Test Harder than You Train: Probing with Extrapolation Splits

Jenny Kunz and **Marco Kuhlmann**
Dept. of Computer and Information Science
Linköping University
jenny.kunz@liu.se and marco.kuhlmann@liu.se

Abstract

Previous work on probing word representations for linguistic knowledge has focused on interpolation tasks. In this paper, we instead analyse probes in an extrapolation setting, where the inputs at test time are deliberately chosen to be 'harder' than the training examples. We argue that such an analysis can shed further light on the open question whether probes actually decode linguistic knowledge, or merely learn the diagnostic task from shallow features. To quantify the hardness of an example, we consider scoring functions based on linguistic, statistical, and learning-related criteria, all of which are applicable to a broad range of NLP tasks. We discuss the relative merits of these criteria in the context of two syntactic probing tasks, part-of-speech tagging and syntactic dependency labelling. From our theoretical and experimental analysis, we conclude that distance-based and hard statistical criteria show the clearest differences between interpolation and extrapolation settings, while at the same time being transparent, intuitive, and easy to control.

1 Introduction

The use of contextualised language models such as ELMo and BERT has brought about remarkable performance gains on a wide range of downstream tasks (Peters et al., 2018a; Devlin et al., 2019); but the question to what extent these models have acquired linguistic knowledge remains open. One way to investigate this question is through the use of *probing classifiers* trained to solve diagnostic prediction tasks that are considered to require linguistic information, such as parts-of-speech, syntactic structure, or semantic roles (Belinkov et al., 2017a; Conneau et al., 2018; Tenney et al., 2019). However, what conclusions can be drawn from probing experiments is disputed. In particular, a central point of debate is how to know whether probes 'decode linguistic knowledge' or simply 'learn to solve the diagnostic task' (Hewitt and Liang, 2019). We suggest that new methods that define more rigorous and harder challenges are needed to get further insights into the capabilities and limitations of probes and probing methodology.

In this paper, we propose to analyse probes in the context of an extrapolation setting, where the inputs at test time are deliberately chosen to be 'harder' than the training examples. While machine learning models and neural networks in particular have proved to be very effective learners in *interpolation* scenarios, where the examples at training time and those at test time are drawn from the same (idealised) underlying distribution, the ability of these models to *extrapolate* from the training data appears to be limited (Dubois et al., 2020). At the same time, extrapolation has been proposed as a litmus test for abstract reasoning in neural networks (Barrett et al., 2018). In the context of probing, we posit that the better the extrapolation capability of a probe, i.e. the higher its performance even in situations where the training and the test examples are substantially different, the more evidence we have for claiming that the probe actually uses abstract linguistic knowledge encoded in the input word representations.

To construct extrapolation challenges, we propose a conceptually simple approach where we start from standard probing datasets, stratify them based on the 'hardness' of examples, and then use the 'easy' examples for training and the 'hard' ones for testing (§ 3). The central decision in this approach is how to measure 'hardness'. Here we identify different scoring functions based on criteria grounded in linguistic theories, statistical properties of the base dataset, and learning behaviour. We apply these scoring functions to create extrapolation challenges from two standard probing tasks, part-of-speech tagging and syntactic dependency labelling (§ 4), and use the results of our experiments to discuss the merits of our approach (§ 5).

Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 15–25

2 Related Work

The method that we propose in this paper synthesises several strands of related work:

2.1 Probing (and its Limitations)

Probing aims at detecting linguistic knowledge in word representations. While this can be done in a zero-shot setting (Goldberg, 2019; Talmor et al., 2020) or as a structural probe (Hewitt and Manning, 2019), a dominant approach is to train and evaluate simple classifiers on relevant diagnostic tasks (Belinkov et al., 2017b; Hewitt and Liang, 2019), where the classifier receives one word representations at a time as its input. This is based on the idea that the accuracy of the trained probe can indicate to what extent the representations encode linguistic knowledge that is useful for the diagnostic task.

Recent work has questioned the validity of this methodology, suggesting that analysis should shift focus to measuring 'amount of effort' rather than task-based accuracy (Pimentel et al., 2020; Voita and Titov, 2020). Moreover, many probing tasks are relatively easy to learn with local context and strong independence assumptions. It thus remains unclear whether the probed word representations actually encode linguistic knowledge, contain predictive but superficial features extracted from the words' linear context (Kunz and Kuhlmann, 2020), or rather provide an effective initialisation for the probing classifier (Prasanna et al., 2020).

2.2 Interpolation and Extrapolation

A growing body of research suggests that, while deep neural models can reach remarkable performance in interpolation settings, they often fail to extrapolate, i.e. to generalise to inputs outside the range of the training data. For example, Barrett et al. (2018) show that in visual reasoning, popular models such as ResNets perform at levels barely above a random choice baseline in extrapolation settings. As the ability to extrapolate is generally considered a hallmark of intelligence, such findings raise the question whether deep models are capable of human-like reasoning. Similar concerns come from observations that performance can suffer greatly when models are confronted with adversarial examples (Goodfellow et al., 2015; Jia and Liang, 2017) or challenge sets (Zellers et al., 2018, 2019). Zellers et al. (2019) suggest that deep models may 'pick up on dataset-specific distributional biases' instead of learning the actual task.

In the domain of natural language understanding, authors have shown that Transformers lack the capability to extrapolate to longer sequences (Dubois et al., 2020) and number representations of higher values (Weiss et al., 2018); and that even large neural models such as RoBERTa can compare ages only within a restricted range (Talmor et al., 2020). Evidently, test data outside the training distribution is a great challenge, and contextualised language models are easily broken on such data.

2.3 What are Hard Examples?

Most of the aforementioned works on extrapolation and abstraction employ synthetic datasets or adversarial attacks to challenge a model. Here we propose a method based on the stratification of existing probing datasets according to a measure of expected difficulty or 'hardness'.

2.3.1 Readability Criteria

One way to quantify the difficulty of training examples is to use readability criteria, which are typically motivated on linguistic grounds or with reference to studies on human language processing (Kocmi and Bojar, 2017; Platanios et al., 2019). A widely used and widely applicable metric is sentence length, which is intuitive and straightforward to measure (Sherman, 1893), but only weakly correlated with processing complexity (Bailin and Grafstein, 2001). There are also many more specific measures, such as the respective averages of parse tree height, length of arcs in syntactic dependency trees, number of noun phrases and number of verb phrases, or word frequency. These measures often inform systems that help authors improve writing quality, and automatically transform texts to make them more understandable or accessibile (Zamanian and Heydari, 2012), but are also used to evaluate systems such as dependency parsers (McDonald and Nivre, 2007; Kulmizev et al., 2019).

2.3.2 Learning-Based Criteria

Instead of using inherent properties, another way to quantify the hardness of training examples is to look at the effort that a model has to put into learning them. Here we take inspiration from developments in curriculum learning, which moved from heuristic metrics on artificial datasets (Bengio et al., 2009) to learning-specific metrics. In particular, *self-paced learning* employs the loss of a model to rate and rank the difficulty of examples in a dataset (Kumar et al., 2010; Hacohen and Wein-

shall, 2019). This approach is widely used, but has also been criticised as being inherently model-specific (Lalor and Yu, 2020). Other approaches that have been successfully employed in curriculum learning are rankings based on the norms of word embeddings (Liu et al., 2020) and on model uncertainty (Zhou et al., 2020).

3 Experimental Setup

In this section we present our specific approach to creating extrapolation datasets, and the setup for our empirical evaluation.[1]

3.1 Word Representations

Our word representations come from the English BERT base (uncased) model (Devlin et al., 2019), accessed via the the Transformers library (Wolf et al., 2020). We probe on the hidden representations of words in all 13 layers, including the uncontextualised layer 0 as a baseline. For words that BERT tokenises into several word pieces, we use the last piece as the representation for the word.

3.2 Probing Classifier

The probing classifier is the same in all experiments: a feed-forward network with one hidden layer, 64 hidden units and ReLU activation. We train this classifier with cross-entropy loss for 5 epochs using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001 and a batch size of 64. Our implementation uses PyTorch (Paszke et al., 2017).

3.3 Tasks and Datasets

We use two prototypical diagnostic tasks which have been widely studied in the probing literature: part-of-speech (POS) tagging and syntactic dependency labelling. The training and test data for both tasks comes from the English Web Treebank (EWT) as released by the Universal Dependencies project (Nivre et al., 2020) (v2.5). More specifically, we extract our examples from two 1,000-sentence sets $\mathcal{S}_{\text{train}}$ and $\mathcal{S}_{\text{test}}$, randomly sampled from the training and the development section of the EWT, respectively.[2] We write s_i^t to denote the ith sentence in $\mathcal{S}_t$, w_{ij}^t to denote its jth word, and x_{ij}^t to denote the BERT representation of w_{ij}. We omit the superscript when the base set (training or development) is understood or irrelevant.

T1: Part-of-speech tagging This is our prototypical single-word labelling task. Examples take the form $e = (x_{ij}, y_{ij})$, where x_{ij} is the representation of a single word w_{ij}, and y_{ij} is the corresponding gold-standard tag. The POS class of a word captures some of its most basic syntactic properties, and can be predicted with local or even without context information at a high accuracy. For our data, probes trained on contextualised word representations usually show a tagging accuracy above 95%, with the highest-performing layers being the lower middle or middle layers of a model (Peters et al., 2018b; Tenney et al., 2019).

T2: Syntactic dependency labelling In this task, which intends to capture the hierarchical structure of a sentence, we aim to predict the grammatical relation for a given dependency arc. Examples take the form $e = ((x_{ij}, x_{ik}), y_{ik})$, where x_{ij} and x_{ik} are word representations of the head and dependent, respectively, and y_{ik} is the gold-standard dependency label. The performance of simple probes on this task is usually lower than for POS tagging, as the syntactic information that is required to accurately predict the labels is more complex and depends on a larger context. Accuracy can however still exceed 90% in the highest-performing layers, which are usually the higher middle layers.

3.4 Scoring Functions

We next introduce the inventory of measures that we use to quantify the 'hardness' of training examples. Formally, each measure is a real-valued function m whose domain is the set of all task-specific examples. If $m(e) > m(e')$, we say that example e is *harder* than example e'.

3.4.1 Length-based Criteria

These scoring functions refer to two different notions of length:

Sentence length (T1, T2) The most basic length is that of the sentence s_i from which the example is derived. Using $|\cdot|$ to denote length,

$$m(x_{ij}, y_{ij}) = |s_i| \qquad \text{(for T1)}$$
$$m((x_{ij}, x_{ik}), y_{ik}) = |s_i| \qquad \text{(for T2)}$$

[1] All code necessary to reproduce our experiments is publicly available at https://github.com/jekunz/extrapolation.

[2] We sub-sample the full data to reduce training time and save resources. Preliminary experiments showed the same trends that we report here for the full data.

Arc length (T2) For dependency labelling, we may also consider the length of the dependency arc: $m((x_{ij}, x_{ik}), y_{ik}) = |j - k|$.

3.4.2 Statistical Criteria

For part-of-speech tagging, we consider criteria related to the distribution of the tags:

Tag proportions (T1) Here the hardness score of an example is the inverse relative frequency of the represented word's gold-standard POS tag in the training set. More formally, for a word w_{ij} from $\mathcal{S}_{\text{train}}$ and a tag t, let $f(w_{ij}, t)$ be the relative frequency of t among all possible tags for w_{ij}; then $m(x_{ij}, y_{ij}) = 1 - f(w_{ij}, y_{ij})$. For examples e that represent words which do not occur in $\mathcal{S}_{\text{train}}$, we let $m(e) = 1$; out-of-vocabulary words will thus always yield the hardest examples.

Most frequent tag (T1) In a related setup, we consider an example to be 'easy' if its gold-standard tag is the most frequent tag (mft) in the training set, and 'hard' otherwise. Formally,

$m(x_{ij}, y_{ij}) = 1 - \mathbb{1}[y_{ij} \text{ is the mft for } w_{ij} \text{ in } \mathcal{S}_{\text{train}}]$.

3.4.3 Learning-based Criteria

Here we implement ideas from curriculum learning. We first train an ensemble of 10 classifiers on all examples derived from $\mathcal{S}_{\text{train}}$. Each classifier has the same architecture and training regime as our probe (§ 3.2), but uses a different random seed. We then use this ensemble to define the hardness of each example e as follows:

Sample-specific loss (T1, T2) Here we let $m(e)$ be the sample-specific loss for e, relative to its gold-standard tag or label, averaged over the 10 classifiers in the ensemble.

Speed of learning (T1, T2) Here we want to classify an example as 'hard' if the probe needs a long time (a large number of updates) to learn it reliably. To implement this idea, at seven specified checkpoints early into training, we let each of the classifiers in the ensemble predict the tag or label of each example e, and define

$$m(e) = 1/(c + 1)\,,$$

where c is the total number of correct predictions. For our checkpoints, we use the partially trained classifiers after 2^n batch updates, for $1 \leq i \leq 7$. As a consequence, the minimal value for c is 0 (never correctly classified), and the maximal value is $7 \cdot 10$ (correctly classified at every checkpoint, by every classifier).

3.5 Easy Sets and Hard Sets

The last step of our approach is to use our scoring functions to split the set of all task-specific examples into an 'easy' set and a 'hard' set. Here, for each specific experiment we choose two values m_1 and m_2 and let

$$\mathcal{D}_{\text{easy}} = \{e \,|\, m(e) < m_1\}$$
$$\mathcal{D}_{\text{hard}} = \{e \,|\, m(e) > m_2\}$$

The difference $m_2 - m_1$ denotes the *distance* between $\mathcal{D}_{\text{easy}}$ and $\mathcal{D}_{\text{hard}}$. The specific criteria according to which we choose the split points vary:

Linguistic criteria For sentence length we base our choice on the classification of Flesch and Gould (1949). Specifically, for $\mathcal{D}_{\text{easy}}$ we use the lengths less than 17 words ($m_1 = 17$), corresponding to (at most) 'fairly easy' readability, understood by 88% of adults in the referenced study. For $\mathcal{D}_{\text{hard}}$ we use the lengths greater than 29 words ($m_2 = 29$), classified as (at least) 'very difficult', understood by 4.5% of adults.

Distributional criteria For the remaining scoring functions, we choose split points based on the empirical distribution of the scores: We let m_1 be the 50th percentile (i.e., the median score), and m_2 be the 75th percentile. The only exception to this rule is for the most frequent tag criterion, as explained in § 3.4.2.

Note that, with our strategies of choosing split points, the sizes of the specific 'easy' and 'hard' sets that we use for each experiment differ from the full set, getting as low as half the number of all examples. To assess the impact of this reduction, in control experiments we randomly sub-sampled the 'standard' training sets down to 50% of their original size, but only observed a moderate drop in accuracy (at most 1%).

3.6 Evaluation

For each experiment, we consider two setups:

- In the *extrapolation setup*, we train on the examples in $\mathcal{D}_{\text{easy}}$ and test on those in $\mathcal{D}_{\text{hard}}$.
- In the *control setup*, we also test on the examples in $\mathcal{D}_{\text{hard}}$, but train on the full set of examples.

For both setups, we report the mean over 10 random seeds of the best accuracy of each classifier among the 5 epochs for which it was trained.

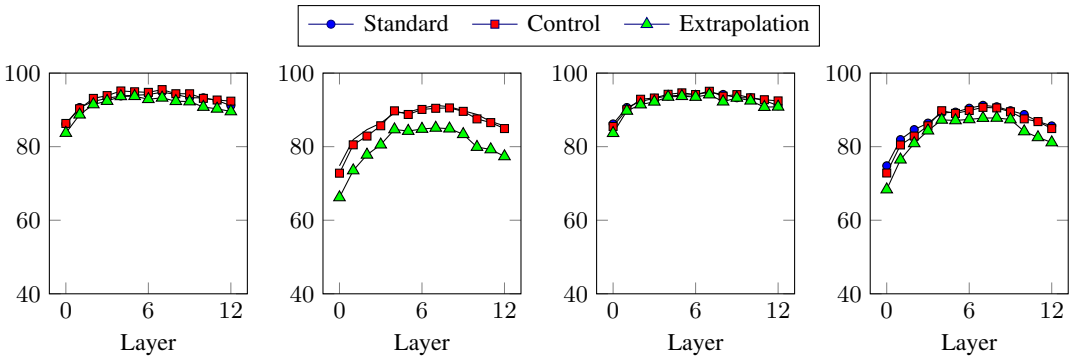

Figure 1: Extrapolation based on sentence length. From left to right: part-of-speech tagging (T1), linguistic criterion; dependency labelling (T2), linguistic; T1, distributional criterion; T2, distributional. In all plots, the x-axis corresponds to the BERT layer used for prediction, and the y-axis corresponds to the mean accuracy.

To interpret an experiment, we compare the two accuracy values: If the accuracy in the extrapolation setup is significantly lower than that of the control, we want to conclude that the probe lacks the ability to extrapolate from 'easy' examples, and that there is thus no evidence that the probe makes use of linguistic knowledge in the probed representations. On the other hand, similar scores in the two setups indicate that we have chosen a test set that is hard even for interpolation learning, in which case we do not want to draw this conclusion.

For comparison, we also report the mean accuracy in the *standard setup*, where we train and evaluate on the full datasets.

4 Results

We now present our experimental results for each of the scoring functions.

4.1 Sentence Length

The results for sentence length can be seen in Figure 1. The accuracies for the extrapolation setups are the highest among all scoring functions, and the differences to the standard setups are by far the smallest. Indeed, for part-of-speech tagging (T1) the difference is so small that a large part of it can probably be explained by the decreased number of training examples: the difference between the control and the extrapolation setup is mostly 1–2 points, and never exceeds 3 points. For dependency labelling (T2), the difference is more pronounced, but sentence length remains the measure with the smallest difference between the two setups.

The distributional split criterion gives $m_1 = 23$ and $m_2 = 34$, so both the longest sentences in the 'easy' set and the shortest sentences in the 'hard' set are longer than with the linguistic criterion. The linguistically motivated split shows a larger gap between the standard setups and the extrapolation setups. This is particularly clear for T2, with a gap as high as 8 points in layer 10.

4.2 Arc Length

The extrapolation accuracy of the probe based on arc length (Figure 2) is comparatively low, suggesting that this setup is more challenging than extrapolation based on sentence length. The control shows that the 'hard' set is clearly harder than the unfiltered test set; but there is an additional substantial accuracy drop in the extrapolation setup.

When using the distributional split criterion, we get $m_1 = 2$ and $m_2 = 4$, and the extrapolation accuracy does not exceed 46% in any layer. As $m_1 = 2$ results in a training set that only consists of arcs of length 1, we perform an additional experiment with a different split, decreasing the distance between $\mathcal{D}_{\text{easy}}$ and $\mathcal{D}_{\text{hard}}$ by setting $m_1 = 3$. This increases accuracy to at most 62%, which is considerably higher than before but still far below the control, which reaches up to 85% on the 'hard' set.

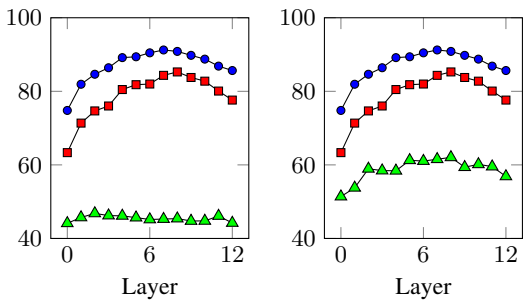

Figure 2: Extrapolation based on arc length. Left: Standard distributional setup. Right: Modified setup.

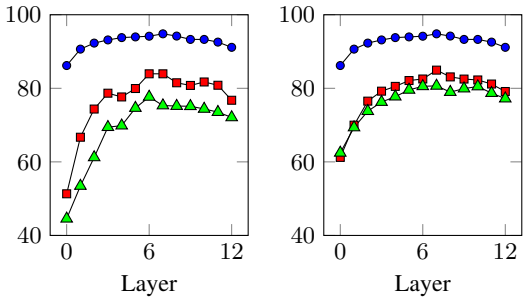

Figure 3: Extrapolation for T1 based on the most frequent tag (left) and tag proportions criteria (right).

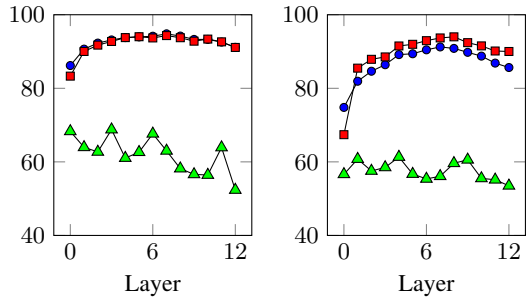

Figure 4: Extrapolation based on speed of learning. Left: Tagging (T1). Right: Dependency labelling (T2).

4.3 Most Frequent Tag and Tag Proportions

The results for the extrapolation splits based on the most frequent tag and the tag proportions criteria are shown in Figure 3.

Splitting based on the most frequent tag criterion leads to an extrapolation setup that is consistently more challenging than the standard setup. We observe a very low accuracy in the first layers, while the higher layers are significantly more predictive. The relative difference in accuracy between the extrapolation setup and the control is also most pronounced in the early layers, although the pattern is less clear in terms of absolute numbers. The gap to the standard (interpolation) setup is substantial: 11–35 points for the control, and 23–42 points for the extrapolation setup.

When using the tag proportions criterion for the extrapolation split, the 'hard' set is now easier, as around half of the examples have a tag that is the most frequent one for the word form. The simpler nature of this challenge is visible in the results: While the performance of the control only sees a modest increase (especially in the lower layers), the difference between the control and the extrapolation setup shrinks more clearly, presumably because the augmentation of the test set with easier examples has a high proportional effect on the previously very low results of the extrapolation probe.

4.4 Speed of Learning

Using the learning-based scoring function, the difference between the control and the extrapolation setup is the largest among all settings. The accuracy of the control is similar to that in the standard setup, suggesting that the 'hard' set may in fact not be (much) harder after all. For the dependency labelling task (T2), control accuracy even slightly *exceeds* accuracy on the standard set, in all layers but the uncontextualised layer 0.

Training the probe on the 'easy' set only, however, has a disastrous effect: in the extrapolation setup, the accuracy drops dramatically. Interestingly, accuracy continues to decrease in higher layers, whereas the typical curve for syntactic probes peaks in the middle layers (Tenney et al., 2019).

4.5 Sample-specific Loss

With the loss-based split (Figure 5), the results for the control setups are the lowest among all scoring functions. In this setting, by construction, the 'hard' set consists of the examples with the highest loss, making it challenging even in an interpolation setting. For the tagging task, we see an extreme drop of accuracy in layers 6–8, the layers on which the other two setups perform best.[3] The probes in these layers appear to be completely unable to extrapolate to the harder examples.

While for POS tagging (T1), extrapolation accuracy is generally very close to that of the control, for dependency labelling (T2) we observe a larger distance between all setups, but in particular between the extrapolation setup and the control.

[3]To put this into context, we recall that we tried to control for a too high model-specificness by averaging the losses of 10 different models.

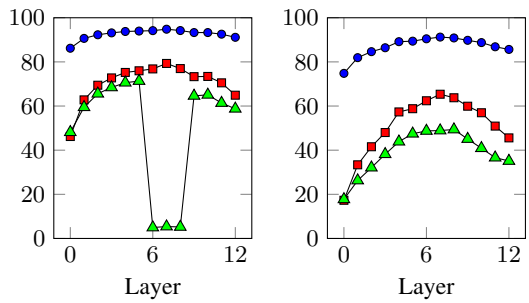

Figure 5: Extrapolation based on loss. Left: Tagging (T1). Right: Dependency Labelling (T2).

5 Discussion

While for all experiments, the accuracy in the extrapolation setup is substantially lower than in the standard interpolation setup, it is still above random guessing, which suggests that probes are able to extract *some* useful information from the word representations even under this experimental regime. However, the success of extrapolating from 'easy' to 'hard' examples varies depending on the choice of the scoring function. In this section we discuss these findings and the limitations of our method.

5.1 Scoring Functions

We start by arguing for the merits of the different scoring functions in the context of probing.

Sentence length Sentence length is the least discriminating metric in the experimental results, which is in line with our expectations and with previous work on curriculum learning discussed in § 2.3: for word-level tasks, sentence length is not a strong indicator for hard examples. In the case of part-of-speech tagging (T1), there is no considerable difference between interpolation and extrapolation accuracy. For dependency labelling (T2), such a difference is present; but it is small compared to other choices of scoring functions. The non-correlation between sentence length and hardness is quite intuitive: long sentences also contain many simple examples, and even short sentences may contain complex syntactic constructions. At the same time, the observed difference between the two tasks suggests that the higher-level the task is, and the wider the context it depends on, the more meaningful sentence length can be as a criterion for creating extrapolation challenges.

Arc length In contrast to sentence length, arc length provides a useful criterion for extrapolation. While the comparatively low accuracy in the control setup shows that longer arcs are a challenge in themselves, restricting the training set to short arcs limits the accuracy of the probe even further. Extrapolation capability is limited even in the softened setup where we decrease the distance between training and test set (Figure 2, right). Thus, under this scoring function, we find no evidence that the probe extracts useful linguistic knowledge from the word representations – a conclusion that establishes a difference between our extrapolation setup and results for interpolation-based learning (Tenney et al., 2020; Hewitt and Liang, 2019).

Most frequent tag and tag proportions Even the most frequent tag criterion is an informative setup. Our empirical results (Figure 3) suggest that the probe seems to heavily rely on word form-specific information at least in the first layers, while it focuses on more generalisable information in the later layers, and thus exhibits better extrapolation capabilities.

Based on the differences between the extrapolation setup and the control, we argue that the most frequent tag criterion is better motivated and provides more insights than tag proportions.

Speed of learning The 'speed of learning' criterion creates a very challenging extrapolation setup, compared to the standard setup (and the control). Probes trained on the full set perform well on the supposedly hard extrapolation test set, sometimes even better than on the standard test set. It is the training set that makes the difference: by only including fast-success examples, we are likely to miss patterns. The extrapolation setup favours patterns that are easy to learn, making it superfluous for the classifier to try harder and extract features that generalise better, even if these may not necessarily be extremely hard to learn – the number of such examples may simply be too small to learn the pattern in the first phase. As a consequence of this behaviour, the speed of learning criterion has a low interpretability. Without further qualitative analyses, we can only make assumptions about the nature of the 'easy' and 'hard' datasets, and in particular about the examples that are left out from either. And obviously, if patterns are completely missed, we cannot expect the model to extrapolate to harder examples of this very pattern.

Sample-specific loss The most opaque of all scoring functions is arguably the loss-based criterion. It is even less transparent than the learning-based criterion, where we can possibly identify the learned (and missed) patterns in an error analysis. With the loss-based criterion, we will be unlikely to identify commonalities between examples that share the same ranking with respect to the scoring function. While the loss-based criterion strongly discriminates between the standard setup and the extrapolation setup, this is largely an effect of the construction of the test set, which in the latter setup will contain all examples that are classified incorrectly. For tasks where the performance of a standard probe is already low, the test set will solely

consist of misclassified examples. Applying the loss of fully trained probes on the test set can therefore be seen as circular.

Summary To summarise, from a perspective of transparency, controllability, and demonstrable success in separating the data into easier and harder examples, we argue that the most interesting metrics for the identification of extrapolation challenges are arc length and the most frequent tag criterion. The learning-based scoring functions, which have the potential to be less ad-hoc, are hard to interpret, give unsurprising results, and are therefore less useful as an analysis tool.

Another benefit of arc length and the most frequent tag criterion is that they are applicable to a wide range of tasks. The most frequent tag criterion can be applied to any word labelling task that has a limited number of labels. Examples for further tasks where it can be applied include named entity recognition and word sense disambiguation. Arc length can be applied to all tasks that can be formulated as operating on pairs of words. Besides other parsing tasks such as semantic dependency parsing, this is the case for e.g. coreference resolution or negation scope detection (Kurtz et al., 2020).

5.2 Contributions and Limitations

The strong differences between the standard setup and the extrapolation setup and the great variability of results across scoring functions illustrate that the interpretation of probing classifiers remains challenging. A more extensive analysis, be it with automated techniques such as our extrapolation splits or with a qualitative analysis, is a necessity for a deeper understanding of a probing classifier's results. Unlike previous restrictions of the model or the training data as proposed by Hewitt and Liang (2019), our approach offers (given an appropriate scoring function, such as arc length or the most frequent tag criterion), more control over and transparency about the nature of the restrictions imposed by the modification of the data.

While the extrapolation setup helps approximating the nature of the features the probe uses, it does not ultimately solve the problem of the lacking interpretability of probing classifiers themselves. Negative results in the extrapolation setup do not imply that the linguistic knowledge of interest is *not* present in the representation. The probe may just have focused on other features – the amount of predictors to approximate a given target function is infinite. However, classical interpolation-based setups using probing classifiers tend to overestimate the information present in the representations, as classifiers can learn a task even from randomly initialised word embeddings (Zhang and Bowman, 2018; Hewitt and Liang, 2019). Therefore we argue that, at this time, we need to be more aware of false positives than of false negatives in probing. Extrapolation probes have the potential to reduce the false positive rate while providing new insights into the generalisability of the features they use.

6 Conclusion

We identified and suggested several ways to define the difficulty of training and validation examples based on linguistic, statistical, and learning-based criteria, to create extrapolation splits for natural language datasets. We demonstrated the usefulness of these measures for the analysis of two linguistic tasks, and proposed an evaluation protocol with baselines and metrics.

Our experimental results suggest that a probe trained on BERT hidden representations is capable of applying patterns learned from easier examples to harder examples to some extent; but in well-motivated scenarios where the scoring function is an appropriate measure of difficulty of the examples, its competence is clearly limited compared to an interpolation probe. In our experiments, the most informative scoring functions are the distance-based arc length criterion that we applied to syntactic dependency labelling, and the word-specific most frequent tag criterion for part-of-speech tagging. These functions allow for a clear and transparent extrapolation setup, while at the same time being simple and also computationally efficient. Sentence length, as expected, did not turn out to be a strong indicator for hard examples, while learning-based criteria show a high margin between interpolation and extrapolation setups, but limited interpretability and qualitative insights.

We conclude that enriching probing experiments with automated extrapolation setups can be a valuable supplement to standard probing methods, as it gives us an instrument to test the generalisation capability of the probe, and thereby the robustness of the features it uses. In addition to interpretation purposes, well-chosen extrapolation splits can provide a cheap but valuable extension of the evaluation of a model, testing its generalisation capabilities and verifying the progress made.

References

Alan Bailin and Ann Grafstein. 2001. The linguistic assumptions underlying readability formulae: A critique. *Language & Communication*, 21(3):285–301.

David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. 2018. Measuring abstract reasoning in neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 511–520. PMLR.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yann Dubois, Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. 2020. Location Attention for Extrapolation to Longer Sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 403–413, Online. Association for Computational Linguistics.

Rudolf Flesch and Alan J Gould. 1949. *The art of readable writing*, volume 8. Harper New York.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. 3rd International Conference for Learning Representations.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2755–2768, Hong Kong, China. Association for Computational Linguistics.

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Jenny Kunz and Marco Kuhlmann. 2020. Classifier probes may just learn from linear context features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5136–5146, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online. Association for Computational Linguistics.

John P. Lalor and Hong Yu. 2020. Dynamic data selection for curriculum learning via ability estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 545–555, Online. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131, Prague, Czech Republic. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229, Online. Association for Computational Linguistics.

Lucius Adelno Sherman. 1893. *Analytics of literature: A manual for the objective study of English prose and poetry*. Ginn.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 107–118, Online. Association for Computational Linguistics.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, Melbourne, Australia. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mostafa Zamanian and Pooneh Heydari. 2012. Readability of texts: State of the art. *Theory & Practice in Language Studies*, 2(1).

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? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.