Reverse Probing: Evaluating Knowledge Transfer via Finetuned Task Embeddings for Coreference Resolution

Tatiana Anikina[1,2,*], **Arne Binder**[1,*], **David Harbecke**[1], **Stalin Varanasi**[1,2], **Leonhard Hennig**[1], **Simon Ostermann**[1,2], **Sebastian Möller**[1,3], and **Josef van Genabith**[1,2]
[1]German Research Centre for Artificial Intelligence, Saarbrücken
[2]Saarland Informatics Campus, Saarbrücken
[3]Technische Universität Berlin, Berlin
tatiana.anikina@dfki.de

Abstract

In this work, we reimagine classical probing to evaluate knowledge transfer from simple source to more complex target tasks. Instead of probing frozen representations from a complex source task on diverse simple target probing tasks (as usually done in probing), we explore the effectiveness of embeddings from multiple simple source tasks on a single target task. We select coreference resolution, a linguistically complex problem that requires contextual understanding, as the focus target task, and we test the usefulness of embeddings from comparably simpler tasks such as paraphrase detection, named entity recognition, and relation extraction. Through systematic experiments, we evaluate the impact of individual and combined task embeddings.

Our findings reveal that task embeddings vary significantly in utility for coreference resolution, with semantic similarity tasks (e.g., paraphrase detection) proving most beneficial. Additionally, representations from intermediate layers of fine-tuned models often outperform those from final layers. Combining embeddings from multiple tasks consistently improves performance, with attention-based aggregation yielding substantial gains. These insights shed light on the relationships between task-specific representations and their adaptability to complex downstream tasks, encouraging further exploration of embedding-level task transfer. Our source code is publicly available.[1]

1 Introduction

Language models have exhibited superior performance in most areas of NLP applications, including natural language inference (Williams et al., 2018), question answering (Rajpurkar et al., 2016, 2018), commonsense reasoning (Talmor et al., 2019; Ostermann et al., 2019), and others. Since the establishment of language models with partial superhuman performance, research has aimed to pinpoint which types of knowledge are exactly encoded by such language models. One technique in the field of explainable artificial intelligence for evaluating the presence of such types of knowledge is *probing* (Conneau et al., 2018; Hewitt and Liang, 2019; Tenney et al., 2019a; Belinkov, 2022). Probing involves adding linear classifiers on top of representations extracted from a pre-trained model, which are trained on simple tasks for predicting a

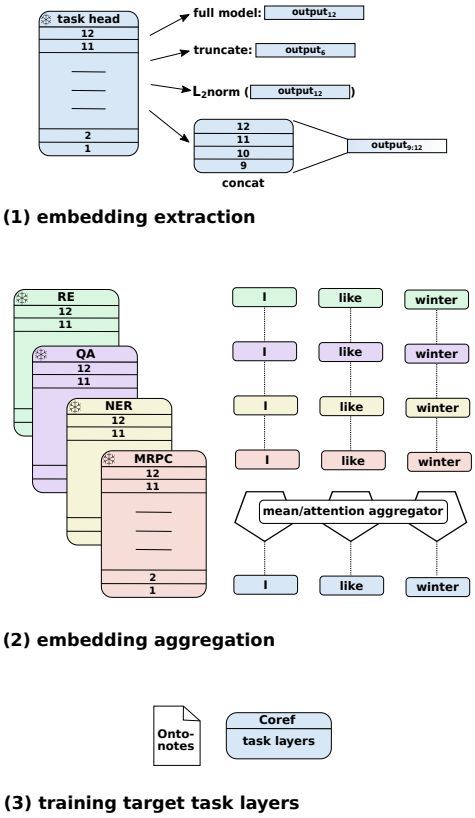


Figure 1: Probing workflow with Coreference Resolution (Coref) as target task and four different source tasks: Relation Extraction (RE), Question Answering (QA), Named Entity Recognition (NER), and Paraphrase Detection (MRPC).

[1]github.com/Cora4NLP/multi-task-knowledge-transfer
* Equal contribution.

Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025), pages 108–119

feature of choice, such as syntactic structures (Lin et al., 2019), entity types (Tenney et al., 2019b), or specific types of commonsense knowledge (Zhou et al., 2020).

A main intuition behind probing is to evaluate to what degree the representations that are learned from the complex source task can be re-purposed to solve a new, simpler task (Belinkov, 2022). In our work we decide to reverse this paradigm (thus *reverse probing*) and investigate how different source task embeddings, from a model trained on simple tasks, can be adapted to a new, more complex target task. In other words, we try to answer the question: **Can we reuse knowledge from simpler tasks for a more complex task?** Such a *recycling* of knowledge is not only interesting to deepen our understanding of what type of knowledge is encoded in language models, but it also results in more energy-efficient deep learning, by reusing network weights and representations.

We choose **coreference resolution** (Lee et al., 2017) as our target task because solving coreference is - up to date - a challenging NLP problem that even newer large language models struggle with (Bohnet et al., 2023; Martinelli et al., 2024). Coreference resolution involves understanding of context, what counts as a valid mention and which mentions refer to the same entity. Solving coreference may require different types of linguistic knowledge. Our goal is to find out which types of information from which source task models are useful and how this information can be combined and/or adapted to work for the target task.

To isolate the effects of single tasks, we rely on small language models, in our case BERT (Devlin et al., 2019). Such models do not possess sophisticated in-context abilities and require finetuning steps in order to perform well on tasks. Our research questions are as follows:

(1) Which **source tasks** are beneficial for combination into a more complex target task, here coreference resolution?

(2) Which **layers** of source task models contribute most to the target model performance?

(3) What are the effects of **combining embeddings** from different models and layers? How should these embeddings be aggregated? Can we improve word representations by extending the **embedding context** and combining the outputs of several hidden layers?

2 Reverse Probing

The goal of our framework is to evaluate the transferability of knowledge embedded in representations from simpler source tasks to a complex target task. Figure 1 gives an overview.

Let $S = \{s_1, s_2, \ldots, s_k\}$ be a set of source tasks with models pre-trained on simpler NLP tasks, and T be the target task (coreference resolution in this case). M_s is a pre-trained model fine-tuned on source task s. We then define H_s^l to be the output embeddings from layer l of M_s.

For each source task $s_i \in S$, we extract embeddings $\mathbf{H}_s$ from layer l of the corresponding source task model M_s (Figure 1, embedding extraction). We either take the output at a single or multiple consecutive layers. Note that these may be also intermediate layers (model truncation). Optionally, we apply L_2 normalization.

Secondly, we aggregate token embeddings from different source task models by using an aggregation function A to combine embeddings across layers and models. The aggregation is done token-wise, so that every token can be represented as a combination of different model outputs. We define A to be either the mean of all vectors, i.e. as

$$\mathbf{E}_T = \frac{1}{k} \sum_{i=1}^{k} \mathbf{H}_{s_i}$$

Alternative, we use a simple attention mechanism for the combination (Figure 1, embedding aggregation):

$$\mathbf{E}_T = \sum_{i=1}^{k} \alpha_i \mathbf{H}_{s_i}$$

where

$$\alpha_i = \text{softmax}(\mathbf{W} \cdot \mathbf{H}_{s_i})$$

In some experiments we use only a single model. In this case the mean corresponds to the original embedding of the source model and attention simply means self-attention.

Next, the aggregated token embeddings are passed to the target task head that includes several layers with trainable weights (Figure 1, training target task layers which follow the *coref-hoi* implementation by Xu and Choi (2020)).

Figure 1 shows the probing workflow with four different source task models. Each source model is pre-trained separately on a corresponding dataset as described in Section 3. The models are based on *bert-base-cased* contextualized embeddings with

different task-specific heads and their weights are frozen. Given that the source models cannot update their weights during probing, our conjecture is that those models that perform better on the coreference task "out-of-the-box" contain some useful information that is relevant for the target task.

3 Tasks

In this section we describe the target task, the source tasks and their respective training data.

3.1 Target Task

As our target model we choose a popular end-to-end coreference resolution model based on the implementation by (Xu and Choi, 2020) and train it on the OntoNotes CoNLL 2012 corpus (Pradhan et al., 2012). We use *bert-base-cased* and the recommended parameters for fine-tuning[2].

3.2 Source Tasks

We focus on the comparison with standard BERT as well as four other task-specific models. As source tasks we take a range of tasks of varying complexity: Paraphrase identification, named entity recognition, relation extraction, and - a (more complex) source task - quesion answering.

The first model is fine-tuned on the Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005). Since paraphrased sentences describe the same entities and events, such sentence pairs likely contain more coreferent mentions than standard (non-paraphrased) texts. Hence, MRPC embeddings are more tuned towards semantic similarity and could be useful for the coreference task.

Named Entity Recognition (NER) model is trained on the CoNLL 2012 dataset (Pradhan et al., 2012) and can generate one of the 37 labels for each token (e.g., PERSON, PRODUCT, DATE etc.). Named entities are often involved in coreference relations and being able to identify mention spans correctly is crucial for coreference resolution.

Next, we also experiment with the Relation Extraction model (RE) trained on the TACRED dataset (Zhang et al., 2017). It provides annotations for the spans of the subject and object mentions as well as the mention types according to the Stanford NER system and relations (if applicable) between the entities. Similarly to the NER model, RE is important for coreference because one of the tasks that this model performs is mention span detection. However, it also classifies different relations between the mentions and such relations are typically non-referential (e.g. *"per:employee_of"*).

Another source task model used in this project is trained on the SQUAD 2.0 dataset (Rajpurkar et al., 2016) for extractive question answering. This model (QA) can identify answer spans given the question and a paragraph of text. Since answering questions often involves coreference resolution, there is an overlap between these two tasks and word embeddings from one task might be beneficial for another.

For single model experiments we also analyse the performance on the vanilla BERT model [3](Devlin et al., 2019) which was trained with a masked language modeling objective on BookCorpus (Zhu et al., 2015) and English Wikipedia. Note that all the other source models are fine-tuned versions of this model.

Additionally, we experiment with the POS-tagging model[4], the models for semantic tagging[5] and chunking[6] as well as another NER model (NER-dslim)[7] trained on the English version of the CoNLL-2003 Named Entity Recognition dataset (Tjong Kim Sang and De Meulder, 2003). However, we limit the number of experiments for these models and focus mostly on MRPC, NER, RE and QA tasks.

4 Experiments and Results

In this section we describe our experiments with various source models and probe them on the coreference resolution task (§4.2). We also evaluate different embedding aggregation methods (§4.3), measure the effects of using intermediate layer output and normalization (§4.4), vary the embedding context from several hidden layers (§4.5) and compare the performance of multiple vs single models (§4.6).

[2] https://github.com/lxucs/coref-hoi/blob/master/experiments.conf

[3] We use the cased variant from HuggingFace under https://huggingface.co/bert-base-cased.

[4] https://huggingface.co/QCRI/bert-base-cased-pos

[5] https://huggingface.co/QCRI/bert-base-cased-sem

[6] https://huggingface.co/QCRI/bert-base-cased-chunking

[7] https://huggingface.co/dslim/bert-base-NER

4.1 Training Details and Evaluation

The coreference-specific layers are trained with the learning rate 1e-4 and early stopping (maximum number of epochs is set to 100 and patience is set to 5). The learning rate was optimized based on the experiments with the standard frozen BERT model.

For evaluation we use an average F1 score that is a combination of MUC (Vilain et al., 1995), CEAF (Luo, 2005) and B^3 (Bagga and Baldwin, 1998) coreference metrics. We run each experiment with three different seeds and report the average F1 values on the validation set. The target (non-frozen) model trained on the coreference resolution task from scratch achieves 73.75 F1 which is an upper bound for our probing task.

4.2 The Choice of the Source Task Models

Figure 2 shows the comparison between different source task models. Our original set of models that includes MRPC, NER, RE, QA and vanilla BERT has two clear winners: BERT and MRPC (64.01 and 64.32 F1). They are followed by RE (52.43) and QA (47.51) models and, finally, NER achieves the lowest score of 36.03. This comparison is based on a single run with the same seed, the averaged results across three runs with standard deviation can be found in Table 1.

We also have a closer look at the cosine similarity between our source models and the pre-trained coreference model. Figure 3 shows similarity scores averaged across all tokens for 15 random batches. The scores are collected before the embedding aggregation. Hence, they show how close the original source model embeddings are to the "ideal" task embeddings. Unsurprisingly, BERT and MRPC have the most similar embeddings to the coreference target. On the other hand, although both QA and NER embeddings are very different from the target task embeddings, QA achieves much better performance than NER on this task (50.79 vs 35.63 F1, see Table 1). This shows that even though cosine similarity is a good approximation for the task similarity, it is not an ideal predictor for the target task performance and even the source models with very different embeddings (QA) can still achieve the scores comparable to the ones achieved by the models with more similar embeddings (RE).

Additional models that we tested demonstrate rather poor performance on the coreference resolution task (see Figure 2). POS-tagging model struggles to learn anything about coreference and the training does not progress. Another NER model trained on a different version of Ontonotes (NER-dslim) achieves the maximum of 37.83 F1. Chunking and semantic labeling are somewhat more successful and achieve 48.81 and 49.82 F1 each, correspondingly.

4.3 How to Combine Task Embeddings

We employ two different aggregation strategies to combine the embeddings of the source task models: mean and attention-based aggregation. Additionally, we experimented with summing instead of using the mean, but the results were comparable or slightly worse: A combination of frozen MRPC with BERT achieves 62.34 F1 with sum and 63.26 with mean (average values across three runs with different seeds). Hence, in all further experiments we focus on the comparison between the mean and attention-based aggregation.

F1 scores for single models as well as for their 2x, 3x and 4x combinations can be found in Tables 1, 2 and 3. We also summarize the results for single models and for pairs of models graphically in Figure 4 that shows how much models benefit from attention. However, the trend holds even when there is only a single source model. This shows how much improvement we get by simply adding additional projections in the case of attention aggregation. The performance gains are different depending on the model. E.g., if we use pre-trained coreference model as our source task, there is almost no difference between attention and mean aggregation. However, other task-specific models can substantially benefit from selective aggregation. E.g., NER gains almost +19.7% and QA improves by +9%. In general, all models except for the one that has the same source and target tasks (CorefTarget) benefit from attention and improvements are larger for the models that have lower original scores.

For multiple model combinations we also see a similar trend with consistent improvements when attention-based aggregation is used, e.g., +13.23% for NER+MRPC and +9.36% for NER+QA (see Figure 4 and Table 2 for further comparisons). Interestingly, when we combine our source models with the model that was pre-trained on the coreference task (CorefTarget), we have only negligible improvements because the attention aggregator quickly learns which source model is beneficial for the task and starts paying almost all attention

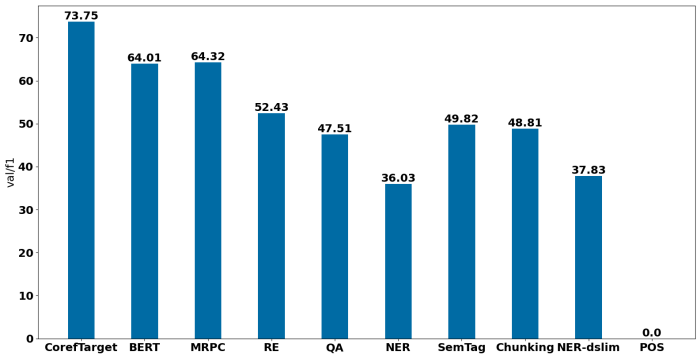


Figure 2: Source task models: CorefTarget, BERT, MRPC, RE, QA, NER, SemTag, Chunking, NER-dslim, POS

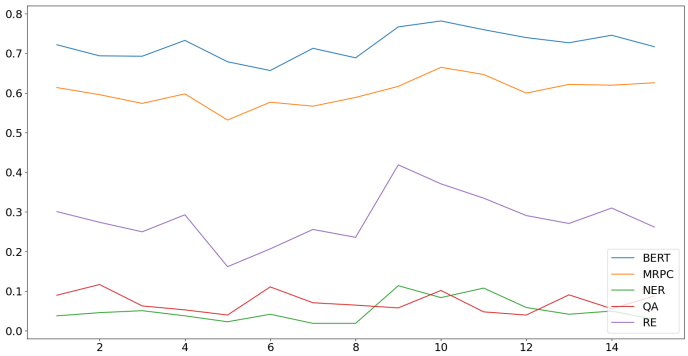


Figure 3: Average cosine similarity between the embeddings of the source tasks and the target coreference task, averaged across all tokens for 15 batches

to the output of this model ignoring all the others. However, if we do not add coreference task to the set of source tasks we observe some interesting patterns that emerge with the combinations of different models. Figure 9 (in the Appendix) shows how attention is distributed across different training epochs for the combination of MRPC, RE and NER. In the beginning, all three models are being paid the same amount of attention (≈33%). However, the aggregator soon starts prioritizing MRPC and NER gets progressively less and less attention. Interestingly, RE model also loses some impact over time but more slowly and remains somewhat important for the aggregator until the end of the training.

4.4 How to Extract Embeddings

We also consider different ways of embedding manipulations since the final layers of BERT-based models might be too specialized on their corresponding tasks, so that their representations are no longer useful for coreference resolution. In fact, after comparing the embeddings from layer 6 to 12 we found that the best performing layer on our probing task was typically not the final one. E.g., it was layer 9 for MRPC and RE, layer 8 for QA and 6 for NER (see Figure 5). Tables 1, 2 and 3 show the detailed comparisons between the original (full) model outputs as well as the normalized and truncated (to the "best" layer) versions for single models and their combinations (see also Figure 6 and 7 for the plot comparison). Truncation seems to be a good strategy for embedding aggregation and consistently yields best results across different settings. Truncation improves NER by up to +26.2%. QA is improved by +14% (Figure 6). Also combinations of models work better with truncation, e.g., RE+QA pair gains +8.17% F1 with mean aggregation and +2.84% with attention aggregation when both models are truncated (Figure 7).

Since combining embeddings from disparate models is a challenging task, especially when the

models	mean			attention			layer concat + attention		
	full	norm	trunc	full	norm	trunc	4	6	12
$MRPC_{(9)}$	$61.16_{\pm 2.84}$	$58.61_{\pm 13.7}$	$66.26_{\pm 0.61}$	$67.05_{\pm 0.70}$	$67.49_{\pm 0.29}$	$67.27_{\pm 0.30}$	**67.94**$_{\pm 1.54}$	$67.03_{\pm 0.99}$	$67.28_{\pm 1.48}$
$NER_{(6)}$	$35.63_{\pm 2.12}$	$47.95_{\pm 3.27}$	$61.76_{\pm 1.53}$	$55.30_{\pm 1.22}$	$54.82_{\pm 1.02}$	$64.31_{\pm 0.22}$	$63.80_{\pm 0.71}$	**66.30**$_{\pm 0.51}$	$65.76_{\pm 0.95}$
$RE_{(9)}$	$52.27_{\pm 2.39}$	$48.03_{\pm 10.8}$	$62.40_{\pm 1.41}$	$60.97_{\pm 0.01}$	$41.01_{\pm 0.14}$	$63.73_{\pm 0.70}$	$65.16_{\pm 1.07}$	**65.79**$_{\pm 0.37}$	$65.43_{\pm 0.93}$
$QA_{(8)}$	$50.79_{\pm 3.01}$	$59.56_{\pm 0.65}$	$64.77_{\pm 0.65}$	$59.82_{\pm 1.51}$	$60.98_{\pm 0.60}$	$66.47_{\pm 0.56}$	**67.70**$_{\pm 0.93}$	$66.82_{\pm 0.11}$	$67.63_{\pm 0.86}$
$BERT_{(10)}$	$64.95_{\pm 0.98}$	$66.50_{\pm 0.09}$	$66.40_{\pm 1.66}$	$67.15_{\pm 0.49}$	$43.94_{\pm 0.70}$	$68.19_{\pm 0.80}$	$69.06_{\pm 0.49}$	**69.07**$_{\pm 1.15}$	$68.79_{\pm 0.99}$
$Coref_{(12)}$	**73.75**$_{\pm 0.29}$	$72.33_{\pm 0.12}$	$73.75_{\pm 0.29}$	$73.60_{\pm 0.31}$	$73.11_{\pm 0.55}$	$73.60_{\pm 0.29}$	$73.19_{\pm 0.65}$	$72.70_{\pm 0.85}$	$72.59_{\pm 0.09}$

Table 1: Results for single models with different settings, mean and attention aggregation. Subscript indicates the best truncation layer for the *trunc* setting.

models	mean			attention		
	full	norm	trunc	full	norm	trunc
$RE_{(9)}$ + $MRPC_{(9)}$	$61.76_{\pm 1.85}$	$64.71_{\pm 0.42}$	$64.49_{\pm 0.76}$	$67.56_{\pm 0.36}$	$51.06_{\pm 13.8}$	**68.78**$_{\pm 0.32}$
$NER_{(6)}$ + $MRPC_{(9)}$	$54.71_{\pm 4.57}$	$64.47_{\pm 0.29}$	$65.71_{\pm 0.22}$	$67.94_{\pm 0.78}$	$67.36_{\pm 0.46}$	**68.67**$_{\pm 0.46}$
$RE_{(9)}$ + $NER_{(6)}$	$56.13_{\pm 1.42}$	$60.68_{\pm 0.48}$	$64.38_{\pm 1.14}$	$64.00_{\pm 0.74}$	$62.80_{\pm 0.36}$	**67.03**$_{\pm 0.31}$
$QA_{(8)}$ + $MRPC_{(9)}$	$60.84_{\pm 1.04}$	$65.58_{\pm 0.48}$	$66.46_{\pm 2.05}$	$67.87_{\pm 0.92}$	$59.00_{\pm 15.0}$	**68.98**$_{\pm 0.44}$
$RE_{(9)}$ + $QA_{(8)}$	$57.66_{\pm 3.61}$	$63.55_{\pm 0.26}$	$65.83_{\pm 0.61}$	$65.02_{\pm 0.39}$	$64.49_{\pm 0.30}$	**67.86**$_{\pm 0.59}$
$NER_{(6)}$ + $QA_{(8)}$	$55.66_{\pm 1.72}$	$61.14_{\pm 0.74}$	$65.06_{\pm 0.68}$	$65.02_{\pm 0.46}$	$62.51_{\pm 0.59}$	**67.65**$_{\pm 0.55}$

Table 2: Results for the pairs of models with different settings, mean and attention aggregation. Subscript indicates the best truncation layer for the *trunc* setting.

source tasks are quite different from each other, we also experiment with applying L_2 norm to the output of each model before aggregating the embeddings. This gives us varied results depending on the model and the aggregation type. E.g., for mean aggregation single NER, QA and BERT benefit from normalization but MRPC and RE result in lower scores. For attention aggregation all models except for MRPC and QA have substantial drop in performance.

It is also interesting to see the effect of normalization on the combinations of different models. For mean aggregation normalization brings substantial improvements, e.g., +9.76 F1 for NE+MRPC and +5.48% for NER+QA and, in general, all 2x models show better performance with normalization (Table 2). However, there is a very different trend for attention-based aggregation. Here we see a large drop in performance for most of the models, e.g., -8.87 F1 for QA+MRPC which indicates that attention can already combine the embeddings from different models quite well and achieves worse results with more uniform normalized embeddings.

4.5 Embedding Context from Multiple Layers

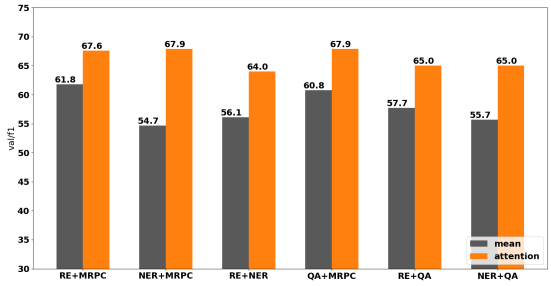
Since we found in our previous experiments that truncation consistently improves the performance for many source models, we decided to explore another setting that involves concatenating the embeddings of the last n hidden layers of the source model before aggregating them with attention. We experiment with the last 4, 6 and 12 layers and compare them to the aggregation that affects only the last layer of each model (see Table 1 for more detail).

Our results show that for single models having more "embedding context" is beneficial. Overall, combinations of the last 4 or 6 layers result in the best F1 scores. However, combining all layers of the model is not necessarily useful and can even hurt the performance. E.g., NER achieves 66.30 F1 with combined 6 layers which is +11 F1 improvement compared to the same model that uses only a single last layer but when we combine all 12 layers of NER the metric decreases from 66.30 to 65.76 F1 (Table 1).

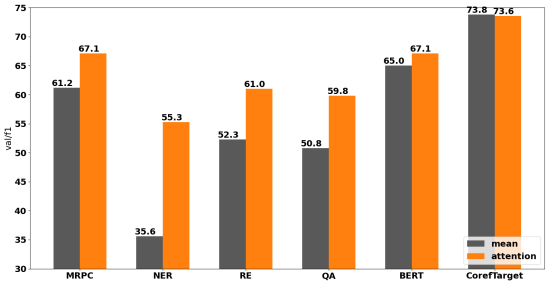
Another interesting observation is that for vanilla BERT combining the outputs of the last 4 or 6 layers does not make any difference, and for other models the difference is more pronounced, although still rather small. NER is the model that gains the most from the increased embedding context, it gains additional +2.5 F1 by combining 6 instead of 4 last layers which is also consistent with our finding for the truncated models and the fact that NER performs better when truncated to 6 layers. The only model that does not show any improvements in the layer concatenation setting is the coreference source model since it is already optimized for the task and performs best as it is, i.e., without truncation, normalization or any other embedding manipulations.

models	mean			attention		
	full	norm	trunc	full	norm	trunc
$RE_{(9)}$ + $NER_{(6)}$ + $QA_{(8)}$	$58.95_{\pm 1.11}$	$63.70_{\pm 0.08}$	$65.04_{\pm 0.71}$	$66.24_{\pm 0.14}$	$64.71_{\pm 0.50}$	$\mathbf{68.98}_{\pm 0.32}$
$MRPC_{(9)}$ + $NER_{(6)}$ + $QA_{(8)}$	$60.35_{\pm 1.42}$	$65.13_{\pm 0.65}$	$65.68_{\pm 0.83}$	$69.21_{\pm 0.17}$	$59.10_{\pm 14.1}$	$\mathbf{69.56}_{\pm 0.35}$
$MRPC_{(9)}$ + $RE_{(9)}$ + $QA_{(8)}$	$61.83_{\pm 0.38}$	$65.22_{\pm 0.20}$	$66.69_{\pm 0.64}$	$68.63_{\pm 0.60}$	$67.17_{\pm 0.21}$	$\mathbf{68.81}_{\pm 0.69}$
$MRPC_{(9)}$ + $RE_{(9)}$ + $NER_{(6)}$	$62.27_{\pm 2.08}$	$65.15_{\pm 0.18}$	$65.96_{\pm 0.52}$	$68.31_{\pm 0.10}$	$66.88_{\pm 0.16}$	$\mathbf{69.30}_{\pm 0.52}$
$MRPC_{(9)}$ + $RE_{(9)}$ + $NER_{(6)}$ + $QA_{(8)}$	$62.19_{\pm 1.49}$	$65.56_{\pm 0.11}$	$65.66_{\pm 0.50}$	$69.03_{\pm 0.45}$	$66.80_{\pm 0.53}$	$\mathbf{69.39}_{\pm 0.74}$

Table 3: Results for multiple models with different settings, mean and attention aggregation. Subscript indicates the best truncation layer for the *trunc* setting.



(a) Pairs of models: comparison of two embedding aggregation methods, mean and attention, to combine the source task model outputs



(b) Single models: performance gains by adding attention projections (attention) compared to having no additional parameters (mean)

Figure 4: Mean vs attention aggregation (full setting)

4.6 Combining Multiple Source Models

An interesting research question with respect to the embedding aggregation is how many models are actually needed to achieve good results and whether such models should be more or less similar to each other. E.g., NER and RE both focus on mention span extraction, RE and QA process relations between the entities in the text and MRPC model is more suitable for the semantic similarity tasks.

Firstly, we found that combinations of two models always outperform single models in the attention aggregation setting and, for the mean setting, pairs of models typically also perform better than the individual models except for the combinations with MRPC that tend to have lower scores (see Figure 8 for the comparison with mean and attention). E.g., NER with mean aggregation achieves 35.6 F1, RE achieves 52.3 and the combination of both (RE+NER) has 56.1 F1.

Secondly, we observed that combining three or more models typically works well for the full models. However, for the truncated setting there are only negligible gains when we combine multiple models (e.g., for RE+MRPC with attention we have 68.78 and for RE+MRPC+QA 68.81).

Lastly, model combinations that include MRPC tend to perform better than the rest which likely indicates the importance of semantic similarity for the coreference task. However, the combinations of RE+NER, RE+QA and QA+NER can also be beneficial, especially in the mean aggregation setting.

5 Related Work

Apart from the work on probing that was presented in the introduction, our work is closely related to the idea of *transfer learning* (Torrey and Shavlik, 2010), one of the ubiquitous paradigms in modern NLP. The idea of transfer learning is to train a model on a task A and then transfer the weights to a task B, either with or without further finetuning. This is the basis of most modern language models, which are pretrained and then applied or evaluated on specific downstream tasks (Devlin et al., 2019; Raffel et al., 2020; Jiang et al., 2023; Dubey et al., 2024). The pretraining data of large language models often contains samples from various natural language tasks, which renders most language models as multi-task learners (Yu et al., 2024). Multi-task learning describes a paradigm where a model is simultaneously trained on a range of tasks. While this concept is related to the work presented here, the main difference is that several source tasks are mixed together during (pre-)training usually, which

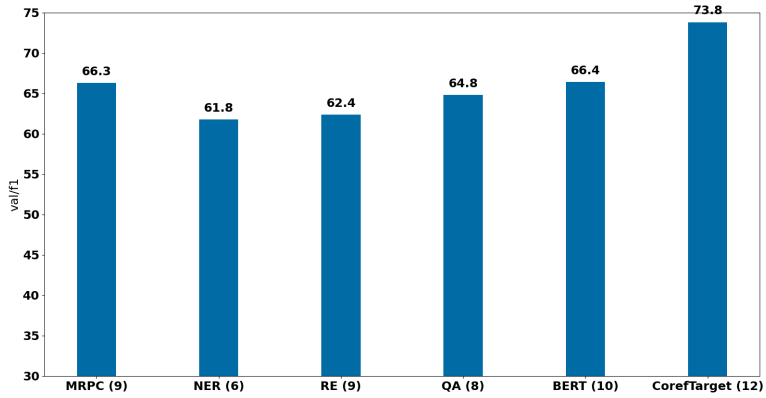
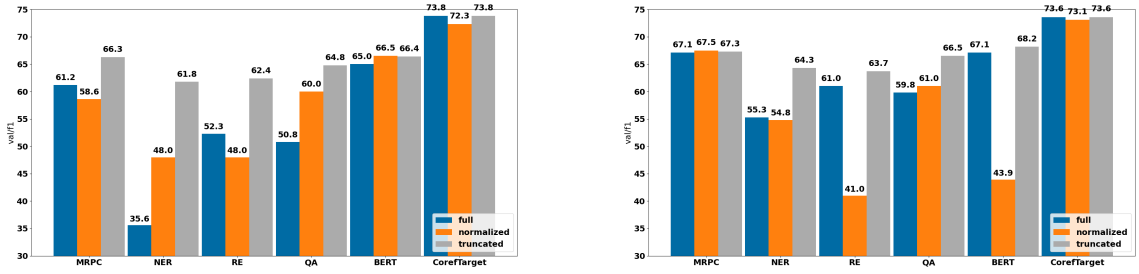


Figure 5: Source task model performance truncated to the best layer (in parentheses) with mean aggregation



(a) Mean aggregation

(b) Attention aggregation

Figure 6: Single models: full vs normalized vs truncated

is not the case with our work.

In some sense, our work is related to research around task arithmetics (Matena and Raffel, 2022; Chronopoulou et al., 2023; Ilharco et al., 2023; Belanec et al., 2024), which has the goal to explicitly compute task representations in networks, e.g. as differences to a random initalization, and implement transfer learning by means of difference vector arithmetics. In contrast, our work concentrates on hidden representations, rather than the parameters of the network.

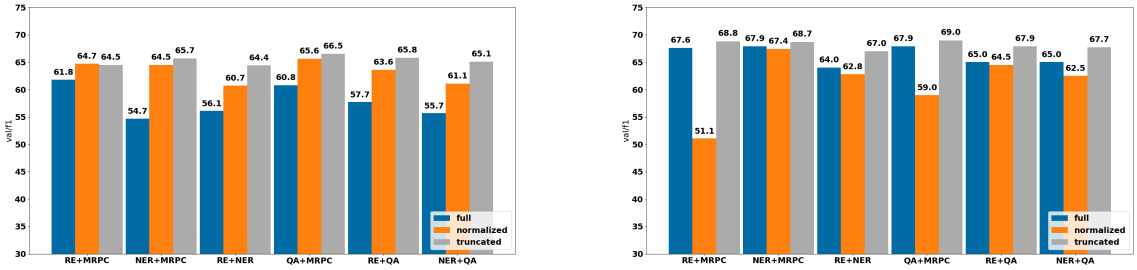
6 Conclusion

In this project we "reversed" the classical probing and investigated how different source task embeddings contribute to a target task (coreference resolution). Our experiments with Paraphrase Detection (MRPC), Named Entity Recognition (NER), Relation Extraction (RE) and Extractive Question Answering (QA) as source tasks show there are quite different embedding representations that achieve different scores on the target task ranging from 35.63 F1 (NER) to 61.16 F1 (MRPC) for single models.

Moreover, we found that the best performing embeddings were typically not the outputs of the last hidden layer but rather the representations generated at lower layers. MRPC was found to be the best source model, whereas, surprisingly, NER performed the worst.

We also explored different combinations of source models and found that two or more models typically outperform single ones. We considered mean and attention-based embedding aggregation methods and demonstrated the effectiveness of attention. For single models, we also showed that combining the outputs of several hidden layers instead of only one layer is beneficial. However, combining the outputs of all available layers is not necessarily a good strategy and usually the best scores can be achieved by combining only the outputs of the last 4 hidden layers that possibly contain more high-level, semantic information important for the coreference task.

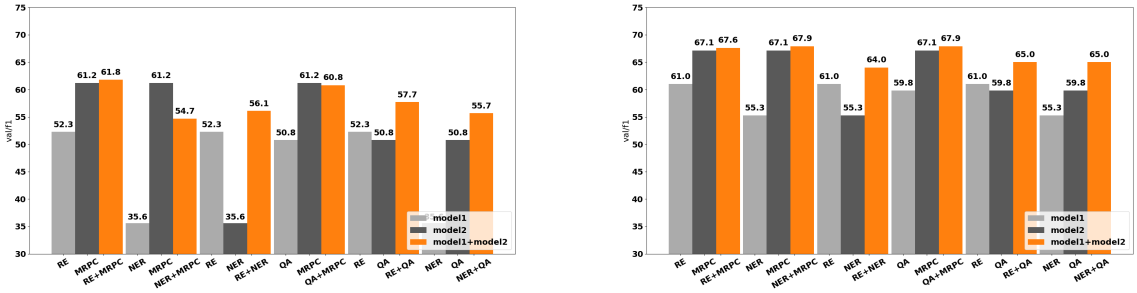
In the future it would be interesting to experiment with more types of embedding manipula-



(a) Mean aggregation

(b) Attention aggregation

Figure 7: Pairs of models: full vs normalized vs truncated



(a) Mean aggregation

(b) Attention aggregation

Figure 8: Single and combined (2x) models

tions. Also, a combination of truncation and normalization could possibly bring some gains for single models. Moreover, it would be interesting to check the effects of attention aggregation with hidden layer concatenations for multiple models (e.g., RE+MRPC). Finally, it would be interesting to replicate our experiments on larger (non-BERT) models and tasks (e.g., semantic role labeling, discourse relation classification etc.).

We hope that our experiments can help to clarify the impact of embeddings and their combinations on the target coreference task. We also hope that the reverse probing idea will facilitate further research on finding useful information in the task-specific representations that originate from different fine-tuned models.

Limitations

While this work sheds light on the potential of reverse probing and task embeddings, some limitations arise.

First, we exclusively work with BERT-based models. This gives us a controlled setup, but it also means our findings might not fully translate to larger models or other architectures like GPT, T5, or multilingual models. Future work needs to investigate a broader range of models.

Our choice of source tasks, Paraphrase Detection (MRPC), Named Entity Recognition (NER), Relation Extraction (RE), and Question Answering (QA), is not exhaustive. There are many other NLP tasks, such as sentiment analysis, syntactic parsing, or commonsense reasoning, that might contribute useful embeddings for coreference resolution. Also, some of the tasks are not necessarily simpler than coreference resolution (e.g., QA), which we chose as our target task. Generally, our conclusions are centered around coreference resolution. While this is a challenging and linguistically complex problem, our approach may not directly apply to other NLP tasks, such as machine translation or text summarization.

Lastly, there is the question of computational efficiency. Although we worked with relatively small models, combining embeddings from multiple layers and tasks does introduce extra processing overhead.

Acknowledgments

This work has been supported by the German Ministry of Education and Research (BMBF) as part of the project TRAILS (01IW24005).

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Robert Belanec, Simon Ostermann, Ivan Srba, and Maria Bielikova. 2024. Task prompt vectors: Effective initialization through multi-task soft-prompt transfer. *arXiv preprint arXiv:2408.01119*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226.

Alexandra Chronopoulou, Jonas Pfeiffer, Joshua Maynez, Xinyi Wang, Sebastian Ruder, and Priyanka Agrawal. 2023. Language and task arithmetic with parameter-efficient layers for zero-shot summarization. *arXiv preprint arXiv:2311.09344*.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Giuliano Martinelli, Edoardo Barba, and Roberto Navigli. 2024. Maverick: Efficient and accurate coreference resolution defying recent trends. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13380–13394, Bangkok, Thailand. Association for Computational Linguistics.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, pages 17703–17716. Curran Associates, Inc.

Simon Ostermann, Sheng Zhang, Michael Roth, and Peter Clark. 2019. Commonsense inference in natural language processing (COIN) - shared task report. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 66–74, Hong Kong, China. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou,

Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. 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, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. Publisher Copyright: © 7th International Conference on Learning Representations, ICLR 2019. All Rights Reserved.; 7th International Conference on Learning Representations, ICLR 2019 ; Conference date: 06-05-2019 Through 09-05-2019.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

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

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533. Association for Computational Linguistics.

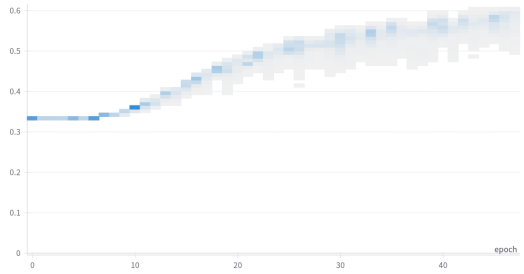
Jun Yu, Yutong Dai, Xiaokang Liu, Jin Huang, Yishan Shen, Ke Zhang, Rong Zhou, Eashan Adhikarla, Wenxuan Ye, Yixin Liu, et al. 2024. Unleashing the power of multi-task learning: A comprehensive survey spanning traditional, deep, and pretrained foundation model eras. *arXiv preprint arXiv:2404.18961*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

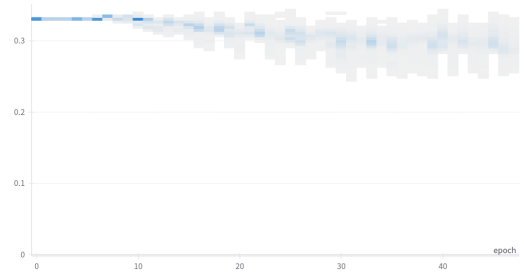
Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

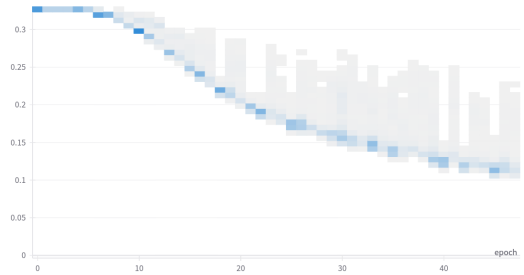
A Additional Figures



(a) MRPC



(b) RE



(c) NER

Figure 9: MRPC+RE+NER with attention aggregation (full setting)