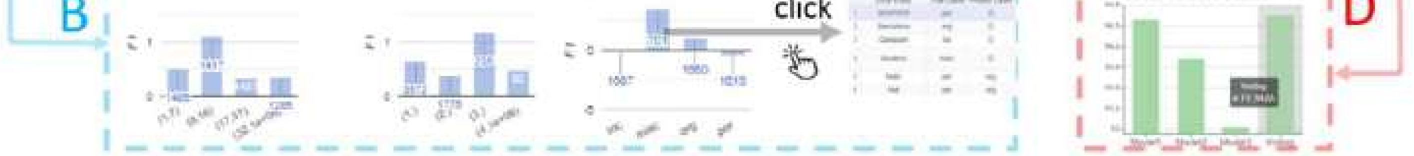
Societal needs have generally not been emphasised in machine learning research (Birhane et al., 2021). However, for real-world applications it is particularly crucial that a model does not exhibit any harmful social biases. Testing for such biases in a task-specific manner should thus become a standard part of algorithm development and model evaluation.

Another aspect that is important for practical applications is efficiency. Depending on the application, this can relate to both sample efficiency, FLOPS, and memory constraints. Evaluating models in resource-constrained settings can often lead to new research directions. For instance, the EfficientQA competition (Min et al., 2020) at NeurIPS 2020 demonstrated the benefits of retrieval augmentation and large collections of weakly supervised question–answer pairs (Lewis et al., 2021).

In order to better understand the strengths and weaknesses of our models, we furthermore require more fine-grained evaluation across a *single* metric, highlighting on what types of examples models excel and fail at. ExplainaBoard (Liu et al., 2021) implements such a fine-grained breakdown of model performance across different tasks, which can be seen below. Another way to obtain a more fine-grained estimate of model performance is to create test cases for specific phenomena and model behaviour, for instance using the CheckList framework (Ribeiro et al., 2020).



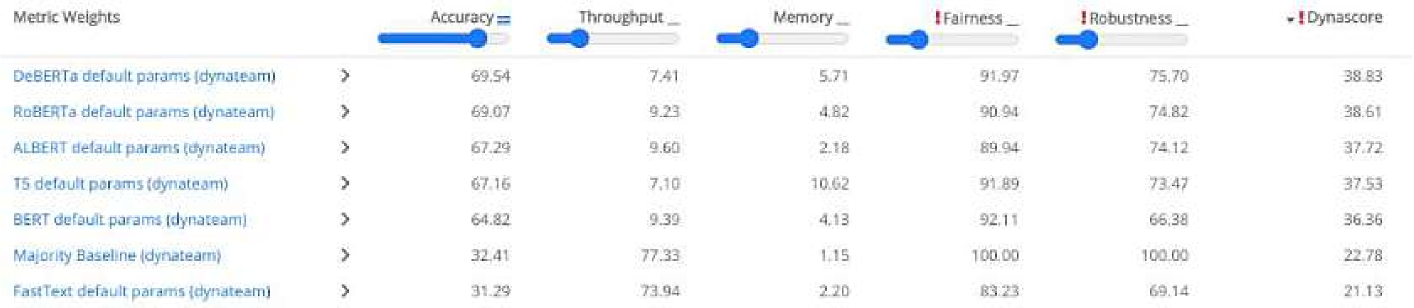
	Year	Dataset	Model	Score	Title	Bib	
⊞	2020	CoNLL-2003	LUKE	94.6	LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, Yuji Matsumoto Data System Analysis Available	Bib	□
⊞	2020	CoNLL-2003	FLERT	94.02	FLERT: Document-Level Features for Named Entity Recognition Stefan Schweter, Alan Akbik Data System Analysis Available	Bib	□
⊞	2019	CoNLL-2003	FLAIR	93.03	Pooled Contextualized Embeddings for Named Entity Recognition Alan Akbik, Tanja Bergmann, Roland Vollgraf Data System Analysis Available	Bib	□



The ExplainaBoard interface for the CoNLL-2003 NER dataset for the three best systems including single-system analyses for the best system (A), pairwise analysis results for the top-2 systems (B), a common error table (C), and combined results (D) (Liu et al., 2021).

As individual metrics can be flawed, it is key to evaluate across multiple metrics. When evaluating on multiple metrics, scores are typically averaged to obtain a single score. A single score is useful to compare models at a glance and provides people outside the community a clear way to assess model performance. However, using the arithmetic mean is not appropriate for all purposes. SPEC used the geometric mean, $\sqrt[n]{x_1 x_2 \ldots x_n}$, which is useful when aggregating values that are exponential in nature, such as runtimes.

An alternative is to use a weighted sum and to enable the user to define their own weights for each component. DynaBench uses such a dynamic weighting to weight the importance of model performance but also consider model throughput, memory consumption, fairness, and robustness, which enables users to effectively define their own leaderboard (Ethayarajh and Jurafsky, 2020), as can be seen below.



Metric Weights		Accuracy	Throughput	Memory	! Fairness	! Robustness	! Dynascore
DeBERTa default params (dynateam)	>	69.54	7.41	5.71	91.97	75.70	38.83
RoBERTa default params (dynateam)	>	69.07	9.23	4.82	90.94	74.82	38.61
ALBERT default params (dynateam)	>	67.29	9.60	2.18	89.94	74.12	37.72
T5 default params (dynateam)	>	67.16	7.10	10.62	91.89	73.47	37.53
BERT default params (dynateam)	>	64.82	9.39	4.13	92.11	66.38	36.36
Majority Baseline (dynateam)	>	32.41	77.33	1.15	100.00	100.00	22.78
FastText default params (dynateam)	>	31.29	73.94	2.20	83.23	69.14	21.13

Dynamic metric weighting in the DynaBench natural language inference task leaderboard

Recommendations:

1. Move away from using a single metric for performance evaluation.
2. Evaluate social bias and efficiency.
3. Perform a fine-grained evaluation of models.
4. Consider how to aggregate multiple metrics.

The long tail of benchmark performance

Given that current models perform surprisingly well on in-distribution examples, it is time to shift our attention to the tail of the distribution, to outliers and atypical examples. Rather than considering only the average case, we should care more about the worst case and subsets of our data where our models perform the worst.

As models become more powerful, the fraction of examples where the performance of models differs and that thus will be able to differentiate between strong and the best models will grow smaller. To ensure that evaluation on this long tail of examples is reliable, benchmarks need to be large enough so that small differences in performance can be detected. It is important to note that larger models are not uniformly better across all examples (Zhong et al., 2021).

As an alternative, we can develop mechanisms that allow us to identify the best systems with few examples. This is particularly crucial in settings where assessing performance of many systems is expensive, such as in human evaluation for natural language generation. Mendonça et al. (2021) frame this as an online learning problem in the context of MT.

Benchmarks can also focus on annotating examples that are much more challenging. This is the direction taken by recent adversarial benchmarks (Kiela et al., 2021). Such benchmarks, as long as they are not biased towards a specific model, can be a useful complement to regular benchmarks that sample from the natural distribution. These directions benefit from the development of *active* evaluation methods to identify or generate the most salient and discriminative examples to assess model performance as well as interpretability methods to allow annotators to better understand models' decision boundaries.

As the budget (and thus size) of benchmarks generally remains constant, statistical significance testing becomes even more important as it enables us to reliably detect qualitatively relevant performance differences between systems.

In order to perform reliable comparisons, the benchmark's annotations should be correct and reliable. However, as models become more powerful, many instances of what look like model errors may be genuine examples of ambiguity in the data. Bowman and Dahl (2021) highlight how a model may exploit clues about such disagreements to reach super-human performance on a benchmark.

If possible, benchmarks should aim to collect multiple annotations to identify ambiguous

If possible, benchmarks should aim to collect multiple annotations to identify ambiguous examples. Such information may provide a useful learning signal (Plank et al., 2014) and can be helpful for error analysis. In light of such ambiguity, it is even more important to report standard metrics such as inter-annotator agreement as this provides a ceiling for a model's performance on a benchmark.

Recommendations:

1. Include many and/or hard examples in the benchmark.
2. Conduct statistical significance testing.
3. Collect multiple annotations for ambiguous examples.
4. Report inter-annotator agreement.

Large-scale continuous evaluation

> *"When a measure becomes a target, it ceases to be a good measure."—Goodhart's law*

Multi-task benchmarks such as GLUE have become key indicators of progress but such static benchmark collections quickly become outdated. Modelling advances generally also do not lead to uniform progress across tasks. While models have achieved super-human performance on most GLUE tasks, a gap to 5-way human agreement remains on some tasks such as CoLA (Nangia and Bowman, 2019). On XTREME, models have improved much more on cross-lingual retrieval.

In light of the fast pace of model improvements, we are in need of more nimble mechanisms for model evaluation. Specifically, beyond dynamic *single-task* evaluations such as DynaBench, it would be useful to define a dynamic *collection* of benchmark datasets on which models have not reached human performance. This collection should be managed by the community, with datasets removed or down-weighted as models reach human performance and new challenging datasets being regularly added. Such a collection needs to be versioned, to enable updates beyond the cycle of academic review and to enable replicability and comparison to prior approaches.

Existing multi-task benchmarks such as GEM (Gehrmann et al., 2021), which explicitly aims to be a 'living' benchmark, generally include around 10–15 different tasks. Rather than limiting the benchmark to a small collection of representative tasks, in light of the number of new datasets constantly being released, it might be more useful to include a larger cross-section of NLP tasks. Given the diverse nature of tasks in NLP, this would provide a more robust and up-to-date evaluation of model performance. LUGE by Baidu is a step towards such a large collection of tasks for Chinese natural language processing, currently consisting of 28 datasets.

The collection of tasks can be broken down in various ways, providing more a fine-grained assessment of model capabilities. Such a breakdown may be particularly insightful if tasks or subsets of task data are categorised according to the behaviour they are testing. BIG-Bench, a recent collaborative benchmark for language model probing includes a categorisation by keyword.

A key challenge for such large-scale multi-task evaluation is accessibility. Tasks need to be available in a common input format so that they can be run easily. In addition, tasks should be efficient to run or alternatively infrastructure needs to be available to run tasks even without much compute.

Another point of consideration is that such a collection favours large general-purpose models, which are generally trained by deep-pocketed companies or institutions. Such models, however, are already used as the starting point for most current research efforts and can be—once trained—more efficiently used via fine-tuning, distillation, or pruning.

Recommendations:

1. Consider collecting and evaluating on a large, diverse, versioned collection of NLP tasks.

Conclusion

In order to keep up with advances in modelling, we need to revisit many tacitly accepted benchmarking practices such as relying on simplistic metrics like F1-score and BLEU. To this end, we should take inspiration from real-world applications of language technology and consider the constraints and requirements that such settings pose for our models. We should also care more about the long tail of the distribution as that is where improvements will be observed for many applications. Lastly, we should be more rigorous in the evaluation on our models and rely on multiple metrics and statistical significance testing, contrary to current trends.

Citation