Measuring Memorization Effect in Word-Level Neural Networks Probing[⋆]

Rudolf Rosa[0000−0003−4908−6127], Tomáš Musil[0000−0002−4013−560X], and David Mareček[0000−0001−5327−488X]

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Malostranské náměstí 25, 118 00 Praha, Czechia
{rosa,musil,marecek}@ufal.mff.cuni.cz
https://ufal.mff.cuni.cz/

Abstract. Multiple studies have probed representations emerging in neural networks trained for end-to-end NLP tasks and examined what word-level linguistic information may be encoded in the representations. In classical probing, a classifier is trained on the representations to extract the target linguistic information. However, there is a threat of the classifier simply memorizing the linguistic labels for individual words, instead of extracting the linguistic abstractions from the representations, thus reporting false positive results. While considerable efforts have been made to minimize the memorization problem, the task of actually measuring the amount of memorization happening in the classifier has been understudied so far. In our work, we propose a simple general method for measuring the memorization effect, based on a symmetric selection of comparable sets of test words seen versus unseen in training. Our method can be used to explicitly quantify the amount of memorization happening in a probing setup, so that an adequate setup can be chosen and the results of the probing can be interpreted with a reliability estimate. We exemplify this by showcasing our method on a case study of probing for part of speech in a trained neural machine translation encoder.

Keywords: probing · memorization · neural networks.

1 Introduction

In recent years, there has been a considerable amount of research into linguistic abstractions emerging in neural networks trained for various natural language processing (NLP) tasks. It has been found that, to some degree, neural networks often capture abstractions which seem to correspond to classical linguistic notions known from the linguistic studies of morphology, syntax or semantics, even if they were not explicitly trained to do so. The common hypothesis is that modern neural networks are sufficiently powerful to unravel many linguistic properties and regularities of language, and that they do so if this is useful for solving the task for which they are trained.

[⋆] This work has been supported by grant 18-02196S of the Czech Science Foundation. It has been using language resources and tools developed, stored and distributed by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101)

In this work, we focus on the subfield of identifying word-level linguistic abstractions, such as part-of-speech (POS) labels, in word-level representations, such as static or contextual word embeddings.

The usual method of assessing the amount to which linguistic abstractions are captured by a neural network is to use *probing*, which we review in Section 2. In word-level probing, we take representations of words from a trained neural network (such as word embeddings or hidden states from an encoder) and train a classifier to predict linguistic labels (such as POS) from the representations corresponding to the words, using linguistically annotated data (such as a tagged corpus). The common assumption is that if the classifier learns to predict the linguistic labels with a high accuracy, it is an indication that the neural word representations contain a latent abstraction similar to the linguistic notion (e.g. that contextual word embeddings encode POS of the words).

1.1 The Memorization Problem

A major threat associated with the probing approach is that of *memorization*. As the probing classifier learns to assign labels to words, it can succeed in two ways. Either, it learns to extract an abstraction from the word representation which corresponds to the label to assign; this is the intended case, which we refer to as *generalization*. Or, it simply memorizes the label associated with each word; we refer to this as *memorization*. If memorization occurs, the result of the probing can be misinterpreted as the representations capturing some linguistic abstractions, while the actual underlying mechanism is that the representations simply capture the word identity. The probing classifier thus only learns to extract the word identity from the representation and memorizes the label for the word.[1] A crucial problem is that, without taking additional measures, there is no way of distinguishing the true positive result from the false positive result.

With context-independent word representations (static word embeddings), it is of course possible to avoid the problem by splitting the vocabulary into two disjoint sets of words, training the classifier on a train set and testing it on a test set. However, for contextual representations, this cannot be done easily, as the representations need to be computed for whole sentences, not for individual words, and the train and test sets thus need to be composed of full sentences, which unavoidably have a large word overlap. While we might evaluate the probe only on test set words unseen in the training data, these are not representative of the language, as such a set of test words will be biased towards low-frequency words. We argue that we rather need to evaluate on the full test set while measuring and minimizing the memorization effect.

1.2 Measuring Memorization

In this paper, we suggest a general method of measuring the amount of memorization occurring in word-level probing of neural network representations, based on comparing

[1] Unlike static word embeddings, contextual representations of the same word in different sentences are different, which makes memorization harder, but not impossible: the identity of the word is still strongly encoded in the contextual representation and can be extracted from it, especially when a stronger classifier is used.

the probing classifier accuracy on sets of *seen* and *unseen* words. Although a standard test set contains both words seen and unseen in training data, the seen words tend to be frequent while the unseen ones are typically rare words; we thus regard an approach of comparing accuracies on these sets of words as inadequate and uninformative. Instead, we propose a method which samples the seen and unseen words in a symmetric way to ensure their comparability.

We do not present a new method for probing itself; our method is designed to complement existing probing approaches by explicitly measuring their reliability with respect to the memorization problem. This can help the researcher to select an adequate probing setup by providing means for quantifying the magnitude of the memorization problem, allowing for a trustworthy interpretation of the probing results.

As a case study, we apply our method to measure the amount of memorization in probing for POS in word representations from a neural machine translation system.

2 Related Work

A comprehensive survey of word embeddings evaluation methods was compiled by Bakarov [2]. An overview can also be found in the survey of methodology for analysis of deep learning models for NLP by Belinkov and Glass [4]. Another overview [12] mentions “[n]o standardized splits & overfitting” as one of the problems of evaluating word embeddings with similarity tasks.

There are various strategies when it comes to the train/dev/test splitting in probing.

When it is possible to predict the probed property from the word type itself, the vocabulary may be split into train/test sets. This strategy is used e.g. in [21,19] to evaluate POS tag and other morphological features prediction.

Some works split the dataset into train/dev/test sets, without regard to the same words occuring in both. These include predicting syntactic and semantic labels (including POS) from hidden states on sentences [22,3,5,11,18] or treebanks [7,15].

Bisazza and Tump [6] address the problem with the overlap. They observe that even a dummy random feature can be predicted with high accuracy when the same words occur both in the train and the test data. They extract one vector per token from the NMT encoder. They randomly split the vocabulary into two parts and use one to filter the training data and the other to filter the test data. They repeat the experiments several times and report mean accuracies.

Another approach to evaluating words in context of sentences is presented by [10]. They propose the word content task that tests whether it is possible to recover information about the original words in the sentence from its embedding. They pick 1000 mid-frequency words from the source corpus vocabulary and sample equal numbers of sentences that contain one and only one of these words. The words can then be partitioned into train and test sets without the risk of their overlapping.

The ability of deep neural networks to memorize is a challenge for the theory of deep learning [1]. It also has implications for the applications of neural networks, because it may be problematic if a portion of the training data can be reconstructed from the trained model [9].

In connection with probing neural networks, memorization was addressed by Hewitt and Liang [14], who propose control tasks to complement the linguistics tasks. A control task associates word types with random labels. If the classifier performs well on the control task, this means that it is able to memorize the training set. However, the data distribution affects the generalization ability of deep neural networks and they tend to learn simple patterns when possible [16]. Our approach differs from [14] by using the original data to measure the memorization effect, evading the problem created by altering the distribution in a control task.

3 Method

In the usual probing approach, we operate with two sets of sentences, a training set and a test set, both labelled with the word-level labels corresponding to the linguistic abstraction for which we are probing the neural word representations (e.g. POS). The training set is used to train a probing classifier to predict the labels from the word representations. The classifier is then evaluated on the test set, and its accuracy, compared to a baseline, is used to estimate to what extent the given linguistic abstraction is encoded in the word representations.

The goal of our method is to measure to what extent the probing classifier only memorizes word identities instead of measuring the generalization captured by the word representations. The main idea is to compare the probing classifier accuracies on words that are part of the training data (*seen* words) and on words that are not (*unseen* words), while keeping the sets of seen and unseen words otherwise comparable (as discussed in Section 1), which we ensure by a symmetric way of creating these sets.

We propose the following approach:

1. Randomly split the training set into two halves, which we will refer to as *seen sentences* and *unseen sentences*.
2. Train the probing classifier only on the seen sentences.
3. Apply the probing classifier to the test set.
4. Define the set of *seen words* as words that are contained in the seen sentences but not in the unseen sentences.
5. Define the set of *unseen words* as words that are contained in the unseen sentences but not in the seen sentences.
6. Evaluate the accuracy of the probing classifier separately on seen words and on unseen words, ignoring words that are neither seen nor unseen.[2]

Using this approach, we can now quantify the magnitude of the memorization effect occurring in the probing setup as the difference between the classifier accuracy on seen and on unseen words. If the memorization problem is not present, these accuracies should be identical, as the classifier only extracts the linguistic abstraction from

[2] Note that words which occur in both seen and unseen sentences are neither seen words nor unseen words. We also need to remove words that are part of the development set if one is used for training the probing classifier. Technically, words that do not appear in the test set can also be removed from the sets of seen and unseen words as they do not influence the results.

the representation, regardless of the word identity; in this case, the classifier accuracy reliably measures the amount of linguistic information encoded by the representation. On the other hand, a higher accuracy on seen words than on unseen words signalizes that the classifier memorized some of the seen words' identities to some extent, instead of extracting the linguistic abstractions from them.

To stabilize the evaluation, we propose to sample the seen and unseen sentences and train the classifier multiple times, and to compute the microaverage accuracy.

We define our method as operating on words and word representations, as this makes the subsequent word-level probing straightforward. Our method is in principle applicable even for setups using subwords. However, in such cases, it is up to the researcher to decide whether for the given language and setup, subword-level memorization is a problem or not, as our method only deals with word-level memorization.

3.1 Which Words Are Selected for Evaluation?

It is important to note that the distribution of words selected for evaluation by our method is strongly biased towards lower-frequency words. Very frequent words are never selected for evaluation, and medium-frequency words are rarely selected, as they always or nearly always appear in both seen and unseen sentences, and our method is thus unable to measure the memorization effect for such words.

Specifically, the probability $P_{sel}(w)$ of a word w being selected as *unseen* (or *seen*) follows a hypergeometric distribution: $P_{sel}(w) \sim \text{Hypergeometric}\left(|S|, \frac{|S|}{2}, |S_w|\right)$, where S is the set of training sentences, out of which its subset S_w contains the word w. For most words,[3] it is similar to the binomial distribution $\text{Bi}(|S_w|, 0.5)$, and $P_{sel}(w)$ is thus inversely exponentially proportional to $|S_w|$: $P_{sel}(w) \approx \left(\frac{1}{2}\right)^{|S_w|}$.

We believe that for **very frequent words** (especially function words such as common prepositions, pronouns, determiners and punctuation), avoiding memorization is hard – a set of sentences constructed not to contain a given word from this class would typically not be very representative of the language. Moreover, the probed neural network is typically not very likely to meaningfully abstract over such words, as it is usually more economical for the network to simply memorize the most frequent words and treat them as special cases.[4,5]

For **medium-frequency words**, such as common nouns and verbs, we see their underrepresentation as a shortcoming of our method which we intend to focus on in future work. We specifically plan to further investigate the approach of Bisazza and Tump [6], reviewed in Section 2, who train the probing classifier on representations of only some words in the training sentences and regard the other words as *unseen*. We appreciate the approach, but we believe that it must be analyzed to what degree

[3] For frequent words, the actual probability is even lower than the (already negligible) approximated value; for words that appear in more than half of the training sentences, the probability is 0. The probability is also technically 0 for words that do not appear in the test set.

[4] Which they often are, as frequent words tend to behave irregularly in language [23, p. 116].

[5] Arguably, it is sane to memorize very frequent words rather than abstracting over them. Nevertheless, we should be able to measure this reliably, not mistaking one for the other.

	Accuracy			Stand. dev.			Accuracy			Stand. dev.	
Train sent.	seen	unseen	diff	seen	unseen	Train sent.	seen	unseen	diff	seen	unseen
Encoder output states, linear classifier						Encoder word embeddings, linear classifier					
50	90.5	87.3	3.3	3.4	5.6	50	98.5	74.3	24.1	0.9	7.6
100	89.1	86.8	2.3	1.8	2.0	100	97.0	78.0	19.0	0.8	2.3
500	93.9	92.8	1.1	0.9	1.1	500	97.6	80.5	17.1	0.7	3.2
1,000	94.7	93.9	0.8	0.9	0.8	1,000	97.0	82.8	14.2	1.0	1.5
5,000	95.5	94.9	0.7	0.5	0.6	5,000	96.2	84.7	11.4	0.5	1.7
10,000	95.7	95.5	0.2	0.8	0.8	10,000	95.2	85.3	10.0	0.8	1.0
30,000	95.8	95.9	0.0	0.4	0.4	30,000	93.5	88.0	5.4	0.6	1.3
Encoder output states, MLP						Encoder word embeddings, MLP					
50	97.7	93.3	4.4	1.5	3.2	50	98.5	76.6	21.8	0.9	6.9
100	96.2	93.6	2.7	1.0	1.4	100	97.0	81.4	15.6	0.7	3.0
500	97.2	94.5	2.7	0.3	0.9	500	97.8	87.4	10.3	0.4	1.9
1,000	96.8	94.9	1.9	0.7	0.7	1,000	97.7	89.8	7.9	0.5	1.4
5,000	97.6	95.7	1.9	0.4	0.5	5,000	98.4	92.7	5.6	0.2	1.0
10,000	98.0	96.2	1.8	0.7	0.7	10,000	98.7	93.5	5.2	0.2	1.0
30,000	97.7	96.1	1.6	0.6	0.7	30,000	98.4	94.2	4.1	0.6	1.2

Table 1. Case study evaluation on POS prediction, varying the number of training sentences, the probed representations, and the probing classifier. The difference between the accuracy of the probe on seen versus unseen words represents the magnitude of the memorization problem. Micro-average over 10 repetitions, in percentage points, with standard deviations.

it may be influenced by the contextual representations of the *seen* words containing information about surrounding words regarded as *unseen*.[6]

Our method mostly focuses on **lower-frequency words**, which we believe to be reasonable, as the lower the frequency of the word, the stronger is the network forced to abstract over the word. We are thus mostly interested in such words in probing, as if the network captures the abstractions that we are probing for, they should be most prominent in representations of lower-frequency words.

Still, we also omit **very rare words**, which either do not appear in the test sentences or in the training sentences (or, obviously, in none of those). For these words, the memorization effect is very unlikely to occur.

4 Case Study

As a case study, we apply our method to probing representations from a neural machine translation model for POS. We study the memorization phenomenon along three dimensions, varying the train set size, the contextuality of the representation (static word

[6] In their method, *unseen* words are part of the training sentences and can thus influence the contextual representations of the *seen* words which are used for training the probing classifier, whereas in our method, the training sentences do not contain the *unseen* words at all.

embeddings versus encoder output states), and the power of the probing classifier, using either a linear classifier or a multi-layer perceptron classifier (MLP).

We analyze a Transformer model [24] implemented within the Neural Monkey framework[7] [13], trained for the task of machine translation from Czech to English on the CzEng dataset[8] [8]. The setup is based on [17], with the exception of splitting the sentences into words instead of subwords, as explained in Section 3; we use a vocabulary of 25,000 words that are most frequent in the parallel training data.

We probe the source word embeddings and source encoder output states for Universal POS with a linear classifier (softmax) or a MLP with one hidden layer of dimension 512, using the Universal Dependencies 1.4 version of the Czech Prague Dependency Treebank [20]. We use the first 500 sentences from the treebank training data as tuning data for the probing classifier, the rest of the training data is used to create the seen and unseen sentence sets, using either the full data or subsampling smaller subsets. The probing classifier is then evaluated using the development part of the treebank using token-based evaluation. For each setup, we repeat the experiment 10 times with different samples of the seen and unseen sentences and report micro-average results.

By comparing the accuracies of the probing classifier on seen and unseen words in Table 1, we can see that the memorization problem is clearly most pronounced with static word embeddings, where the magnitude of the effect (the difference in the accuracies) ranges from 4 points for the full training set up to 24 points for a training set of 50 sentences, while for the contextual representations, the effect does not surpass 5 points. The memorization effect is more pronounced with the stronger classifier, and disappears only with the linear classifier applied to contextual representations when trained with the largest train set.

5 Conclusion

We presented a method for measuring the memorization effect in word-level probing of neural representations of words, based on a comparison of the accuracy of the probing classifier on symmetrically sampled comparable sets of *seen* and *unseen* words. As we showed in a case study on probing for POS, our method can measure the magnitude of the memorization problem and can thus serve as a means for selecting an appropriate probing setup, as well as for estimating the reliability of the findings of the probing experiment with respect to the threat of mistaking memorization for generalization.

In future, we intend to tackle the shortcoming of our method of underrepresenting medium-frequency words. We also plan to apply the method to a wider range of word-based probing tasks, as well as to measure the memorization effect for existing previous probing works and reassess results reported by their authors from this perspective.

References

1. Arpit, D., Jastrzundefinedbski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., et al.: A closer look at memorization in deep networks. In: Proc. ICML. pp. 233–242 (2017)

[7] https://github.com/ufal/neuralmonkey

[8] http://ufal.mff.cuni.cz/czeng

2. Bakarov, A.: A survey of word embeddings evaluation methods. arXiv preprint arXiv:1801.09536 (2018)
3. Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., Glass, J.: What do neural machine translation models learn about morphology? In: Proc. ACL. pp. 861–872. Vancouver, Canada (2017)
4. Belinkov, Y., Glass, J.: Analysis methods in neural language processing: A survey. TACL **7**, 49–72 (2019)
5. Belinkov, Y., Màrquez, L., Sajjad, H., Durrani, N., Dalvi, F., Glass, J.: Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In: Proc. IJCNLP. pp. 1–10. Taipei, Taiwan (2017)
6. Bisazza, A., Tump, C.: The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation. In: Proc. EMNLP. pp. 2871–2876. Brussels, Belgium (2018)
7. Blevins, T., Levy, O., Zettlemoyer, L.: Deep RNNs encode soft hierarchical syntax. In: Proc. ACL. pp. 14–19. Melbourne, Australia (2018)
8. Bojar, O., Dušek, O., Kocmi, T., Libovický, J., Novák, M., Popel, M., et al.: Czeng 1.6: Enlarged Czech-English parallel corpus with processing tools dockered. In: Sojka, P., et al. (eds.) TSD. pp. 231–238. Springer International Publishing, Cham (2016)
9. Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D.: The secret sharer: Evaluating and testing unintended memorization in neural networks. In: USENIX Security. pp. 267–284. Santa Clara, CA (2019)
10. Conneau, A., Kruszewski, G., Lample, G., Barrault, L., Baroni, M.: What you can cram into a single vector: Probing sentence embeddings for linguistic properties. arXiv preprint arXiv:1805.01070 (2018)
11. Dalvi, F., Durrani, N., Sajjad, H., Belinkov, Y., Vogel, S.: Understanding and improving morphological learning in the neural machine translation decoder. In: Proc. IJCNLP. pp. 142–151. Taipei, Taiwan (2017)
12. Faruqui, M., Tsvetkov, Y., Rastogi, P., Dyer, C.: Problems with evaluation of word embeddings using word similarity tasks. arXiv preprint arXiv:1605.02276 (2016)
13. Helcl, J., Libovický, J., Kocmi, T., Musil, T., Cífka, O., Variš, D., et al.: Neural Monkey: The current state and beyond. In: Proc. AMTA. pp. 168–176. Boston, MA (2018)
14. Hewitt, J., Liang, P.: Designing and interpreting probes with control tasks. In: Proc. of EMNLP-IJCNLP. pp. 2733–2743. Hong Kong, China (2019)
15. Köhn, A.: What's in an embedding? Analyzing word embeddings through multilingual evaluation. In: Proc. EMNLP. pp. 2067–2073 (2015)
16. Krueger, D., Ballas, N., Jastrzebski, S., Arpit, D., Kanwal, M.S., Maharaj, T., et al.: Deep nets don't learn via memorization. In: Proc. of ICLR. p. 4 (2017)
17. Libovický, J., Rosa, R., Helcl, J., Popel, M.: Solving three Czech NLP tasks end-to-end with neural models. In: Proc. SloNLP (2018)
18. Linzen, T., Dupoux, E., Goldberg, Y.: Assessing the ability of LSTMs to learn syntax-sensitive dependencies. TACL **4**, 521–535 (2016)
19. Musil, T.: Examining structure of word embeddings with PCA. In: Ekštein, K. (ed.) TSD. pp. 211–223. Springer International Publishing, Cham (2019)
20. Nivre, J., et al.: Universal dependencies 1.4 (2016), LINDAT/CLARIAH-CZ digital library at ÚFAL MFF UK, Charles University
21. Qian, P., Qiu, X., Huang, X.: Investigating language universal and specific properties in word embeddings. In: Proc. ACL. pp. 1478–1488 (2016)
22. Shi, X., Padhi, I., Knight, K.: Does string-based neural MT learn source syntax? In: Proc. EMNLP. pp. 1526–1534. Austin, Texas (2016)
23. Stubbs, M.: Language corpora. In: Davies, A., Elder, C. (eds.) The handbook of applied linguistics, chap. 4, pp. 106–132. Blackwell Publishing, Malden, MA, USA (2004)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) NeurIPS 30, pp. 6000–6010. Curran Ass. (2017)