On the Transformation of Latent Space in Fine-Tuned NLP Models

WARNING: This paper contains model outputs which may be disturbing to the reader

Nadir Durrani[◇] **Hassan Sajjad**[♣*] **Fahim Dalvi**[◇] **Firoj Alam**[◇]
[◇]Qatar Computing Research Institute, Hamad Bin Khalifa University, Qatar
[♣]Faculty of Computer Science, Dalhousie University, Canada
{ndurrani,faimaduddin, fialam}@hbku.edu.qa, hsajjad@dal.ca

Abstract

We study the evolution of latent space in fine-tuned NLP models. Different from the commonly used probing-framework, we opt for an unsupervised method to analyze representations. More specifically, we discover latent concepts in the representational space using hierarchical clustering. We then use an alignment function to gauge the similarity between the latent space of a pre-trained model and its fine-tuned version. We use traditional linguistic concepts to facilitate our understanding and also study how the model space transforms towards task-specific information. We perform a thorough analysis, comparing pre-trained and fine-tuned models across three models and three downstream tasks. The notable findings of our work are: i) the latent space of the higher layers evolve towards task-specific concepts, ii) whereas the lower layers retain generic concepts acquired in the pre-trained model, iii) we discovered that some concepts in the higher layers acquire polarity towards the output class, and iv) that these concepts can be used for generating adversarial triggers.

1 Introduction

The revolution of deep learning models in NLP can be attributed to transfer learning from pre-trained language models. Contextualized representations learned within these models capture rich linguistic knowledge that can be leveraged towards novel tasks e.g. classification of COVID-19 tweets (Alam et al., 2021; Valdes et al., 2021), disease prediction (Rasmy et al., 2020) or natural language understanding tasks such as SQUAD (Rajpurkar et al., 2016) and GLUE (Wang et al., 2018).

Despite their success, the opaqueness of deep neural networks remain a cause of concern and has spurred a new area of research to analyze these models. A large body of work analyzed the knowledge learned within representations of pre-trained models (Belinkov et al., 2017a; Conneau et al., 2018; Liu et al., 2019; Tenney et al., 2019; Durrani et al., 2019; Rogers et al., 2020) and showed the presence of core-linguistic knowledge in various parts of the network. Although transfer learning using pre-trained models has become ubiquitous, very few papers (Merchant et al., 2020; Mosbach et al., 2020; Durrani et al., 2021) have analyzed the representations of the fine-tuned models. Given their massive usability, interpreting fine-tuned models and highlighting task-specific peculiarities is critical for their deployment in real-word scenarios, where it is important to ensure fairness and trust when applying AI solutions.

In this paper, we focus on analyzing fine-tuned models and investigate: *how does the latent space evolve in a fine-tuned model?* Different from the commonly used probing-framework of training a post-hoc classifier (Belinkov et al., 2017b; Dalvi et al., 2019a), we opt for an unsupervised method to analyze the latent space of pre-trained models. More specifically, we cluster contextualized representations in high dimensional space using hierarchical clustering and term these clusters as the *Encoded Concepts* (Dalvi et al., 2022). We then analyze how these encoded concepts evolve as the models are fine-tuned towards a downstream task. Specifically, we target the following questions: *i) how do the latent spaces compare between base*[1] *and the fine-tuned models? ii) how does the presence of core-linguistic concepts change during transfer learning?* and *iii) how is the knowledge of downstream tasks structured in a fine-tuned model?*

We use an alignment function (Sajjad et al., 2022) to compare the concepts encoded in the fine-tuned models with: i) the concepts encoded in their pre-trained base models, ii) the human-defined concepts (e.g. parts-of-speech tags or semantic properties), and iii) the labels of the downstream task towards which the model is fine-tuned.

[*] This work was carried out while the author was at QCRI.

[1] We use "base" and "pre-trained" models interchangeably.

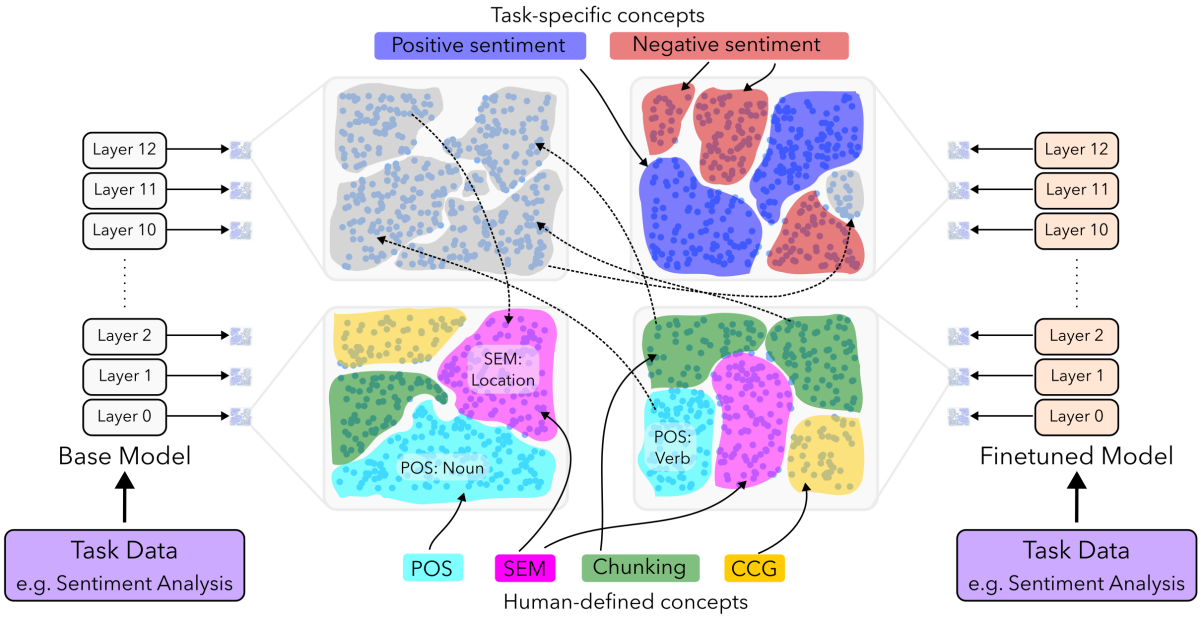


Figure 1: Comparing encoded concepts of a model across different layers with: i) the concepts encoded its base model (dashed lines), ii) human-defined concepts (e.g. POS tags or semantic properties), and iii) task specific concepts (e.g. positive or negative sentiment class).

We carried out our study using three pre-trained transformer language models; BERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020) and ALBERT (Lan et al., 2019), analyzing how their representation space evolves as they are fine-tuned towards the task of Sentiment Analysis (SST-2, Socher et al., 2013), Natural Language Inference (MNLI, Williams et al., 2018) and Hate Speech Detection (HSD, Mathew et al., 2020). Our analysis yields interesting insights such as:

- The latent space of the models substantially evolve from their base versions after fine-tuning.
- The latent space representing core-linguistic concepts is limited to the lower layers in the fine-tuned models, contrary to the base models where it is distributed across the network.
- We found task-specific polarity concepts in the higher layers of the Sentiment Analysis and Hate Speech Detection tasks.
- These polarized concepts can be used as triggers to generate adversarial examples.
- Compared to BERT and XLM, the representational space in ALBERT changes significantly during fine-tuning.

2 Methodology

Our work builds on the Latent Concept Analysis method (Dalvi et al., 2022) for interpreting representational spaces of neural network models. We cluster contextualized embeddings to discover *Encoded Concepts* in the model and study the evolution of the latent space in the fine-tuned model by aligning the encoded concepts of the fine-tuned model to: i) their pre-trained version, ii) the human-defined concepts and iii) the task-specific concepts (for the task the pre-trained model is fine-tuned on). Figure 1 presents an overview of our approach. In the following, we define the scope of *Concept* and discuss each step of our approach in detail.

2.1 Concept

We define concept as a group of words that are clustered together based on any linguistic relation such as lexical, semantic, syntactic, morphological etc. Formally, consider $C_t(n)$ as a concept consisting of a unique set of words $\{w_1, w_2, \ldots, w_J\}$ where J is the number of words in C_t, n is a concept identifier, and t is the concept type which can be an encoded concept (ec), a human-defined concept ($pos : verbs, sem : loc, \ldots$) and a class-based concept ($sst : +ive, hsd : toxic, \ldots$).

Encoded Concepts: Figure 2 shows a few examples of the encoded concepts discovered in the

(a) Nouns ending with "y" (b) Named Entities – TV (c) Racial Slurs

Figure 2: Examples of encoded concepts. The size of a specific word is based on its frequency in the cluster, defined by the number of times different contextual representations of a word were grouped in the same cluster.

BERT model, where the concept is defined by a group based on nouns ending with "y" (Figures 2a) or a group based on TV related named entities (Figure 2b). Similarly, Figure 2c is a concept representing racial slurs in a BERT model tuned for Hate Speech Detection (HSD) task. We denote this concept as C_{ec}(`bert-hsd-layer10-c227`) = $\{paki, nigger, mudslime, redneck \dots\}$, i.e. the concept was discovered in the layer 10 of the BERT-HSD model and $c227$ is the concept number.

Human Concepts: Each individual tag in the human-defined concepts such as parts-of-speech (POS), semantic tagging (SEM) represents a concept C. For example, $C_{pos}(JJR) = \{greener, taller, happier, \dots\}$ defines a concept containing comparative adjectives in the POS tagging task, $C_{sem}(MOY) = \{January, February, \dots, December\}$ defines a concept containing months of the year in the semantic tagging task.

Task-specific Concepts: Another kind of concept that we use in this work is the task-specific concepts where the concept represents affinity of its members with respect to the task labels. Consider a sentiment classification task with two labels "positive" and "negative". We define $C_{sst}(+ve)$ as a concept containing words when they only appear in sentences that are labeled positive. Similarly, we define $C_{hsd}(toxic)$ as a concept that contain words that only appear in the sentences that were marked as toxic.

2.2 Latent Concept Discovery

A vector representation in the neural network model is composed of feature attributes of the input words. We group the encoded vector representations using a clustering approach discussed below. The underlying clusters, that we term as the encoded concepts, are then matched with the human-defined concepts using an alignment function.

Formally, consider a pre-trained model $\mathbf{M}$ with L layers: $\{l_1, l_2, \dots, l_L\}$. Given a dataset $\mathbb{W} = \{w_1, w_2, ..., w_N\}$, we generate feature vectors, a sequence of latent representations: $\mathbb{W} \xrightarrow{\mathbb{M}} \mathbf{z}^l = \{\mathbf{z}_1^l, \dots, \mathbf{z}_n^l\}$[2] by doing a forward-pass on the data for any given layer l. Our goal is to cluster representations $\mathbf{z}^l$, from task-specific training data to obtain *encoded concepts*.

We use agglomerative hierarchical clustering (Gowda and Krishna, 1978), which assigns each word to its individual cluster and iteratively combines the clusters based on Ward's minimum variance criterion, using intra-cluster variance. Distance between two vector representations is calculated with the squared Euclidean distance. The algorithm terminates when the required K clusters (i.e. encoded concepts) are formed, where K is a hyper-parameter. Each encoded concept represents a latent relationship between the words present in the cluster.

2.3 Alignment

Once we have obtained a set of encoded concepts in the base (pre-trained) and fine-tuned models, we want to align them to study how the latent space has evolved during transfer learning. Sajjad et al. (2022) calibrated representational space in transformer models with different linguistic concepts to generate their explanations. We extend their alignment function to align latent spaces within a model and its fine-tuned version. Given a concept $C_1(n)$ with J number of words, we consider it to be θ-aligned (Λ_θ) with concept $C_2(m)$, if they satisfy the following constraint:

[2]Each element z_i denotes contextualized word representation for the corresponding word w_i in the sentence.

$$\Lambda_\theta(C_1, C_2) = \begin{cases} 1, & \text{if } \frac{\sum_{w \in C_1} \sum_{w' \in C_2} \delta(w, w')}{J} \geq \theta \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where Kronecker function $\delta(w, w')$ is defined as

$$\delta(w, w') = \begin{cases} 1, & \text{if } w = w' \\ 0, & \text{otherwise} \end{cases}$$

Human-defined Concepts The function can be used to draw a mapping between concepts different types of discussed in Section 2.1. To investigate *how the transfer learning impacts human-defined knowledge*, we align the latent space to the human-defined concepts such as $C_{pos}(NN)$ or $C_{chunking}(PP)$.

Task Concepts Lastly, we compare the encoded concepts with the task-specific concepts. Here, we use the alignment function to mark affinity of an encoded concept. For the Sentiment Analysis task, let a task-specific concept $C_{sst}(+ve) = \{w_1^+, w_2^+, \ldots, w_n^+\}$ defined by a set words that only appeared in positively labeled sentences $S = \{s_1^+, s_2^+, \ldots, s_n^+\}$. We call a concept $C_{ec} = \{x_1, x_2, \ldots, x_n\}$ aligned to $C_{sst}(+ve)$ and mark it positive if all words ($\geq \theta$) in the encoded concept appeared in positively labeled sentences. Note that here a word represents an instance based on its contextualized embedding. We similarly align C_{ec} with $C_{sst}(-ve)$ to discover negative polarity concepts.

3 Experimental Setup

3.1 Models and Tasks

We experimented with three popular transformer architectures namely: BERT-base-cased (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020) and ALBERT (v2) (Lan et al., 2019) using the base versions (13 layers and 768 dimensions). To carryout the analysis, we fine-tuned the base models for the tasks of sentiment analysis using the Stanford sentiment treebank dataset (SST-2, Socher et al., 2013), natural language inference (MNLI, Williams et al., 2018) and the Hate Speech Detection task (HSD, Mathew et al., 2020).

3.2 Clustering

We used the task-specific training data for clustering using both the base (pre-trained) and fine-tuned models. This enables to accurately compare the representational space generated by the same data. We do a forward-pass over both base and fine-tuned models to generate contextualized feature vectors[3] of words in the data and run agglomerative hierarchical clustering over these vectors. We do this for every layer independently, obtaining K clusters (a.k.a encoded concepts) for both base and fine-tuned models. We used $K = 600$ for our experiments.[4] We carried out preliminary experiments (all the BERT-base-cased experiments) using $K = 200, 400, \ldots, 1000$ and all our experiments using $K = 600$ and $K = 1000$. We found that our results are not sensitive to these parameters and the patterns are consistent with different cluster settings (please see Appendix B).

3.3 Human-defined Concepts

We experimented with traditional tasks that are defined to capture core-linguistic concepts such as word morphology: part-of-speech tagging using the Penn TreeBank data (Marcus et al., 1993), syntax: chunking tagging using CoNLL 2000 shared task dataset (Tjong Kim Sang and Buchholz, 2000), CCG super tagging using CCG Tree-bank (Hockenmaier, 2006) and semantic tagging using the Parallel Meaning Bank data (Abzianidze et al., 2017). We trained BERT-based sequence taggers for each of the above tasks and annotate the task-specific training data. Each core-linguistic task serves as a human-defined concept that is aligned with encoded concepts to measure the representation of linguistic knowledge in the latent space. Appendix A presents the details on human defined concepts, data stats and tagger accuracy.

3.4 Alignment Threshold

We consider an encoded concept to be aligned with another concept, if it has at least 95%[5] match in the number of words. We only consider concepts that have more than 5 word-types. Note that the encoded concepts are based on contextualized embedding where a word has different embeddings depending on the context.

[3] We use NeuroX toolkit (Dalvi et al., 2019b) to extract contextualized representations.

[4] We experimented with ELbow (Thorndike, 1953) and Silhouette (Rousseeuw, 1987) methods to find the optimal number of clusters, but could not observe a reliable pattern. Selecting between $600 - 1000$ clusters gives the right balance to avoid over-clustering (many small clusters) and under-clustering (a few large clusters).

[5] Using an overlap of $\geq 95\%$ provides a very tight threshold, allowing only 5% of noise. Our patterns were consistent at lower and higher thresholds.

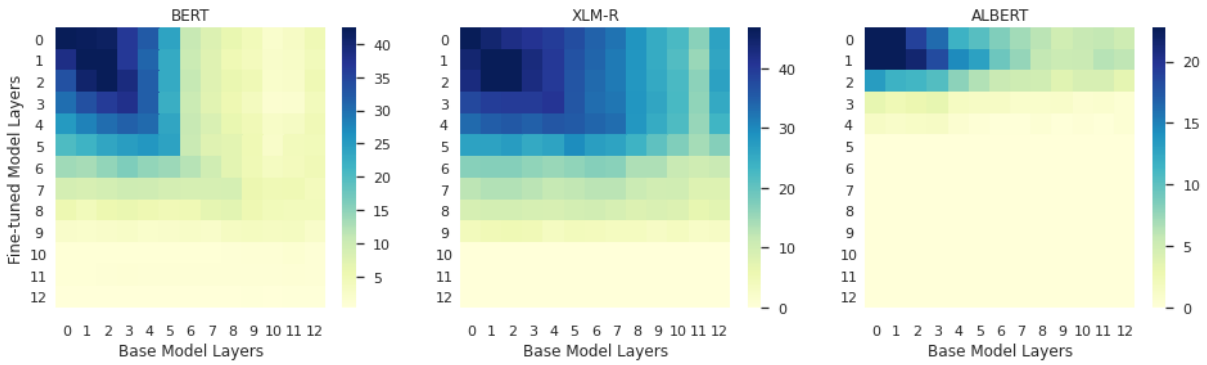


Figure 3: Comparing encoded concepts of base models with their SST fine-tuned versions. X-axis = base model, Y-axis = fine-tuned model. Each cell in the matrix represents a percentage (aligned concepts/total concepts in a layer) between the base and fine-tuned models. Darker color means higher percentage. Detailed plots with actual overlap values are provided in the Appendix.

4 Analysis

Language model pre-training has been shown to capture rich linguistic features (Tenney et al., 2019; Belinkov et al., 2020) that are redundantly distributed across the network (Dalvi et al., 2020; Durrani et al., 2022). We analyze how the representational space transforms when tuning towards a downstream task: i) how much knowledge is carried forward and ii) how it is redistributed, using our alignment framework.

4.1 Comparing Base and Fine-tuned Models

How do the latent spaces compare between base and fine-tuned models? We measure the overlap between the concepts encoded in the different layers of the base and fine-tuned models to guage the extent of transformation. Figures 3 compares the concepts in the base BERT, XLM-RoBERTa and ALBERT models versus their fine-tuned variants on the SST-2 task.[6] We observe a high overlap in concepts in the lower layers of the model that starts decreasing as we go deeper in the network, completely diminishing towards the end. We conjecture that ***the lower layers of the model retain generic language concepts learned in the base model, where as the higher layers are now learning task-specific concepts.***[7] Note, however, that the lower layers also do not completely align between the models, which shows that all the layers go through substantial changes during transfer learning.

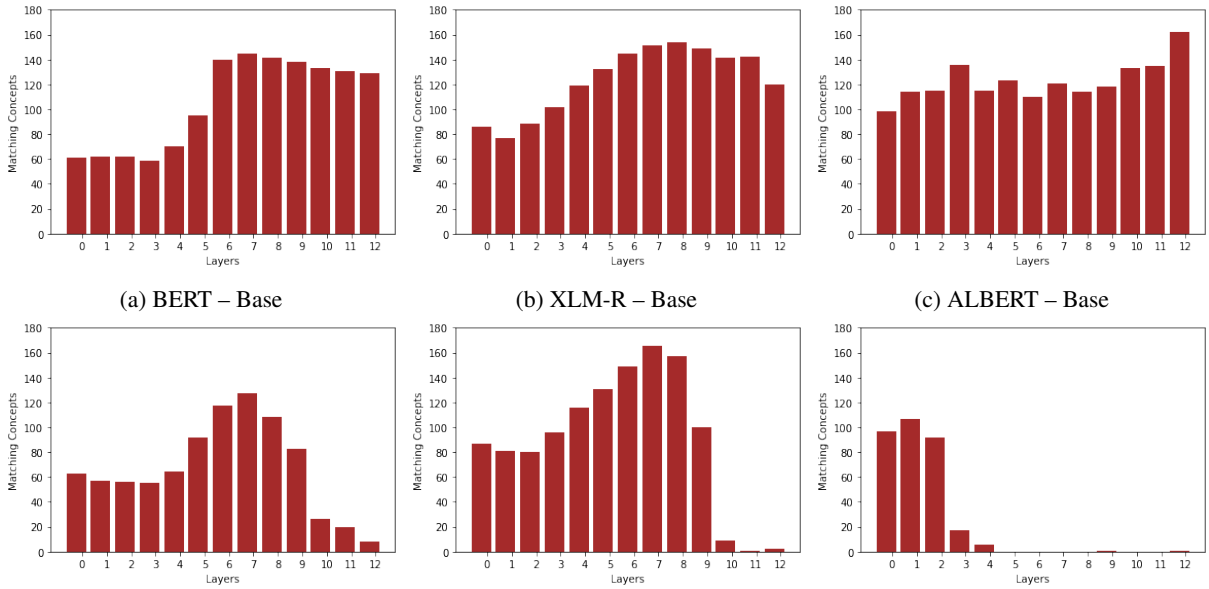
Comparing Architectures: The spread of the shaded area along the x-axis, particularly in XLM-R, reflects that some higher layer latent concepts in the base model have shifted towards the lower layers of the fine-tuned model. The latent space in the higher layers now reflect task-specific knowledge which was not present in the base model. ALBERT shows a strikingly different pattern with only the first 2-3 layers exhibiting an overlap with base concepts. This could be attributed to the fact that ALBERT shares parameters across layers while the other models have separate parameters for every layer. ALBERT has less of a luxury to preserve previous knowledge and therefore its space transforms significantly towards the downstream task. Notice that the overlap is comparatively smaller (38% vs. 52% and 46% compared to BERT and XLM-R respectively), even in the embedding layer, where the words are primarily grouped based on lexical similarity.

4.2 Presence of Linguistic Concepts in the Latent Space

How does the presence of core-linguistic concepts change during transfer learning? To validate our hypothesis that generic language concepts are now predominantly retained in the lower half, we analyze how the linguistic concepts spread across the layers in the pre-trained and fine-tuned models by aligning the latent space to the human-defined concepts. Figure 4 shows that the latent space of the models capture POS concepts (e.g., determiners, past-tense verbs, superlative adjectives etc.) The information is present across the layers in the pre-trained models, however, as the model is fine-tuned towards downstream task, it is retained only

[6] Please see all results in Appendix C.1.

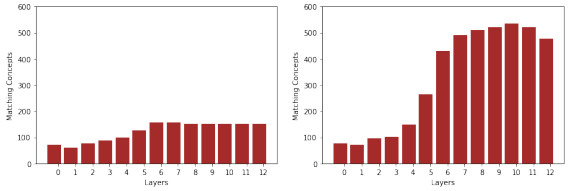
[7] Our next results comparing the latent space with human-defined language concepts (Section 4.2) and the task specific concepts (Section 4.3) reinforces this hypothesis.



(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

(d) BERT – SST (e) XLM-R – SST (f) ALBERT – SST

Figure 4: Alignment of the encoded concepts with POS concepts (e.g., determiners, past-tense verbs, superlative adjectives) in the base and fine-tuned SST models. The maximum possible concepts per layer are 600 (total # of clusters). Note that the POS information depreciates significantly in the final layers in the SST-tuned models.



(a) BERT – Base (b) BERT – POS

Figure 5: Alignment of the the encoded concepts with POS in the BERT base versus fine-tuned POS models. In contrast to the results in Figure 4, POS concepts appreciate significantly when the model is tuned towards the POS task. At most 23% concepts align in the BERT-base model as opposed to BERT-pos where close to 84% encoded concepts are aligned to the POS tags.

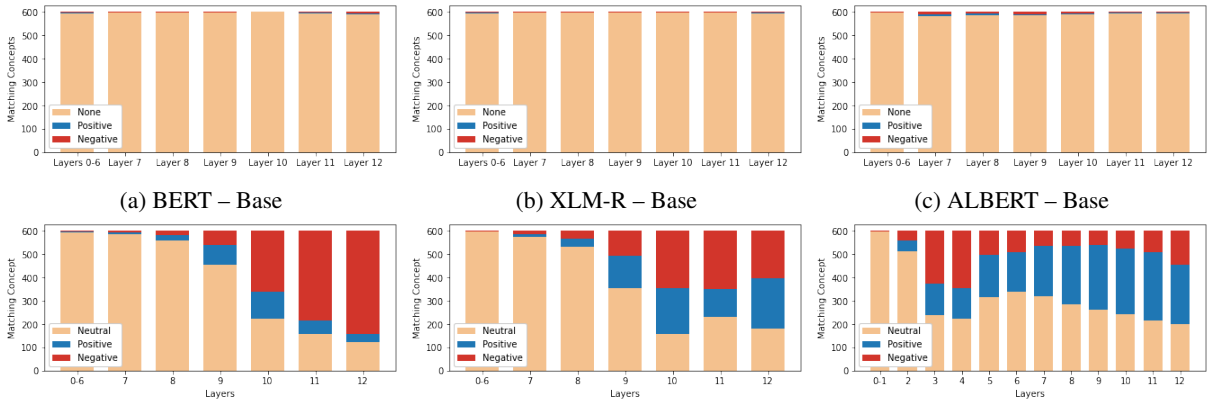
at the lower and middle layers. We can draw two conclusions from this result: i) POS information is important for a foundational task such as language modeling (predicting the masked word), but not critically important for a sentence classification task like sentiment analysis. To strengthen our argument and confirm this further, we fine-tuned a BERT model towards the task of POS tagging itself. Figure 5 shows the extent of the alignment between POS concept with BERT-base and BERT tuned models towards the POS. Notice that more than 80% encoded concepts in the final layers of the BERT-POS model are now aligned with the POS concept as opposed to the BERT-SST model where POS concept (as can be seen in Figure 4) decreased to less than 5%.

Comparing Tasks and Architectures We found these observations to be consistently true for other tasks (e.g., MNLI and HSD) and human-defined concepts (e.g., SEM, Chunking and CCG tags) across the three architectures (i.e., BERT, XLM-R and ALBERT) that we study in this paper.[8] Table 1 compares an overall presence of core-linguistic concepts across the base and fine-tuned models. We observe a consistent deteriorating pattern across all human-defined concepts. In terms of architectural difference we again found ALBERT to show a substantial difference in the representation of POS post fine-tuning. The number of concepts not only regressed to the lower layers, but also decreased significantly as opposed to the base model.

4.3 Task-specific Latent Spaces

How is the knowledge of downstream tasks structured in a fine-tuned models? Now that we have established that the latent space of higher layers are substantially different from base models and from linguistic concepts, we probe: *what kind of knowledge is learned in the latent space of higher layers?* Previous research (Kovaleva et al., 2019; Merchant et al., 2020; Durrani et al., 2021) found

[8] Please see Appendix C.2 for all the results.



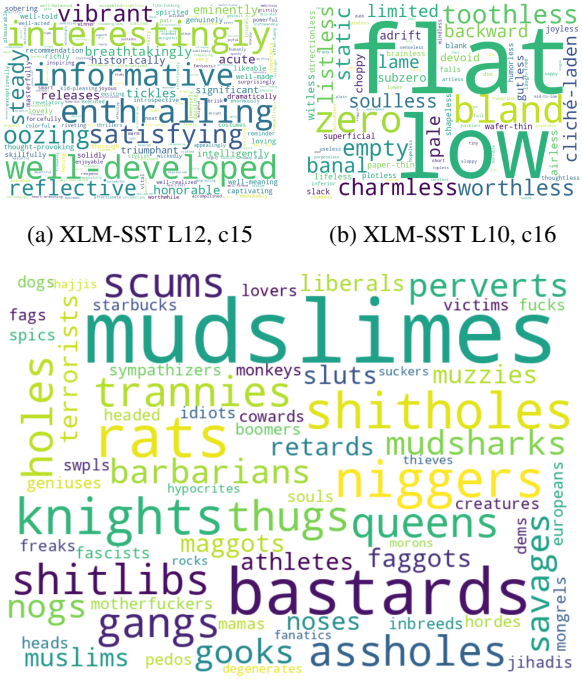
(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

(d) BERT – SST (e) XLM-R – SST (f) ALBERT – SST

Figure 6: Aligning encoded concepts with the task specific concepts in Base and their corresponding SST tuned models.

Tasks	POS	SEM	Chunking	CCG
BERT(B)	17.6	23.5	27.3	20.6
BERT(SST)	11.2	17.0	21.4	15.1
XLM-R(B)	20.6	22.6	22.1	17.9
XLM-R(SST)	15.1	18.0	19.9	15.2
ALBERT(B)	20.4	27.3	32.6	25.6
ALBERT(SST)	4.1	5.3	8.2	4.1

Table 1: Overall presence (percentage of aligned concepts) of human-defined concepts in base (B) versus SST fine-tuned models.



(a) XLM-SST L12, c15 (b) XLM-SST L10, c16

(c) XLM-HSD L10, c576

Figure 7: Polarity Concepts in XLM-R models: Positive (top left) and Negative (top right) in the SST task, Toxic Concept (bottom) in the HSD task.

that the higher layers are optimized for the task. We also noticed how the concepts learned in the top 6 layers of the BERT-POS model completely evolve towards the (POS) task labels (See Figure 5). We now extend this experiment towards the sentence-level tasks and investigate the extent of alignment between latent concepts of the fine-tuned models with its task labels. The SST-2 task predicts the sentiment (positive or negative) of a sentence. Using the class label, we form positive and negative polarity concepts, and align the polarity concepts with the encoded concepts.[9] If an encoded concept is not aligned with any polarity concept, we mark the concept as "Neutral". Figure 6 shows that the concepts in the final layers acquire polarity towards the task of output classes compared to the base model where we only see neutral concepts throughout the network. Figure 7 shows an example of positive (top left) and negative polarity (top right) concepts in the XLM-R model tuned for the SST task. The bottom half shows a toxic concept in the model trained towards the HSD task. Please see Appendix D for more examples.

Comparing architectures Interestingly, the presence of polarity clusters is not always equal. The last two layers of BERT-SST are dominated by negative polarity clusters, while ALBERT showed an opposite trend where the positive polarity concepts were more frequent. We hypothesized that the im-

[9]Positive polarity concept is made up of words that only appeared in the positively labeled sentences. We say an encoded concept (C_{ec}) is aligned to positive polarity concept (C^+) if $\geq 95\%$ words in $C_{ec} \in C^+$. Note that the opposite is not necessarily true.

balance in the presence of polarity clusters may reflect prediction bias towards/against a certain class. However, we did not find a clear evidence for this in a pilot experiment. We collected predictions for all three models over a random corpus of 37K sentences. The models predicted negative sentiment by 69.5% (BERT), 67.4% (XLM) and 64.4% (ALBERT). While the numbers weakly correlate with the number of negative polarity concepts in these models, a thorough investigation is required to obtain accurate insights. We leave a detailed exploration of this for future.

ALBERT showed the evolution of polarity clusters much earlier in the network (Layer 3 onwards). This is inline with our previous results on aligning encoded concepts of base and fine-tuned models (Figure 3). We found that the latent space in ALBERT evolved the most, overlapping with its base model only in the first 2-3 layers. The POS-based concepts were also reduced just to the first two layers (Figure 4). Here we can see that the concepts learned within the remaining layers acquire affinity towards the task specific labels. We found these results to be consistent with the hate speech task (See Appendix C.3) but not in the MNLI task, where we did not find the latent concepts to acquire affinity towards the task labels. This could be attributed to the complexity and nature of the MNLI task. Unlike the SST-2 and HSD tasks, where lexical triggers serve as an important indicators for the model, MNLI requires intricate modeling of semantic relationships between premise and hypothesis to predict entailment. Perhaps an alignment function that models the interaction between the concepts of premise and hypothesis is required. We leave this exploration for the future.

5 Adversarial Triggers

The discovery of polarized concepts in the SST-2 and HSD tasks, motivated us to question: ***whether the fine-tuned model is learning the actual task or relying on lexical triggers to solve the problem.*** Adversarial examples have been used in the literature to highlight model's vulnerability (Kuleshov et al., 2018; Wallace et al., 2019). We show that our polarity concepts can be used to generate such examples using the following formulation:

Let $C_{ec}(+ve) = \{C_1^+, C_2^+, \ldots, C_N^+\}$ be a set of latent concepts that are identified to have a strong affinity towards predicting positive sentiment in the SST task. Let $S^- = \{s_1^-, s_2^-, \ldots, s_M^-\}$ be the sentences in a dev-set that are predicted as negative by the model. We compute the flipping accuracy of each concept C_x^+ using the following function:

$$F(C_x^+, S^-) = \frac{1}{N_a} \sum_{w_i \in C_x^+} \sum_{s_j \in S_-} \gamma(w_i, s_j)$$

where $\gamma(w_i, s_j) = 1$, if prepending w_i to the sentence s_j flips the model's prediction from negative to positive. Here N_a is the total number of adversarial examples that were generated, and equates to $|C_x^+| \times |S^-|$. We similarly compute the flipping accuracy $F(C_x^-, S^+)$ of the concepts that acquire affinity towards the negative class. The concepts with high flipping accuracy can be used to generate adversarial examples.

We compute the flipping accuracy of each polarized concept on a small hold-out set. Table 2 shows the average flipping accuracy of the top-5 polarized concepts for each class (positive/negative in SST-2 and toxic/non-toxic in the HSD task) across final three layers on the test-set. We observed that by just prepending the words in highly polarized concepts, we are able to effectively flip the model's prediction by up to 91.5%. This shows that ***these models are fragile and heavily rely on lexical triggers to make predictions***. In the case of Hate Speech Detection task, we observed that while it is easy to make a non-toxic sentence toxic, it is hard to reverse the affect.

Tasks	Layer 10	Layer 11	Layer 12
	BERT SST		
+ve → −ve	43.6	41.2	43.4
−ve → +ve	41.0	42.1	44.7
	XLM-RoBERTa SST		
+ve → −ve	42.8	41.7	43.0
−ve → +ve	29.7	31.1	33.7
	ALBERT SST		
+ve → −ve	69.2	73.8	77.2
−ve → +ve	69.6	74.2	70.9
	BERT HS		
nt → tx	65.2	41.5	59.3
tx → nt	11.2	6.74	8.91
	XLM-RoBERTa HS		
nt → tx	57.7	69.0	38.9
tx → nt	7.23	9.14	9.60
	ALBERT HS		
nt → tx	84.9	65.4	91.5
tx → nt	0.00	0.00	0.00

Table 2: Flipping accuracy (%age) of top-5 polarized concepts: +ve → −ve = flipping a positive sentence to negative using negative polarity concept, nt → tx = converting a non-toxic sentence toxic using toxic concept.

Comparing Architectures We found ALBERT to be an outlier once again with a high flipping accuracy, which shows that ALBERT relies on these cues more than the other models and is therefore more prone to adversarial attacks.

6 Related Work

A plethora of papers have been written in the past five years on interpreting deep NLP models. The work done in this direction can be broadly classified into: i) post-hoc representation analysis that encode the contextualized embedding for the knowledge learned (Dalvi et al., 2017; Belinkov et al., 2020; Rogers et al., 2020; Lepori and McCoy, 2020) and ii) causation analysis that connect input features with model behavior as a whole and at a level of individual predictions (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018).[10] Our work mainly falls in the former category although we demonstrated a causal link between the encoded knowledge and model predictions by analyzing the concepts in the final layers and demonstrating how they can be used to generate adversarial examples with lexical triggers. Recent work (Feder et al., 2021; Elazar et al., 2021) formally attempts to bridge the gap by connecting the two lines of work.

Relatively less work has been done on interpreting fine-tuned models. Zhao and Bethard (2020) analyzed the heads encoding negation scope in fine-tuned BERT and RoBERTa models. Merchant et al. (2020); Mosbach et al. (2020) analyzed linguistic knowledge in pre-trained models and showed that while fine-tuning changes the upper layers of the model, but does not lead to "catastrophic forgetting of linguistic phenomena". Our results resonate with their findings, in that the higher layers learn task-specific concepts.[11] However, similar to Durrani et al. (2021) we found depreciation of linguistic knowledge in the final layers. Mehrafarin et al. (2022) showed that the size of the datasets used for fine-tuning should be taken into account to draw reliable conclusions when using probing classifiers. A pitfall to the probing classifiers is the difficulty to disentangle probe's capacity to learn from the actual knowledge learned within the representations (Hewitt and Liang, 2019). Our work is different from all the previous work done on interpreting fine-tuned models. We do away from the limitations of probing classifiers by using an unsupervised approach.

Our work is inspired by the recent work on discovering latent spaces for analyzing pre-trained models (Michael et al., 2020; Dalvi et al., 2022; Fu and Lapata, 2022; Sajjad et al., 2022). Like Dalvi et al. (2022); Sajjad et al. (2022) we discover encoded concepts in pre-trained models and align them with pre-defined concepts. Different from them, we study the evolution of latent spaces of fine-tuned models.

7 Conclusion

We studied the evolution of latent space of pre-trained models when fine-tuned towards a downstream task. Our approach uses hierarchical clustering to find encoded concepts in the representations. We analyzed them by comparing with the encoded concepts of base model, human-defined concepts, and task-specific concepts. We showed that the latent space of fine-tuned model is substantially different from their base counterparts. The human-defined linguistic knowledge largely vanishes from the higher layers. The higher layers encode task-specific concepts relevant to solve the task. Moreover, we showed that these task-specific concepts can be used in generating adversarial examples that flips the predictions of the model up to 91% of the time in the case of ALBERT Hate Speech model. The discovery of word-level task-specific concepts suggest that the models rely on lexical triggers and are vulnerable to adversarial attacks.

8 Limitations

The hierarchical clustering is memory intensive. For instance, the clustering of 250k representation vectors, each of size 768 consumes 400GB of CPU memory. This limits the applicability of our approach to small to medium data sizes. Moreover, our approach is limited to word-level concepts. The models may also learn phrasal concepts to solve a task. We speculate that the low matches of affinity concepts in the MNLI task is due to the limitation of our approach in analyzing phrasal units.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback and suggestions on the earlier draft of this paper.

[10] Please read (Belinkov and Glass, 2019; Sajjad et al., 2021) for comprehensive surveys of methods.

[11] Other works such as (Ethayarajh, 2019; Sajjad et al., 2023) have also shown higher layers to capture task-specific information.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '17, pages 242–247, Valencia, Spain.

Firoj Alam, Fahim Dalvi, Shaden Shaar, Nadir Durrani, Hamdy Mubarak, Alex Nikolov, Giovanni Da San Martino, Ahmed Abdelali, Hassan Sajjad, Kareem Darwish, and Preslav Nakov. 2021. Fighting the covid-19 infodemic in social media: A holistic perspective and a call to arms. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1):913–922.

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*, ACL '17, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2020. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 45(1):1–57.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

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*, IJCNLP '17, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

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*, ACL '18, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D. Anthony Bau, and James Glass. 2019a. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI, Oral presentation)*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and Improving Morphological Learning in the Neural Machine Translation Decoder. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in BERT. In *International Conference on Learning Representations*.

Fahim Dalvi, Avery Nortonsmith, Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, and James Glass. 2019b. Neurox: A toolkit for analyzing individual neurons in neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9851–9852.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP-2020)*, Online.

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 '19, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Nadir Durrani, Fahim Dalvi, and Hassan Sajjad. 2022. Linguistic correlation analysis: Discovering salient neurons in deepnlp models.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL '19, pages 1504–1516, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. 2021. How transfer learning impacts linguistic knowledge in deep NLP models? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4947–4957, Online. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of*

the Association for Computational Linguistics, 9:160–175.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Yao Fu and Mirella Lapata. 2022. Latent topology induction for understanding contextualized representations.

K Chidananda Gowda and G Krishna. 1978. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2):105–112.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL '18, pages 1195–1205, New Orleans, Louisiana, USA. Association for Computational Linguistics.

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.

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, ACL '06, pages 505–512, Sydney, Australia. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4364–4373, Hong Kong, China. Association for Computational Linguistics.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: a lite BERT for self-supervised learning of language representations. *ArXiv:1909.11942*.

Michael Lepori and R. Thomas McCoy. 2020. Picking BERT's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521– 535.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL '19, pages 1073–1094, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection.

Houman Mehrafarin, Sara Rajaee, and Mohammad Taher Pilehvar. 2022. On the importance of data size in probing fine-tuned models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 228–238, Dublin, Ireland. Association for Computational Linguistics.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Julian Michael, Jan A. Botha, and Ian Tenney. 2020. Asking without telling: Exploring latent ontologies in contextual representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 6792–6812, Online. Association for Computational Linguistics.

Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 68–82, Online. 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.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. 2020. Med-bert: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Peter Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Computer Speech and Language*, 77(C):101429.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2021. Neuron-level interpretation of deep NLP models: A survey. *CoRR*, abs/2108.13138.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Rafae Khan, and Jia Xu. 2022. Analyzing encoded concepts in transformer language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '22, Seattle, Washington, USA. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

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.

Robert L. Thorndike. 1953. Who belongs in the family. *Psychometrika*, pages 267–276.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Alberto Valdes, Jesus Lopez, and Manuel Montes. 2021. UACH-INAOE at SMM4H: a BERT based approach for classification of COVID-19 Twitter posts. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 65–68, Mexico City, Mexico. Association for Computational Linguistics.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Yiyun Zhao and Steven Bethard. 2020. How does BERT's attention change when you fine-tune? an analysis methodology and a case study in negation scope. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4729–4747, Online. Association for Computational Linguistics.

Appendix

A Linguistic Concepts

We used parts-of-speech tags (48 concepts) using Penn Treebank data (Marcus et al., 1993), semantic tags (73 concepts) (Abzianidze et al., 2017), chunking tags (Tjong Kim Sang and Buchholz, 2000) (22 concepts) and CCG super tags (1272 concepts). Please see all the concepts below. This provides a good coverage of linguistic concepts including morphology, syntax and semantics.

#	Tag	Description
1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential there
5	FW	Foreign word
6	IN	Preposition or subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRP$	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol
25	TO	to
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund or present participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd person singular present
32	VBZ	Verb, 3rd person singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRB	Wh-adverb
37	#	Pound sign
38	$	Dollar sign
39	.	Sentence-final punctuation
40	,	Comma
41	:	Colon, semi-colon
42	(	Left bracket character
43	)	Right bracket character
44	"	Straight double quote
45	’	Left open single quote
46	"	Left open double quote
47	’	Right close single quote
48	"	Right close double quote

Table 3: Penn Treebank POS tags.

Chunking tags: NP (Noun phrase), VP (Verb phrase), PP (Prepositional phrase), ADVP (Adverb phrase), SBAR (Subordinate phrase), ADJP (Adjective phrase), PRT (Particles), CONJP (Conjunction), INTJ (Interjection), LST (List marker), UCP (Unlike coordinate phrase). For the annotation, chunks are represented using IOB format, which results in 22 tags in the dataset as reported in Table 5.

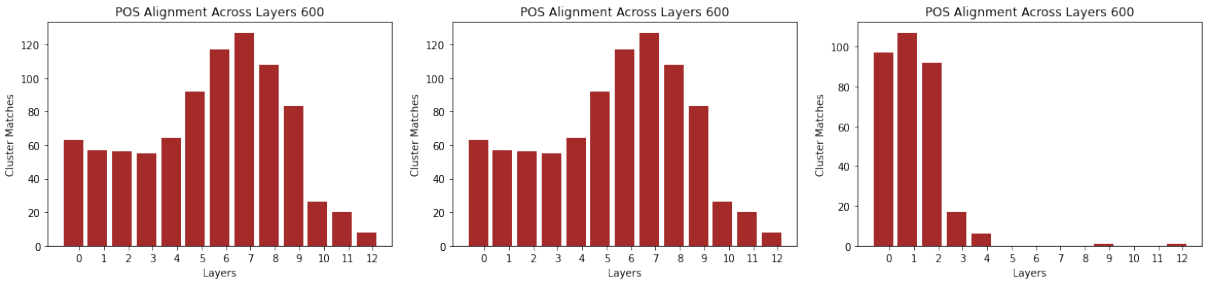
A.1 BERT-based Sequence Tagger

We trained a BERT-based sequence tagger to auto-annotate our training data. We used standard splits for training, development and test data for the 4 linguistic tasks (POS, SEM, Chunking and CCG super tagging) that we used to carry out our analysis on. The splits to preprocess the data are available through git repository[12] released with Liu et al. (2019). See Table 5 for statistics and classifier accuracy.

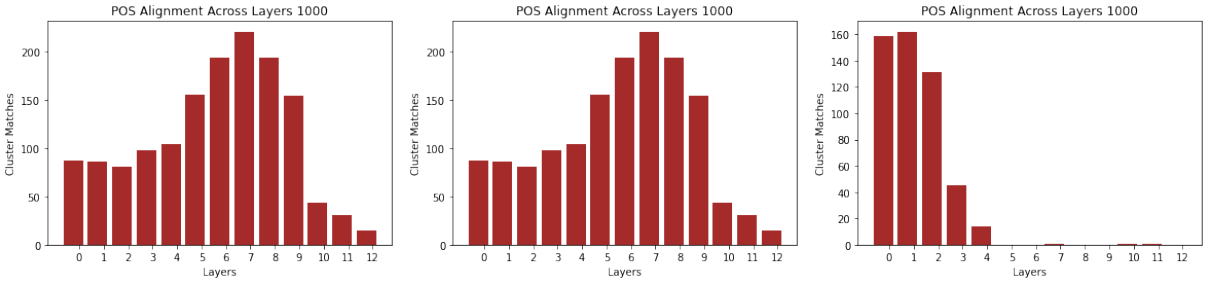
B Selection of the number of Clusters

We tried Elbow and Silhouette to get the optimum number of clusters, but did not observe any reliable patterns. In Elbow the distortion scores kept increasing, resulting in over-clustering (a large number of clusters consisted of less than 5 words). Over-clustering results in high but wrong alignment scores e.g. consider a two word cluster having words “bad” and “worse”. The cluster will have a successful match with “adjective” since more than 95% of the words in the cluster are adjectives. In this way, a lot of small clusters will have a successful match with many human-defined concepts and the resulting alignment scores will be high. On the other hand, Silhouette resulted in under-clustering, giving the best score at number of clusters = 10. We handled this empirically by trying several values for the number of clusters i.e., 200 to 1600 with step size 200. We selected 600 to find a good balance with over and under clustering. We understand that this may not be the best optimal point. We presented the results of 600 and 1000 clusters to show that our findings are not sensitive to the number of clusters parameter. Please See Figures 9 and 8 for comparison.

[12] https://github.com/nelson-liu/contextual-repr-analysis

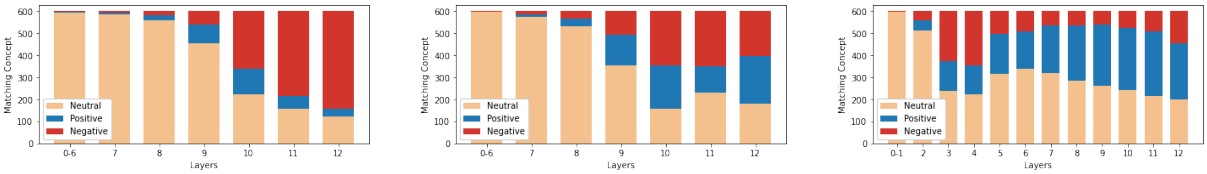


(a) BERT – SST (600) (b) XLM-R – SST (600) (c) ALBERT – SST (600)



(d) BERT – SST (1000) (e) XLM-R – SST (1000) (f) ALBERT – SST (1000)

Figure 8: Comparing encoded concepts when using 600 or 1000 clusters



(a) BERT – SST (600) (b) XLM-R – SST (600) (c) ALBERT – SST (600)

(d) BERT – SST (1000) (e) XLM-R – SST (1000) (f) ALBERT – SST (1000)

Figure 9: Aligning encoded concepts with the task specific concepts

C Analysis

C.1 Comparing Base and Fine-tuned Models

In Section 4.1 we showed the overlap between the encoded concepts of base and fine-tuned SST models. In Figures 10 and 11 we show the same for MNLI and Hate Speech models. We also report the results on how the concepts evolve across the layers. We found that lower layers of the model (until layer 5) show a substantial overlap (up to 40% overlapping clusters). Higher layers show less than 10% overlap. Please See Figure 12 for this result.

C.2 Presence of Linguistic Concepts in the Latent Space

In Section 4.2 we showed the overlap of the encoded concepts in the base and fine-tuned SST models with human-defined POS concepts. In Figures 13-19, we provide alignment results for SEM, CCG and Chunking concepts with SST and also MNLI tasks.

C.3 Task-specific Latent Spaces

In Section 4.3 we studied how the concepts in SST models acquire polarity towards the task. We did not show the base models due to space limitations. Here we show the base models as well to demonstrate that all concepts had no polarity in the base

ANA (anaphoric)		**MOD (modality)**	
PRO	anaphoric & deictic pronouns: he, she, I, him	NOT	negation: not, no, neither, without
DEF	definite: the, loIT, derDE	NEC	necessity: must, should, have to
HAS	possessive pronoun: my, her	POS	possibility: might, could, perhaps, alleged, can
REF	reflexive & reciprocal pron.: herself, each other	**DSC (discourse)**	
EMP	emphasizing pronouns: himself	SUB	subordinate relations: that, while, because
ACT (speech act)		COO	coordinate relations: so, {,}, {;}, and
GRE	greeting & parting: hi, bye	APP	appositional relations: {,}, which, {(}, —
ITJ	interjections, exclamations: alas, ah	BUT	contrast: but, yet
HES	hesitation: err	**NAM (named entity)**	
QUE	interrogative: who, which, ?	PER	person: Axl Rose, Sherlock Holmes
ATT (attribute)		GPE	geo-political entity: Paris, Japan
QUC	concrete quantity: two, six million, twice	GPO	geo-political origin: Parisian, French
QUV	vague quantity: millions, many, enough	GEO	geographical location: Alps, Nile
COL	colour: red, crimson, light blue, chestnut brown	ORG	organization: IKEA, EU
IST	intersective: open, vegetarian, quickly	ART	artifact: iOS 7
SST	subsective: skillful surgeon, tall kid	HAP	happening: Eurovision 2017
PRI	privative: former, fake	UOM	unit of measurement: meter, $, %, degree Celsius
DEG	degree: 2 meters tall, 20 years old	CTC	contact information: 112, info@mail.com
INT	intensifier: very, much, too, rather	URL	URL: http://pmb.let.rug.nl
REL	relation: in, on, 's, of, after	LIT	literal use of names: his name is John
SCO	score: 3-0, grade A	NTH	other names: table 1a, equation (1)
COM (comparative)		**EVE (events)**	
EQU	equative: as tall as John, whales are mammals	EXS	untensed simple: to walk, is eaten, destruction
MOR	comparative positive: better, more	ENS	present simple: we walk, he walks
LES	comparative negative: less, worse	EPS	past simple: ate, went
TOP	superlative positive: most, mostly	EXG	untensed progressive: is running
BOT	superlative negative: worst, least	EXT	untensed perfect: has eaten
ORD	ordinal: 1st, 3rd, third	**TNS (tense & aspect)**	
UNE (unnamed entity)		NOW	present tense: is skiing, do ski, has skied, now
CON	concept: dog, person	PST	past tense: was baked, had gone, did go
ROL	role: student, brother, prof., victim	FUT	future tense: will, shall
GRP	group: John {,} Mary and Sam gathered, a group of people	PRG	progressive: has been being treated, aan hetNL
DXS (deixis)		PFT	perfect: has been going/done
DXP	place deixis: here, this, above	**TIM (temporal entity)**	
DXT	temporal deixis: just, later, tomorrow	DAT	full date: 27.04.2017, 27/04/17
DXD	discourse deixis: latter, former, above	DOM	day of month: 27th December
LOG (logical)		YOC	year of century: 2017
ALT	alternative & repetitions: another, different, again	DOW	day of week: Thursday
XCL	exclusive: only, just	MOY	month of year: April
NIL	empty semantics: {.}, to, of	DEC	decade: 80s, 1990s
DIS	disjunction & exist. quantif.: a, some, any, or	CLO	clocktime: 8:45 pm, 10 o'clock, noon
IMP	implication: if, when, unless		
AND	conjunction & univ. quantif.: every, and, who, any		

Table 4: Semantic tags.

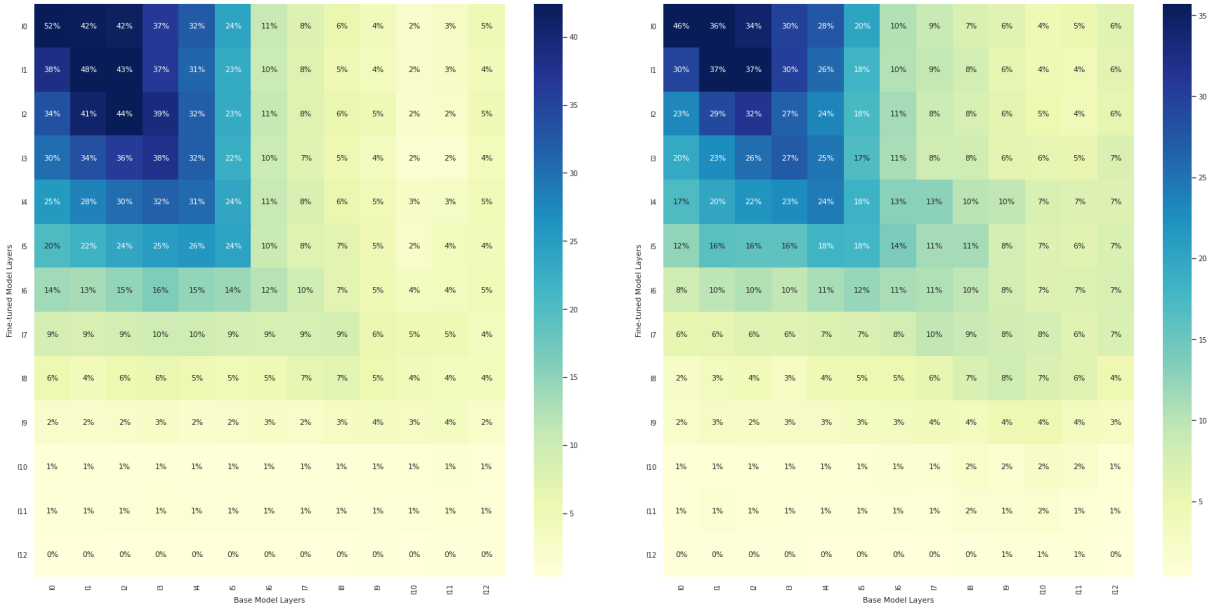
Task	Train	Dev	Test	Tags	F1
POS	36557	1802	1963	48	96.81
SEM	36928	5301	10600	73	96.32
Chunking	8881	1843	2011	22	96.93
CCG	39101	1908	2404	1272	95.24

Table 5: Data statistics (number of sentences) on training, development and test sets using in the experiments and the number of tags to be predicted

models. In Figure 21, we show the same for the Hate-Speech task. We do not show the MNLI task, because we could not find polarity concepts in that task.

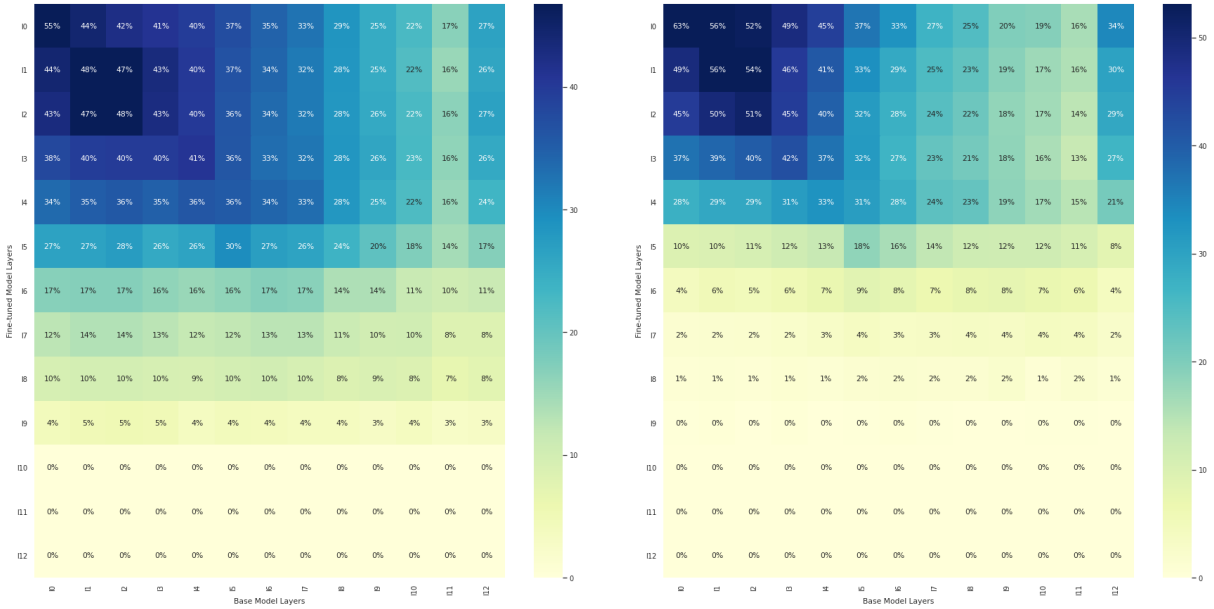
D Selection of task-specific Latent clusters

Figure 22 shows some task-specific latent clusters from various models and layers.



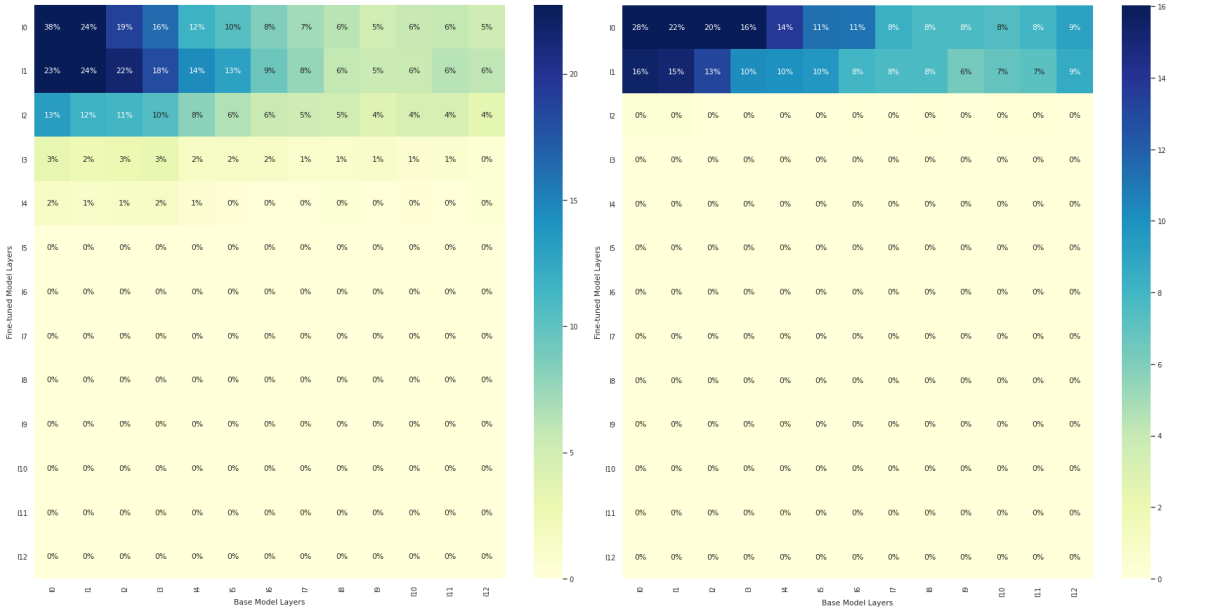
(a) BERT (SST)

(b) BERT (MNLI)



(c) XLM-R (SST)

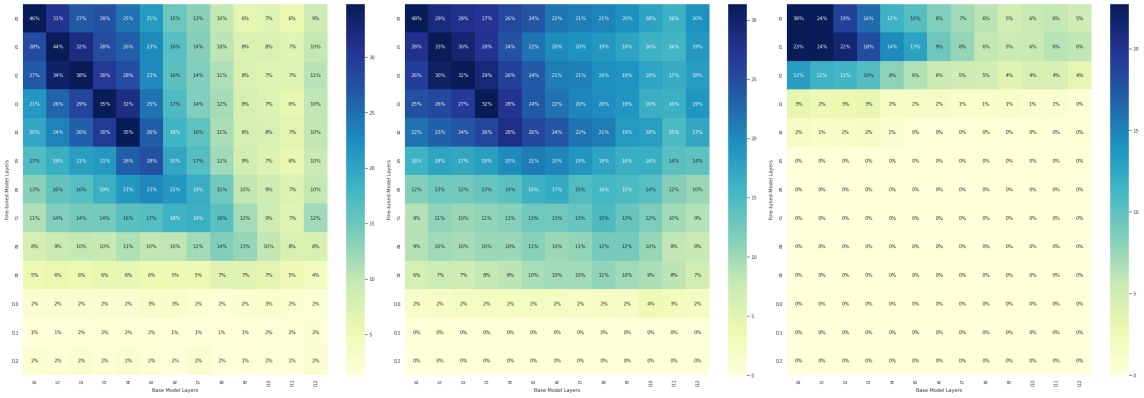
(d) XLM-R (MNLI)



(e) ALBERT (SST)

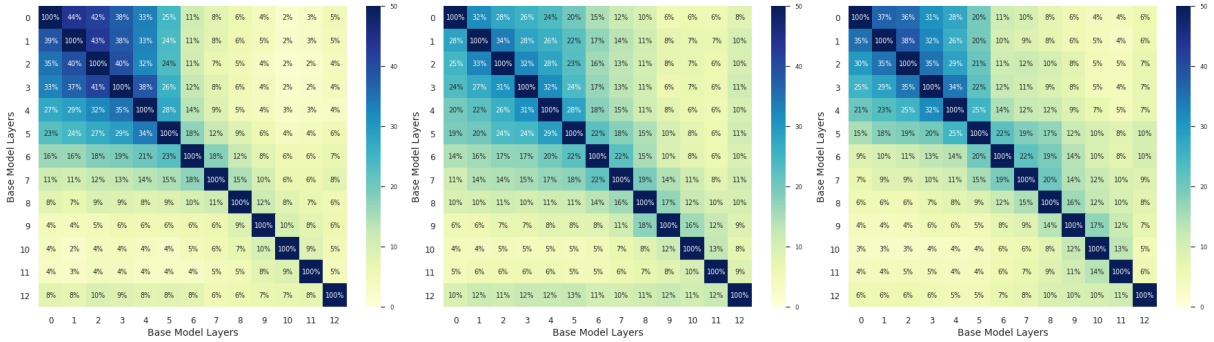
(f) ALBERT (MNLI)

Figure 10: Comparing Latent Concepts of Base models with their SST and MNLI fine-tuned versions. X-axis = base model, Y-axis = fine-tuned model



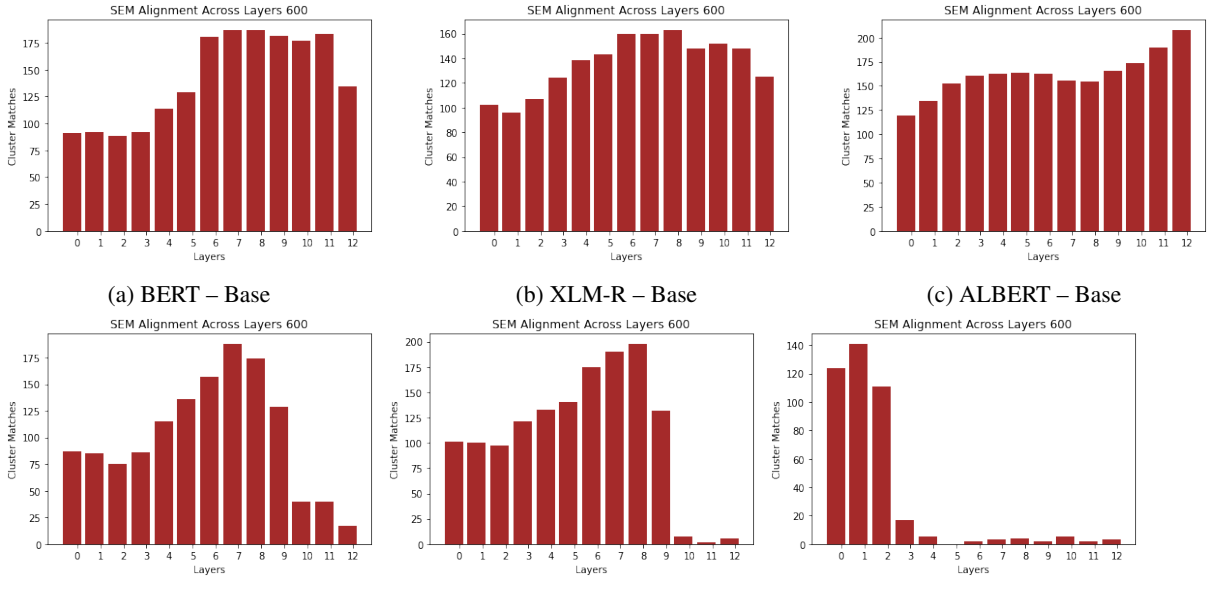
(a) BERT (HS) (b) XLM-R (HS) (c) ALBERT (HS)

Figure 11: Comparing Latent Concepts of Base models with their Hate Speech fine-tuned versions. X-axis = base model, Y-axis = fine-tuned model



(a) BERT (SST) (b) XLM-R(SST) (c) ALBERT (SST)

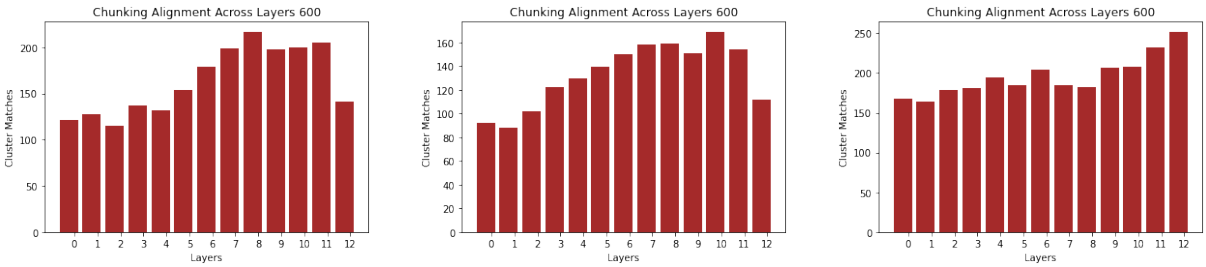
Figure 12: Comparing Latent Concepts of Base models with themselves. X-axis = base model, Y-axis = fine-tuned model



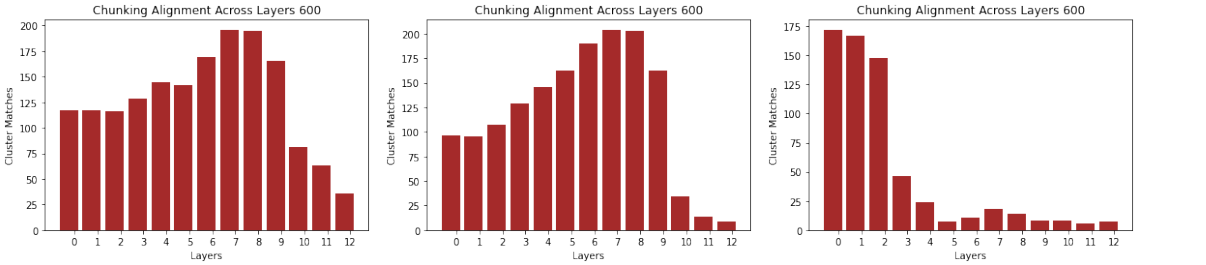
(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

(d) BERT – SST (e) XLM-R – SST (f) ALBERT – SST

Figure 13: Aligning encoded concepts with human-defined concept (SEM) in base and pre-trained models

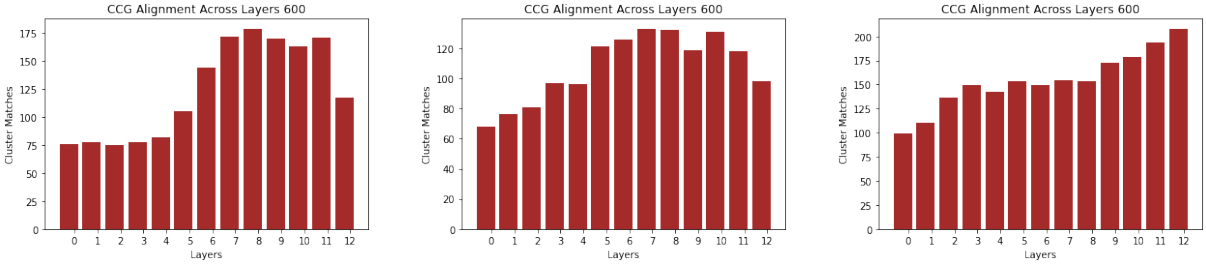


(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

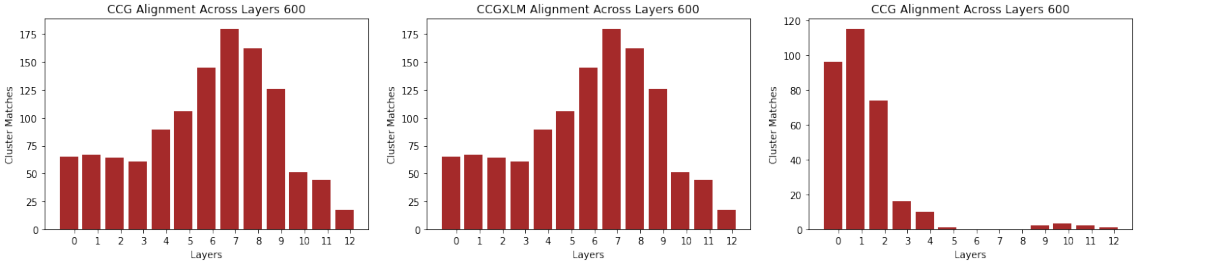


(d) BERT – SST (e) XLM-R – SST (f) ALBERT – SST

Figure 14: Aligning encoded concepts with human-defined concept (Chunking) in base and pre-trained models

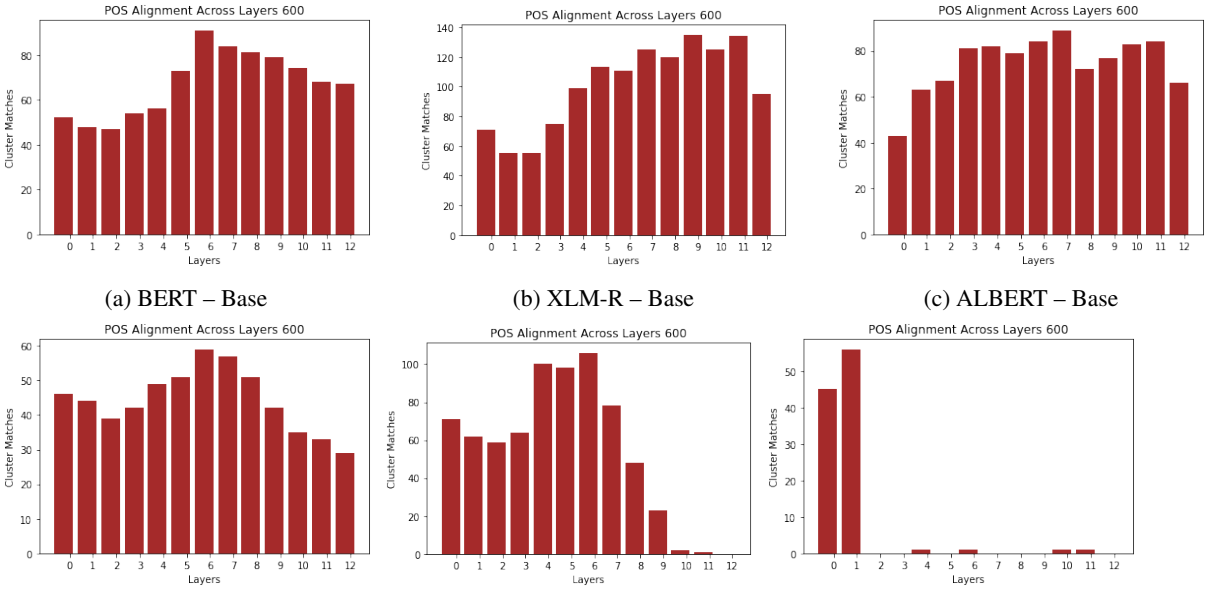


(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base



(d) BERT – SST (e) XLM-R – SST (f) ALBERT – SST

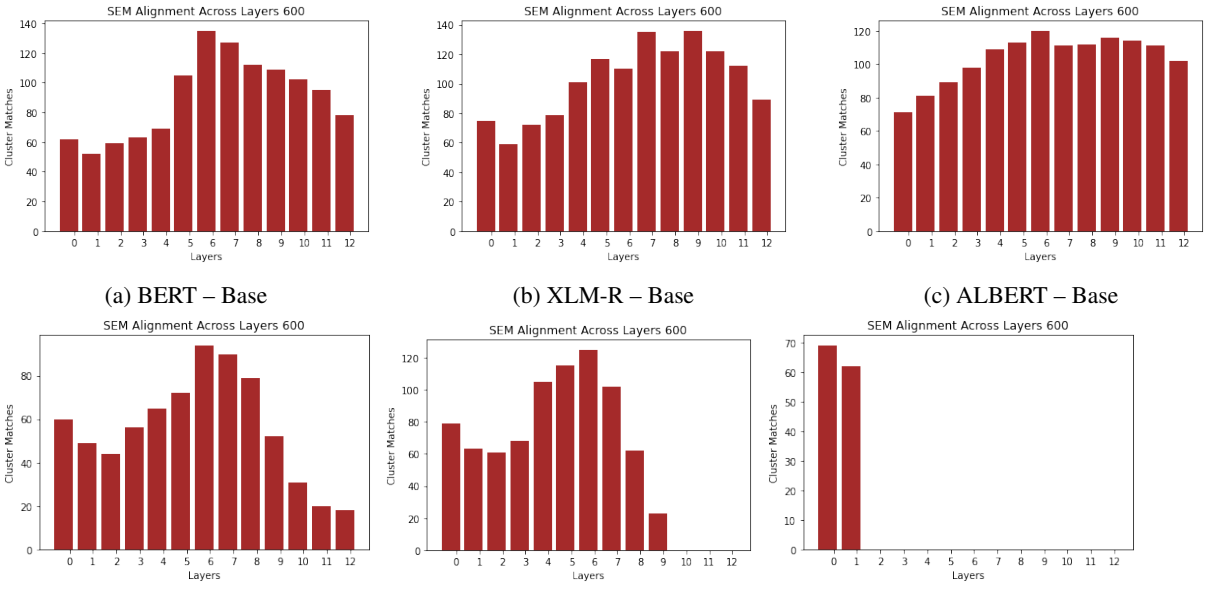
Figure 15: Aligning encoded concepts with human-defined concept (CCG) in base and pre-trained models



(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

(d) BERT – MNLI (e) XLM-R – MNLI (f) ALBERT – MNLI

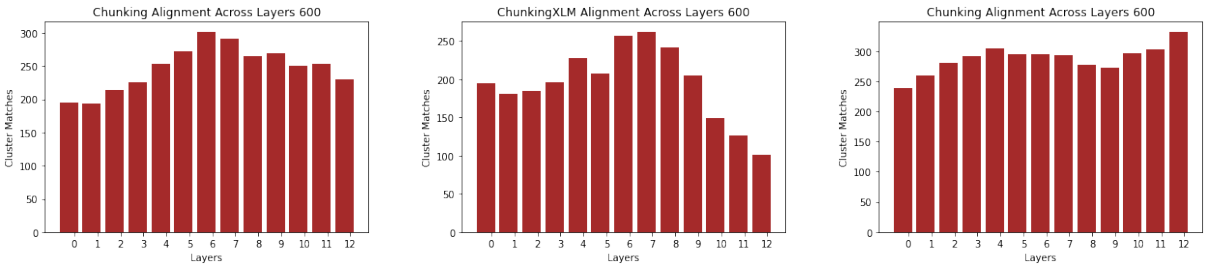
Figure 16: Aligning encoded concepts with human-defined concept (POS) in base and pre-trained models



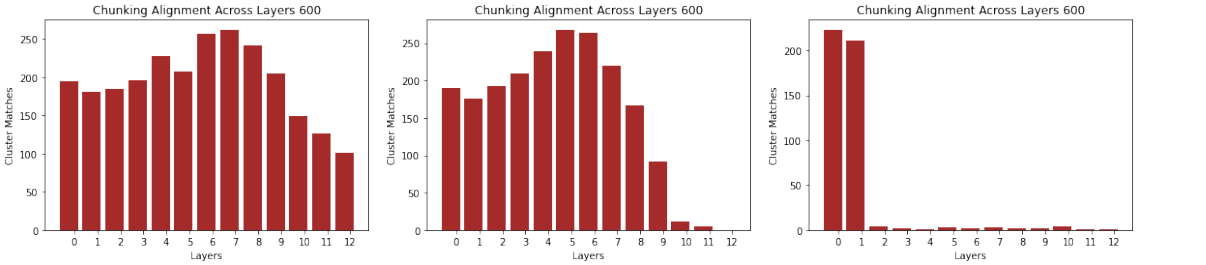
(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

(d) BERT – MNLI (e) XLM-R – MNLI (f) ALBERT – MNLI

Figure 17: Aligning encoded concepts with human-defined concept (SEM) in base and pre-trained models

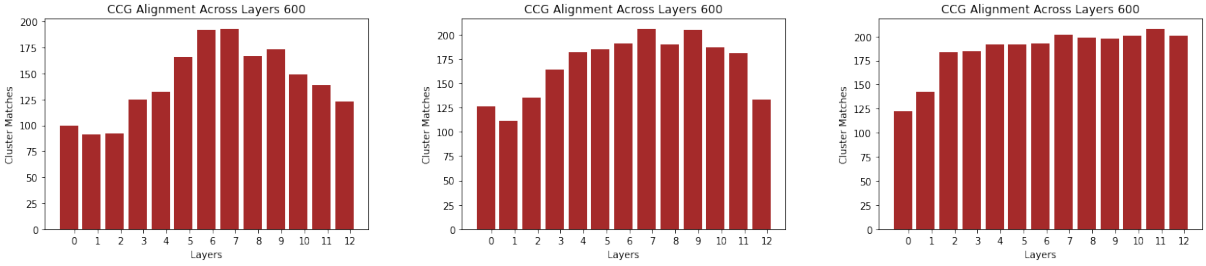


(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

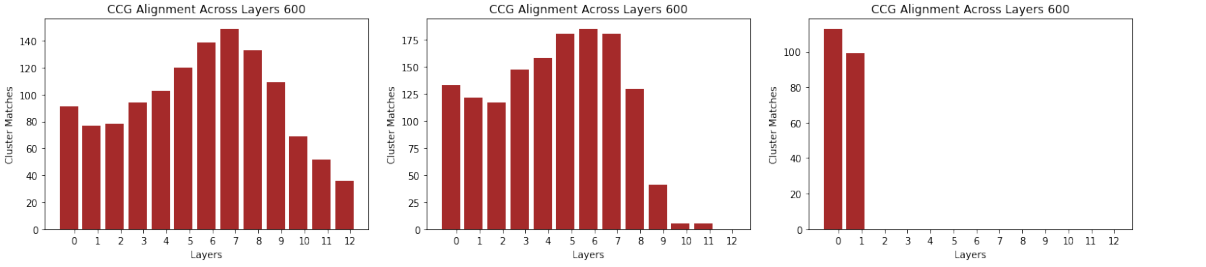


(d) BERT – MNLI (e) XLM-R – MNLI (f) ALBERT – MNLI

Figure 18: Aligning encoded concepts with human-defined concept (CHUNKING) in base and pre-trained models

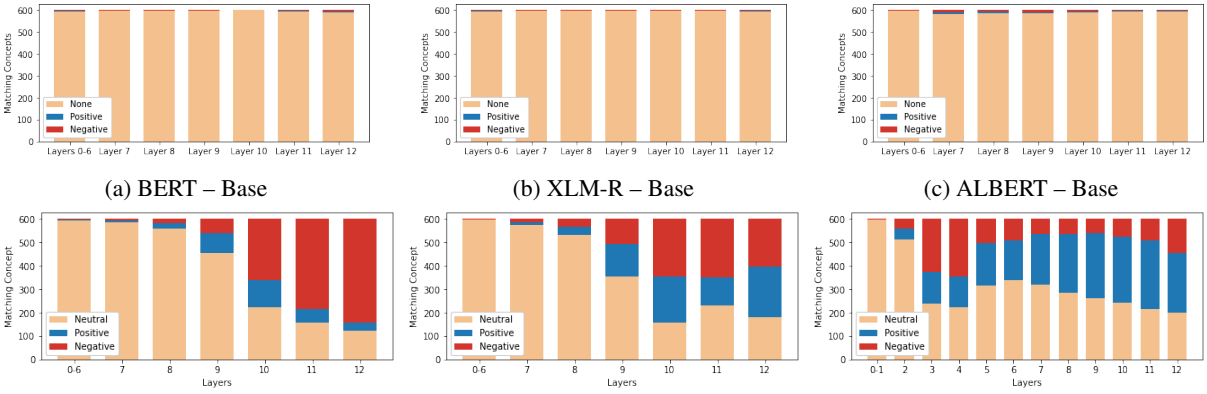


(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base



(d) BERT – MNLI (e) XLM-R – MNLI (f) ALBERT – MNLI

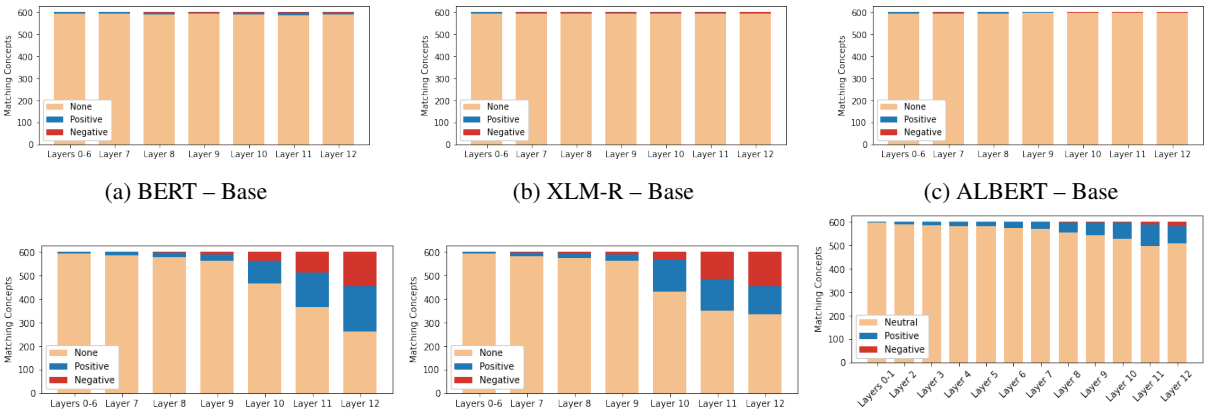
Figure 19: Aligning encoded concepts with human-defined concept (CCG) in base and pre-trained models



(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

(d) BERT – SST (e) XLM-R – SST (f) ALBERT – SST

Figure 20: Aligning encoded concepts with the task specific concepts



(a) BERT – Base (b) XLM-R – Base (c) ALBERT – Base

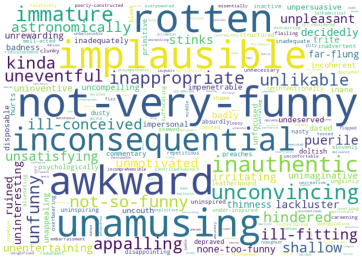
(d) BERT – HS (e) XLM-R – HS (f) ALBERT – HS

Figure 21: Aligning encoded concepts with the task specific (Hate Speech: toxic vs. non-toxic) concepts. Positive = Toxic, Negative = Non-Toxic

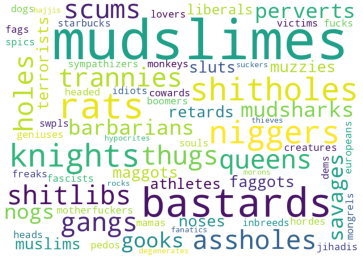
(a) XLM-R Layer 12 Cluster 470 (Positive Sentiment)

(b) XLM-R Layer 12 Cluster 15 (Positive Sentiment)

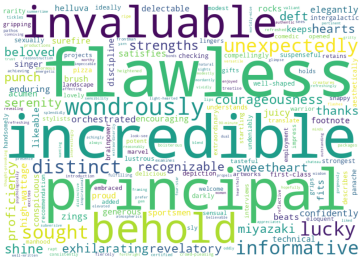
(c) XLM-R Layer 10 Cluster 16 (Negative Sentiment)



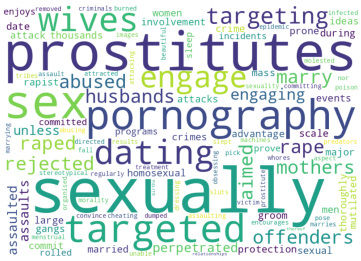
(d) XLM-R Layer 10 Cluster 121 (Negative Sentiment)



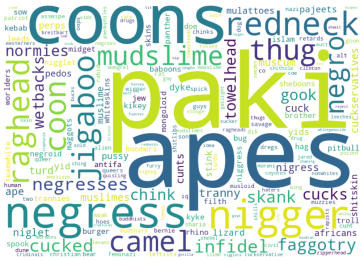
(e) XLM-R Layer 10 Cluster 576 (Toxic Hatespeech)



(f) BERT Layer 12 Cluster 432 (Positive Sentiment)



(g) BERT Layer 12 Cluster 272 (Toxic Hatespeech)



(h) BERT Layer 10 Cluster 227 (Toxic Hatespeech)

(i) ALBERT Layer 11 Cluster 489 (Positive Sentiment)

(j) ALBERT Layer 11 Cluster 215 (Negative Sentiment)

Figure 22: Task-specific latent clusters from various models and layers