Deriving Contextualised Semantic Features from BERT (and Other Transformer Model) Embeddings

Jacob Turton
Department of Computer Science
UCL
j.turton@cs.ucl.ac.uk

David Vinson
Department of Psychology and Language Sciences
UCL
d.vinson@ucl.ac.uk

Robert Elliott Smith
Department of Computer Science
UCL
rob.smith@cs.ucl.ac.uk

Abstract

Models based on the transformer architecture, such as BERT, have marked a crucial step forward in the field of Natural Language Processing. Importantly, they allow the creation of word embeddings that capture important semantic information about words in context. However, as single entities, these embeddings are difficult to interpret and the models used to create them have been described as opaque. Binder and colleagues proposed an intuitive embedding space where each dimension is based on one of 65 core semantic features. Unfortunately, the space only exists for a small data-set of 535 words, limiting its uses. Previous work (Utsumi, 2018, 2020; Turton et al., 2020) has shown that Binder features can be derived from static embeddings and successfully extrapolated to a large new vocabulary. Taking the next step, this paper demonstrates that Binder features can be derived from the BERT embedding space. This provides two things; (1) semantic feature values derived from contextualised word embeddings and (2) insights into how semantic features are represented across the different layers of the BERT model.

1 Introduction

The last decade or so has seen a rapid progress in the field of Natural Language Processing (NLP) with a combination of new models and increasingly powerful hardware resulting in state of the art performances across a number of common tasks (Wang et al., 2020). One important area of improvement has been in the vector-space representation of words, known as word embeddings. Embedding models create word vectors within a vector space that captures important semantic and grammatical information (Boleda, 2020). Models such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) were popular in the 2010s, but are static, meaning only one embedding is produced for each word. In reality words can have multiple meanings; 7% of common English word forms have homonyms and over 80% are polysemous (Rodd et al., 2002).

Deep learning language models such as ELMO: Embeddings from Language Models (Peters et al., 2018) addressed this issue, using deep neural-network language models to incorporate context and produce contextualised embeddings. Following this, the introduction of the transformer architecture and in particular its implementation in the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) model, resulted in even better performing contextual embeddings.

Regardless of whether the embeddings mentioned are static or contextual, they all have the issue that, as individual objects, they are hard to interpret (Şenel et al., 2018). Whilst efforts have been made to produce more interpretable embeddings e.g. (Şenel et al., 2020; Panigrahi et al., 2019), the general approach has been to interpret them in relation to each-other. For example, the relative distance between word embeddings can indicate their semantic similarity (Schnabel et al., 2015). Alternatively, dimensionality reduction techniques can be used to visualise where the words sit within the embedding space (Liu et al., 2017). However, these methods may just show how the embeddings are related, rather than why, further feeding into the general criticism levelled at deep learning architectures; that they are opaque and difficult to interpret (Belinkov and Glass, 2019).

Binder et al. (2016) presented an alternative embedding space for words, based on 65 core semantic features, where each dimension relates to a feature. Unfortunately, the Binder dataset only contains 535 words, severely limiting its use for large scale text analysis. Previous research (Utsumi, 2018, 2020;

Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pages 248–262
Bangkok, Thailand (Online), August 6, 2021.

Turton et al., 2020) has shown that the Binder feature values can be derived from static embeddings, such as Word2Vec, and successfully extrapolated to a large new vocabulary of words. The purpose of this research is to demonstrate that Binder features can be successfully derived from BERT embedding space allowing the features to be derived from contextual embeddings. Along the way, this also provided the opportunity to study how different types of semantic information are represented across the different layers of the BERT model.

2 Related Work

2.1 Probing Transformer Models

Whilst transformer models such as BERT have led to impressive improvements in NLP tasks, alongside other deep learning models they have been criticised as opaque "black boxes" that are difficult to interpret (Castelvecchi, 2016). To address this researchers have made efforts to better understand how they work. For example, Clark et al. (2019) were able to show that patterns of attention in BERT respond to certain syntactic relations between words. Other work has looked at how semantic information is represented in BERT. Researchers have shown that BERT can learn to represent semantic roles (Ettinger, 2020), entity types and semantic relations (Tenney et al., 2019). Reif et al. (2019) demonstrated clear 'clusters' for different senses of the same word, when visualising the spatial location of their BERT embeddings. Jawahar et al. (2019) demonstrated that embeddings from different layers of BERT performed better at different tasks, with semantic information tending to be better represented by the later layers. Whilst these studies provide important insights into the inner workings of transformer models, they do little to improve interpretability of individual word embeddings extracted from them.

2.2 Interpretable Word Embeddings

Research has also been done to produce more interpretable static word embeddings e.g. (Şenel et al., 2020; Panigrahi et al., 2019). For contextual embeddings, Aloui et al. (2020) produced embeddings with semantic super-senses as dimensions, but these are quite broad. The embedding space of Binder et al. (2016) offers a more fine-grained representation of semantics, but there are challenges in applying it to contextualised word embeddings.

2.3 Binder Semantic Features

Through a meta-analysis, Binder et al. (2016) identified 65 semantic features all believed to, and some demonstrated to, have neural correlates within the brain. They produced a 535 word data-set scored by participants across the 65 features. The features ranged from concrete object properties such as visual and auditory, to more abstract properties such as emotional aspects. This resulted in a 65-dimensional embedding for each word, where each dimension relates to a specific semantic feature.

This embedding space is useful as each dimension is easily interpretable and theoretically connected to a specific aspect of how people understand the meaning of words and concepts. Furthermore, representing words in this way makes it easy to understand how they are similar or different in terms of their semantic features. Figure 1 below demonstrates this by comparing the feature scores of the words *raspberry* and *mosquito*. It shows how the concepts differ across a range of dimensions. Also, since these features are derived from the psychological and neuroscience literature, it may mirror how people differentiate these concepts.

Unfortunately, the Binder dataset only exists for 535 words, which severely limits its uses. However, previous work (Utsumi, 2018, 2020) has shown that Binder feature values can be derived from static word embeddings such as Word2Vec and this can be used to extrapolate the feature space to a large

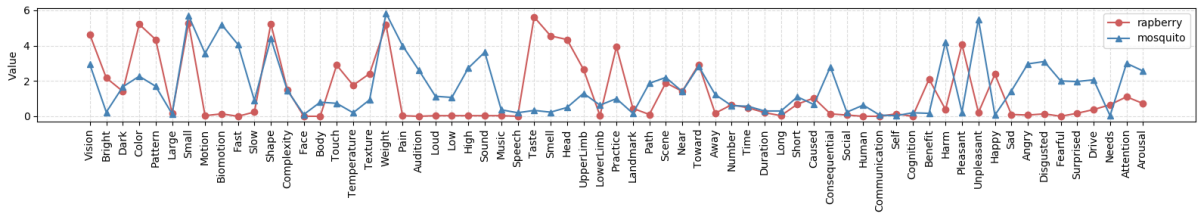


Figure 1: Binder feature values for *raspberry* and *mosquito*.

number of new words (Turton et al., 2020). Being able to do this using BERT embeddings would allow the features to be derived for words in context. Not only would this tackle the issues of polysemy and homonymy, but hopefully also mirror more subtle differences between words when used in context. Beyond this, the dataset also offers a powerful way to probe the semantic representation of words in models like BERT, by looking at: how well the different semantic features can be predicted overall, how the semantic representations build over the layers of the models and whether there are distinct patterns in how different types of semantic feature are represented across the layers.

3 Experiment 1a: Deriving Binder Embeddings from BERT and other Transformer Model Embeddings

3.1 Introduction

The aim of the first experiment was to derive Binder feature scores from the BERT embedding space. Words in the Binder dataset are presented out of context so the BERT embeddings were treated as *static* by taking an the average embedding over 250 randomly sampled sentences for each word. A selection of alternative transformer models were included for comparison: RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019) and GPT-2 (Radford et al.). Numberbatch embeddings (the best performing static embeddings from Turton et al. (2020)) (Speer et al., 2017) were used as a baseline comparison.

This experiment also offered the opportunity to investigate how different semantic features are represented across the different layers of BERT.

3.2 Materials

The Binder et al. (2016) data-set was used, providing scores across the 65 features for 535 words. For random sentences containing the Binder words, the One Billion Word Benchmark (BWB) (Chelba et al., 2014) was used. Author provided pre-trained versions of each transformer model were used. As far as possible, models of the same size were selected (see Appendix Table a for further details). Pre-trained Numberbatch embeddings were also used (Speer et al., 2017) as a benchmark. A simple 4 hidden-layer (300,200,100,50) neural network was used to predict semantic features from embeddings.

3.3 Method

The method here describes the process for the $\text{BERT}_{\text{BASE}}$ model, but was the same for all other models as well.

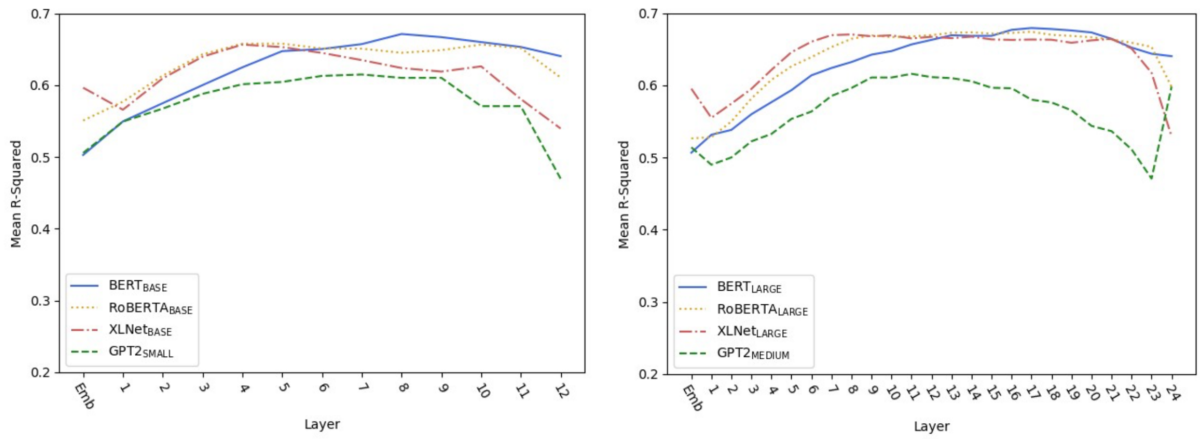
To produce static embeddings for each of the Binder words, 250 sentences containing each one were randomly sampled from the BWB dataset. Then using the pre-trained $\text{BERT}_{\text{BASE}}$ model the embeddings from all 12 layers (24 for large models) and the embedding layer were extracted for the target word for each of the sentences. A mean of the target word embedding across the 250 sentences was then taken. Additionally, for each model the best performing sub-word approach was used (see Table b and Figure a in Appendix for comparisons).

Semantic feature scores were predicted by feeding the extracted embeddings into a feed-forward neural network. 10-fold validation was used across the data-set and the final R-squared score averaged across the folds. Each of the 65 features was evaluated separately as was each of the layers. A Wilcoxon Ranks-sums test (Demšar, 2006) was used to compare performance of the different embedding models.

To investigate how the different semantic features are represented across the layers, each feature's R-squared score was re-scaled between 0-1 across the layers. A k-means clustering algorithm was then used to group the features according to similar patterns across the layers. The re-scaling ensured it was the pattern of behaviour across the layers rather than the absolute performance of each feature that was captured in the clustering. The membership of the clusters was compared to the categories of the features given in Binder data-set using the Adjusted Rand Index (Yeung and Ruzzo, 2001).

3.4 Results

Figure 2 below shows the mean R-squared scores across all semantic features for the different layers for the large and small models. The models showed slightly different performance across the layers with XLNet and RoBERTa peaking earlier than BERT. As per Table 1 row 2, BERT had the best performing single layer for both model sizes. Table 1 row 1 (combined) shows the performance of the models combining the best performing layer for each semantic feature. All models except $\text{GPT-2}_{\text{SMALL}}$ significantly outperformed the Numberbatch baseline ($p<0.05$ for all). $\text{BERT}_{\text{BASE}}$



(a) (b)

Figure 2: Mean R-squared scores across all semantic features for layers of (a) small and (b) large models.

MEAN R-SQUARED	NumbrBatch	**MODEL** GPT-2		RoBERTa		XL-Net		BERT	
		Small	*Med.*	*Base*	*Large*	*Base*	*Large*	*Base*	*Large*
Combined	-	.631	.638	.673	**.692**	.665	.688	**.678**	.692
Best Layer	.646	.615	.616	.658	.674	.656	.670	**.667**	**.679**

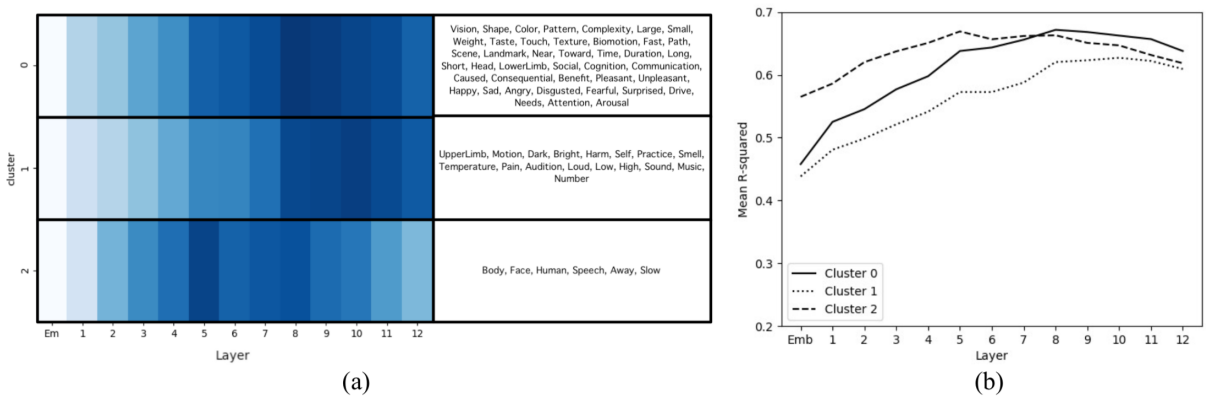
Table 1: Best overall mean R-squared scores for the models across all 65 semantic features

also outperformed XLNet$_{BASE}$ ($p<0.05$) but not RoBERTa$_{BASE}$ ($p=0.17$).

There was variation in how well different features were predicted from the embeddings (some as low as 0.3 with others over 0.8) (See Figure b in the Appendix for full results). There was also general consistency between the models as to which features were well and poorly predicted with inter-feature variance (mean=0.011) larger than inter-model variance (mean=0.001). This indicates certain semantic features are difficult to predict regardless of the model.

For all models the larger version performed significantly better than the base version ($p<0.05$ for all). For the larger models there was no longer any significant difference between the BERT$_{LARGE}$, RoBERTa$_{LARGE}$ and XLNet$_{LARGE}$ models ($p>0.05$ for all), but all three did outperform GPT-2$_{MEDIUM}$ ($p>0.05$ for all).

The k-means clustering on the re-scaled BERT$_{BASE}$ R-squared scores indicated an optimal 3 clusters identified using an elbow plot. Figure 3 (a) below shows the memberships of the k-means clusters, along with their respective mean scores



(a) (b)

Figure 3: (a) mean re-scaled R-squared scores for the three clusters with member features and (b) mean layer raw R-squared scores for the three clusters.

across each layer. Cluster 0 and 1 show a similar pattern showing a peak in the later layers. Cluster 2 shows a very different pattern with the peak much earlier in the mid-layers. Figure 3 (b) shows the mean raw R-squared layer scores for the different clusters. Clusters 0 2 achieve higher max scores than cluster 1. Whilst this does suggest different patterns of representation for the different features in the model, the clusters do not appear to match the categorisation of features given by Binder et al (2016) as the adjusted rand index was 0.02.

3.5 Discussion

The main purpose of this first experiment was to demonstrate that Binder style embeddings can successfully be derived from the BERT (and other similar model) embedding space. The secondary purpose was to explore how the representation of the semantic features varies across the different layers of a $\text{BERT}_{\text{BASE}}$ model. The results demonstrated that Binder features could be derived from BERT embeddings, outperforming static Numberbatch embeddings. This is interesting as Numberbatch embeddings make use of additional human provided information from a concept network, whereas BERT and the other models are purely trained on raw text. This hints towards the power of these bidirectional transformer models in capturing semantic information from word usage alone.

The poor performance of GPT-2 is not surprising due to its uni-directional attention architecture. GPT-2 has shown success when using very large models (up to 1.5B parameters, compared to $\text{BERT}_{\text{LARGE}}$'s 340M). These results highlight the power of the bidirectional architecture used by BERT, XLNet and RoBERTa

Perhaps most interesting results from this experiment are in relation to how the different semantic features are represented across the layers of BERT. In line with the findings of Jawahar et al. (2019), semantic features tended to be better represented by the later layers. However, a small subset of features were better represented by the middle layers. Clustering the features according to these behaviours did not match the Binder categories. However, the Binder categories are not the only way to group the features and there still are some similarities between the features in the different clusters. For example, Cluster 3 appears to capture a number of features (Human, Face, Speech, Body) relating to people and Cluster 2 captures 6 of the 7 features relating to audition.

Variation in how well different features were predicted by the models is more difficult to explain conclusively. On one hand, it may be that certain features are better represented by the transformer models than others. However, there is also variation in the underlying distributions of the different Binder features, with some more equally distributed across the score range than others. For certain features with very unbalanced distributions, this may have had a detrimental effect on their final R-squared score (see Appendix Figure f for residual plot examples).

Further improvements in predictive power may be possible by fine-tuning the transformer models directly on the Binder feature prediction task. For the purposes of this paper extracted embeddings rather than fine-tuning were used as (1) there were concerns over the small dataset size and (2) to keep the models as close as possible to their pre-trained state when comparing them.

4 Experiment 1b: Towards Contextualised Binder Features

4.1 Introduction

Experiment 1a demonstrated that Binder semantic features can be predicted from the BERT (and other model) embedding space, outperforming the best performing static embeddings (Numberbatch). However, the real power of the transformer architecture and its self-attention mechanism, is being able to represent a contextualised form of words (Reif et al., 2019). By treating the embeddings as "static" as in Experiment 1a, the embeddings were limited to an average of the word over many contexts. This may have added noise to the embeddings and consequently reduced performance by including word-senses not matching the sense suggested by the Binder features. Instead, hand selecting sentences that match the word-sense inferred from the Binder feature scores should help reduce this noise and improve performance.

4.2 Material

Same materials as Experiment 1a.

4.3 Method

For each word in the Binder data-set, ten sentences were hand-picked from the 250 randomly selected BWB sentences used in Experiment 1a. Sentences were picked by matching them to the word-sense

MEAN R-SQUARED	MODEL									
	BASELINE		GPT-2		RoBERTa		XL-Net		BERT	
	Base	*Large*	*Small*	*Med.*	*Base*	*Large*	*Base*	*Large*	*Base*	*Large*
Combined	.678	.692	.656	.670	**.736**	**.755**	.707	.730	.725	.741
Best Layer	.667	.679	.638	.643	.723	**.741**	.697	.714	.718	.729

Table 2: Mean R-squared scores for the models using selected sentences vs BERT baseline from Experiment 1a (randomly selected sentences)

inferred from the Binder feature scores. Following this, the exact same method as Experiment 1a was used, this time using the average embedding across the ten hand-selected sentences.

4.4 Results

Table 2 above gives the mean R-squared scores for the models. BERT scores from Experiment 1a are used as a baseline. (Individual feature results can be found in Figure c of the Appendix). Except from GPT-2, all embeddings from Experiment 1b outperformed the baseline from Experiment 1a.

4.5 Discussion

Using hand selected rather than purely randomly selected sentences improved the performance as expected. This was likely due to removing noise from unrelated uses of the word in the averaged embedding. Importantly, this shows to some degree that context can be captured in the derived semantic features as using more appropriate contexts improved performance. However, since the Binder data-set lacks explicit context for its words this experiment still falls short of a true ground-truth test of deriving contextualised semantic features from transformer word embeddings. To investigate how well semantic features can be predicted for words in specific contexts, it is necessary to look at other data-sets.

5 Experiment 2: Predicting Contextualised Features

5.1 Introduction

Together Experiments 1a and 1b demonstrate that semantic features ratings can be derived from transformer embeddings and that introducing some degree of context improves the performance. But the Binder data-set unfortunately lacks explicit context for its words.

An alternative data-set (Van Dantzig et al., 2011) of contextualised semantic features for words in context pairs can be used. In each context pair a property word e.g. *abrasive* is paired an object word e.g. *lava* and participants scored the property word across five semantic features in a similar way to the Binder dataset. In each case, the object should influence the meaning of the property word, in turn influencing its feature scores. Each property is paired with two different objects giving two word-pairs for each property and with different semantic feature scores for each one (see Table 3). By feeding the property-object pairs into the transformer models, the extracted embedding for the property word should capture its specific feature values influenced by its context object word. Since each property word is paired with two different objects, a static version of its embedding can be created by taking the mean of its embeddings across both of its context pairs. If the models successfully capture the specific feature values of the property words in the individual contexts, the individual contextual embeddings should outperform the static property embeddings in predicting semantic feature scores.

Due to its poor performance GPT-2 was dropped and only the better performing LARGE versions of BERT, XLNet and RoBERTa were used.

5.2 Materials

The Van Dantzig et al. (2011) data-set consists of 774 property-object pairs. Each word pair con-

		FEATURE				
PROPERTY	**OBJECT**	Visual	Auditory	Haptic	Gustatory	Olfactory
Abrasive	Lava	3.83	1.27	2.37	0.07	0.46
Abrasive	Sandpaper	3.37	2.35	4.81	0.26	0.09

Table 3: Feature scores for Property word *Abrasive* with its two different Object word pairs.

FEATURE	PROPERTY-MEAN			CONTEXTUALISED		
	BERT	XL-Net	RoBERTa	BERT	XL-Net	RoBERTa
Visual	0.532	0.448	0.456	**0.652**	0.583	0.633
Auditory	0.722	0.668	0.680	**0.793**	0.733	0.772
Haptic	0.556	0.512	0.505	**0.660**	0.616	0.634
Gustatory	0.611	0.531	0.591	0.800	0.704	**0.813**
Olfactory	0.610	0.587	0.597	**0.740**	0.736	0.731
MEAN	0.607	0.549	0.556	**0.729**	0.674	0.717

Table 4: Mean R-squared scores for the five features for mean and contextualised embeddings from the three different models, compared to a Numberbatch baseline.

sists of a property and object word, and has a rating across five semantic features: Visual, Auditory, Haptic, Gustatory and Olfactory. The ratings are between 0-5 for each. The same pre-trained $\text{BERT}_{\text{LARGE}}$, $\text{XL-Net}_{\text{LARGE}}$ and $\text{ROBERTA}_{\text{LARGE}}$ models from Experiment 1a and b were used and the pre-trained Numberbatch embeddings.

5.3 Method

The property-object word pairs were fed into the transformer models as the input sequences and the embedding for the property word was extracted. Embeddings from all 24 layers and the embedding layer were extracted. The different layer embeddings were then fed into a simple 4 hidden-layer (300, 200, 100, 50) neural network for training prediction with each of the five semantic features used separately as the target variable.

For the *Property-mean* condition, for each property word, the extracted embeddings across both of its object context pairs were averaged. For the *contextualised* condition, the extracted property embeddings were left unique for each object context pair.

Like Experiment 1, the data-set was split into ten-folds with 90% of the data for training and the reaming 10% for evaluation. The mean r-squared scores across the ten-folds was calculated for each of the five semantic features.

5.4 Results

Table 4 shows the R-squared scores for the best performing layer from each model. (See Appendix Figure d for per layer results). The contextualised transformer embeddings outperform both the mean transformer embeddings. Overall, the BERT model performed best.

5.5 Discussion

The purpose of experiment 2 was demonstrate the ability to derive contextual semantic features from transformer embeddings. As predicted, the contextual transformer embeddings performed better than the "static" ones. This suggests that, for each context pair, the model representations of the property words were able to capture the specific semantic features as influenced by the object it was paired with. Taking the mean across both object pairs was detrimental for performance as the embedding was no longer unique to the context pair.

Whilst this experiment demonstrates it is possible to derive contextualised semantic features from transformer embeddings, it only involves a small number of features for words in short word-pair contexts. Ideally, we would be able to predict the full 65 semantic features in the Binder embedding space for words contextualised in longer, more natural sequences.

6 Experiment 3: Evaluation of Contextualised Binder Embeddings

6.1 Introduction

Experiment 1a and b demonstrated that Binder features can be derived from various transformer embedding spaces and that some effects of context can be picked up, whilst Experiment 2 demonstrated that the embeddings can be used to derive contextualised semantic features, but for a very limited number of features and only in word-pair sequences. The lingering issue is the lack of a data-set of the full 65 Binder semantic features for words context which would provide a ground-truth test for deriving contextualised semantic features from transformer embeddings.

To address this, this experiment used the word-sense disambiguation (WSD) task as an indirect evaluation of derived semantic features for words

METRIC	Raw $BERT_{LARGE}$	Experiment 2	BERT Binder 1a	BERT Binder 1b
Accuracy	0.68	0.60	0.67	0.67
F1-Score	0.71	0.66	0.70	0.71

Table 5: Accuracy & F1 score of raw BERT & BERT-derived Binder embeddings on the validation set.

in context. WSD is an open problem in NLP where the task is to determine which sense of word is being used in a sequence (Navigli, 2009). Models that perform well on this task are able to separate the different semantic meanings of a word, depending on the context it is used in. By evaluating how well derived Binder embeddings perform at this task, it should indicate how good the embeddings are at representing the contextualised semantic features of the words. In this experiment the Binder embeddings are compared to raw BERT embeddings which have shown reasonable performance in the task (Pilehvar and Camacho-Collados, 2019).

For comparison, the different approaches for deriving Binder embeddings from Experiments 1a and 1b were used as well as the much smaller Van Dantzig feature set from Experiment 2.

6.2 Materials

The Word in Context (WiC) WSD data-set (Pilehvar and Camacho-Collados, 2019) was used. It consists of sentence pairs each containing the same target word and a binary classification (True/False) of whether the target word has the same word-sense or not between them. The data-set is already divided into a training (5429) and separate validation (639) set.

The same $BERT_{LARGE}$ model and trained neural networks from Experiment 1a, 1b and 2 were used to predict semantic feature values.

6.3 Method

Using the pre-trained $BERT_{LARGE}$ model, word embeddings from all 24 layers + the embedding layer were extracted for the target word in each of the sentences of the WiC dataset. Using the neural networks trained in Experiment 1a and 1b the Binder features were predicted using the optimal $BERT_{LARGE}$ layer for each of the 65 features and for the smaller Van Dantzig feature set from Experiment 2.

For each sentence pair, the cosine similarity was calculated between the embeddings for the target words, either using the raw $BERT_{LARGE}$ embeddings or the derived Binder or Van Dantzig embeddings. For evaluation a logistic regression model was used with the cosine similarity scores as input. The model was trained on the train set and evaluated on the validation set using accuracy and F1 Score.

6.4 Results

Table 5 shows the performance of the best performing layer (21) raw $BERT_{LARGE}$ embeddings, Binder and Van Dantzig embeddings on the WiC dev set (see Appendix Figure e for all layer performances). Overall the Binder embeddings performed comparatively to the raw $BERT_{LARGE}$ embeddings. The five feature Van Dantzig embeddings (from Exp. 2) performed worst.

6.5 Discussion

The purpose of this final experiment was to evaluate contextualised Binder embeddings. In the absence of a ground-truth data-set for contextualised Binder features, the WSD task was used as an indirect measure. The contextualised Binder embeddings performed comparatively to raw BERT embeddings

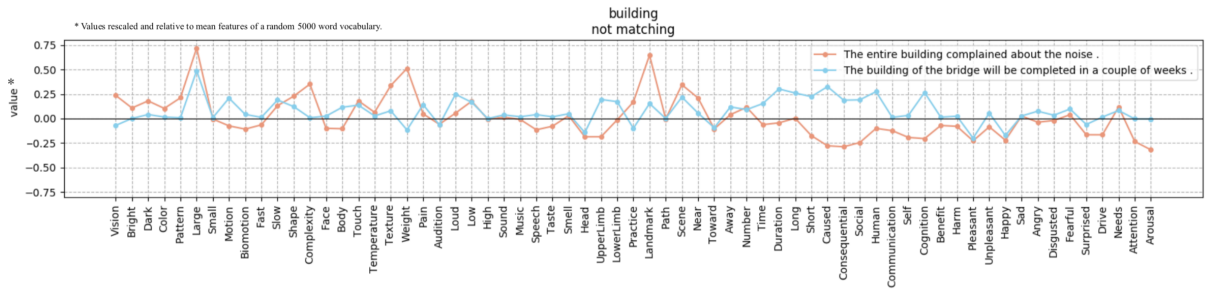


Figure 4: Example of predicted semantic features for the word *building* in two different context sentences

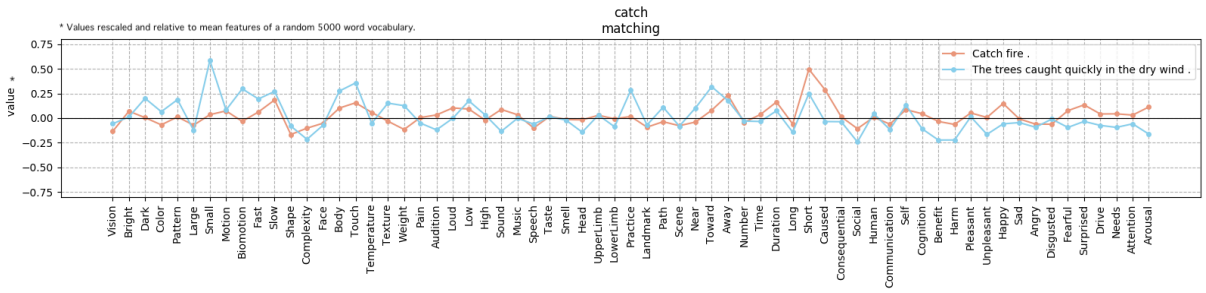


Figure 5: Example of predicted semantic features for the word *catch* in two different context sentences

which have been shown to capture contextualised semantics (Reif et al., 2019; Pilehvar and Camacho-Collados, 2019). This suggests that the Binder embeddings also capture contextualised semantic features to some extent. The improved performance of the approach in experiment 1b did not meaningfully contribute to improved performance in this downstream task. But, the Binder embeddings did outperform the smaller Van Dantzig feature-set embeddings from Experiment 2, suggesting that the larger Binder feature set is a more complete semantic representation of words.

Importantly, the nature of the Binder feature space makes interpreting the embeddings easier. Figure 4 below illustrates how the meaning of the word *building* differs in the two different context sentences from the WiC data-set.

However, Binder features predicted from transformer embeddings did not always match what would be expected. Figure 5 illustrates this, where the representation of *catch* in the second sentence appears closer to the *physical act of catching* rather than the intended meaning of *to catch fire*. Qualitative evaluation of the embeddings like this is powerful for understanding their quality, but comes at the cost of being time consuming.

7 Conclusion

The overarching aim of this work was to demonstrate that Binder style semantic feature embeddings can be derived from the BERT embedding space in the same way that previous research (Utsumi, 2018, 2020; Turton et al., 2020) has shown for static embeddings. It also offered the opportunity to probe how semantic information is represented across the different layers of BERT. Treating the embeddings as static, Experiment 1a supported this aim with BERT and other transformer embeddings outperforming the best performing static embeddings model Numberbatch. The results also supported the findings of Jawahar et al. (2019) that semantic information tends to be represented in the later layers of BERT. Hand-picking sentences in Experiment 1b lead to better performance indicating that some degree of context is represented in the derived semantic features.

Experiment 2 provided further evidence of the ability of transformer models to derive contextualised semantic features but was limited by the small set of features and the short word-pair context sequences.

Finally, the ability of Binder embeddings to perform comparatively to raw BERT embeddings in Experiment 3 suggests that they do capture, to some degree, contextualised semantic features when derived from transformer embeddings.

In conclusion, within the limitations of the Binder dataset, this paper suggests that it is possible to derive contextualised semantic features from contextualised word embeddings as a proof of concept. However, without a ground-truth test, it is not able to demonstrate this conclusively. To do this would likely require the production of a Binder feature set for words explicitly in context, and this may be a necessary next step if the Binder feature set is considered useful for further use. Furthermore, as the Binder dataset focuses on general use words, for researchers wishing to derive semantic features useful for specific domains, they likely would need to construct datasets of domain-specific features for a domain-specific vocabulary.

Beyond the direct findings of this paper, we also hope that this work highlights the usefulness of using existing psychological research data to improve the understanding and interpretability of what can otherwise be somewhat opaque deep learning models.

References

Cindy Aloui, Carlos Ramisch, Alexis Nasr, and Lucie Barque. 2020. Slice: Supersense-based lightweight interpretable contextual embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3357–3370.

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

Jeffrey R Binder, Lisa L Conant, Colin J Humphries, Leonardo Fernandino, Stephen B Simons, Mario Aguilar, and Rutvik H Desai. 2016. Toward a brain-based componential semantic representation. *Cognitive neuropsychology*, 33(3-4):130–174.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6:213–234.

Davide Castelvecchi. 2016. Can we open the black box of ai? *Nature News*, 538(7623):20.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.

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

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Shusen Liu, Peer-Timo Bremer, Jayaraman J Thiagarajan, Vivek Srikumar, Bei Wang, Yarden Livnat, and Valerio Pascucci. 2017. Visual exploration of semantic relationships in neural word embeddings. *IEEE transactions on visualization and computer graphics*, 24(1):553–562.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. 2019. Word2sense: sparse interpretable word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5692–5705.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32:8594–8603.

Jennifer Rodd, Gareth Gaskell, and William Marslen-Wilson. 2002. Making sense of semantic ambiguity: Semantic competition in lexical access. *Journal of Memory and Language*, 46(2):245–266.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 298–307.

Lütfi Kerem Şenel, İhsan Utlu, Furkan Şahinuç, Haldun M Ozaktas, and Aykut Koç. 2020. Imparting interpretability to word embeddings while preserving semantic structure. *Natural Language Engineering*, pages 1–26.

Lütfi Kerem Şenel, Ihsan Utlu, Veysel Yücesoy, Aykut Koc, and Tolga Cukur. 2018. Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1769–1779.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

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.

Jacob Turton, David Vinson, and Robert Smith. 2020. Extrapolating binder style word embeddings to new words. In *Proceedings of the second workshop on linguistic and neurocognitive resources*, pages 1–8.

Akira Utsumi. 2018. A neurobiologically motivated analysis of distributional semantic models. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 1147–1152.

Akira Utsumi. 2020. Exploring what is encoded in distributional word vectors: A neurobiologically motivated analysis. *Cognitive Science*, 44(6):e12844.

Saskia Van Dantzig, Rosemary A Cowell, René Zeelenberg, and Diane Pecher. 2011. A sharp image or a sharp knife: Norms for the modality-exclusivity of 774 concept-property items. *Behavior Research Methods*, 43(1):145–154.

Yuxuan Wang, Yutai Hou, Wanxiang Che, and Ting Liu. 2020. From static to dynamic word representations: a survey. *International Journal of Machine Learning and Cybernetics*, pages 1–20.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763.

Ka Yee Yeung and Walter L Ruzzo. 2001. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774.

Appendix

	MODEL							
	BERT		GPT-2		XLNet		RoBERTA	
	Base	*Large*	*Small*	*Medium*	*Base*	*Large*	*Base*	*Large*
Parameters	110M	340M	117M	345M	110M	340M	125M	355M
Layers	12	24	12	24	12	24	12	24
Attention Heads	12	16	12	16	12	16	12	16
Hidden state size	768	1024	768	1024	768	1024	768	1024

Table a. Selected properties of the different transformer models used (large models shaded).

	MODEL											
	$BERT_{BASE}$			$GPT\text{-}2_{SMALL}$			$XLNet_{BASE}$			$RoBERTa_{BASE}$		
R-sq.	*First*	*Last*	*Mean*	*First*	*Last*	*Mean*	*First*	*Last*	*Mean*	*First*	*Last*	*Mean*
Comb.	.668	**.678**	.677	.548	**.630**	.611	.655	.660	**.665**	.660	.670	**.673**
Best	.657	**.671**	.667	.520	**.615**	.591	.645	.652	**.657**	.647	.652	**.658**

Table b. Mean R-squared across all Binder features for different subword embedding approaches (first subword, last subword or mean across all subwords). Comb. = combined best layer per feature. Best = best single layer overall.

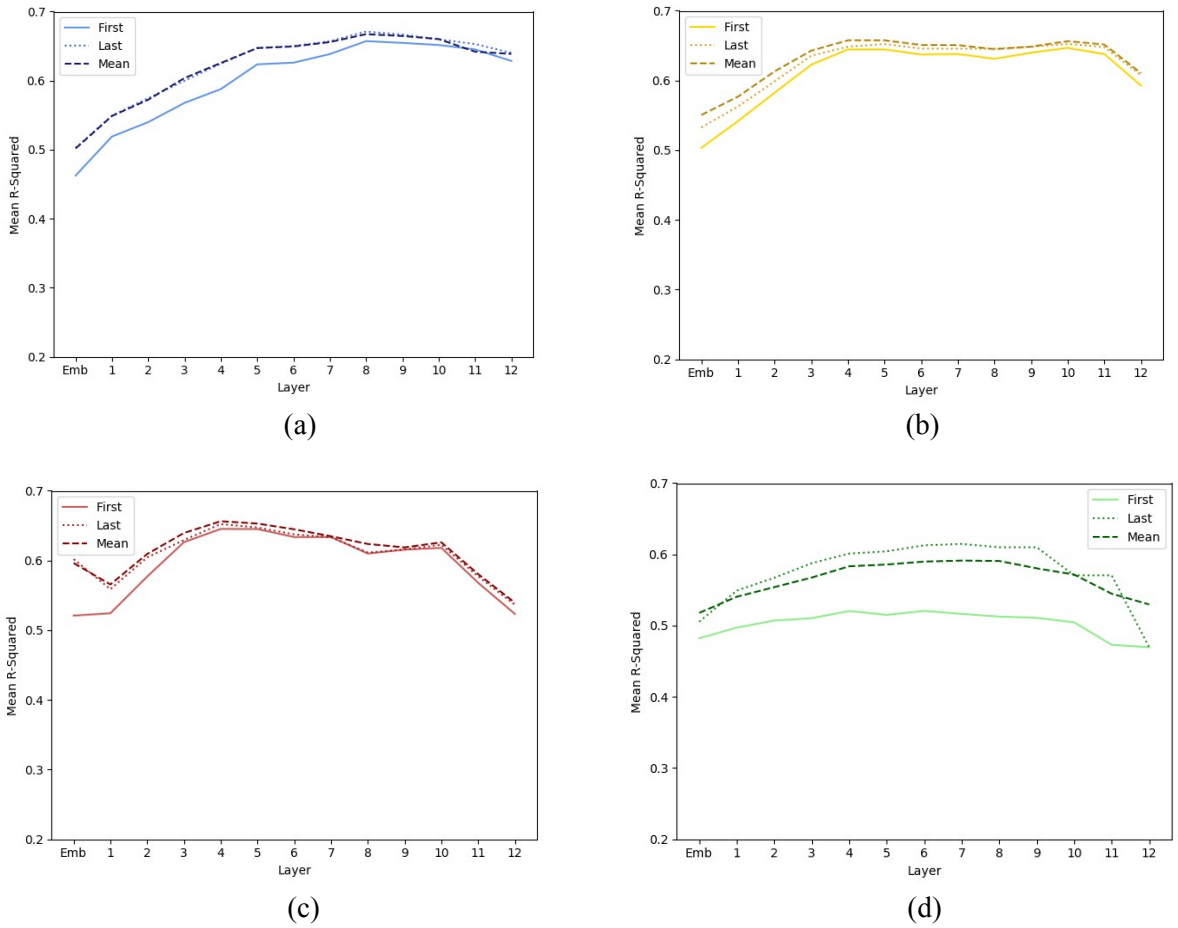


Figure a. performance of different subword embeddings across the 12 layers for (a) $BERT_{BASE}$ (b) $RoBERTA_{BASE}$ (c) $XLNet_{BASE}$ and (d) $GPT\text{-}2_{SMALL}$

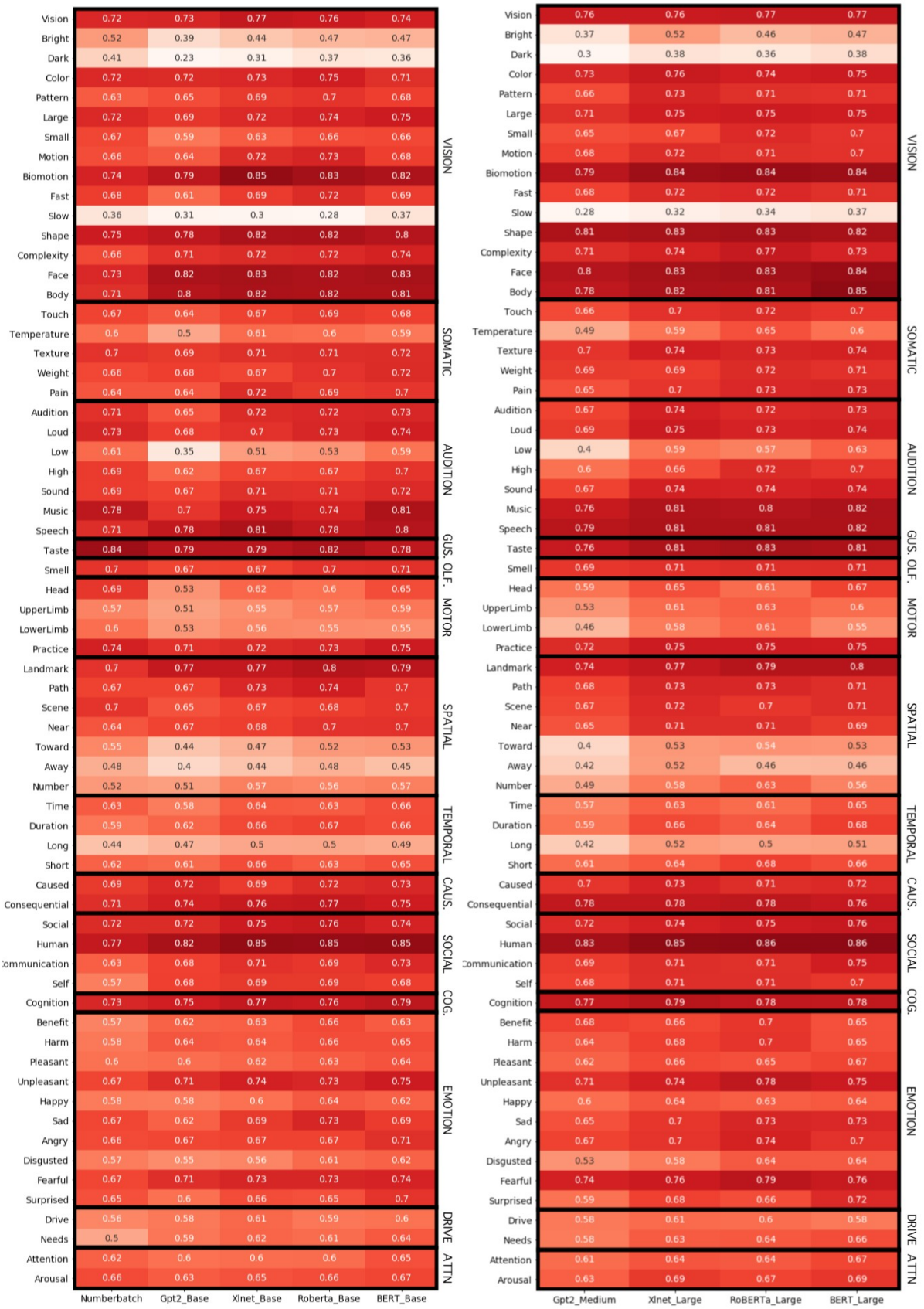


Figure b. All feature R-squared scores for the Numberbatch baseline and (a) small models (b) large models, with Binder et al (2016) categories indicated.

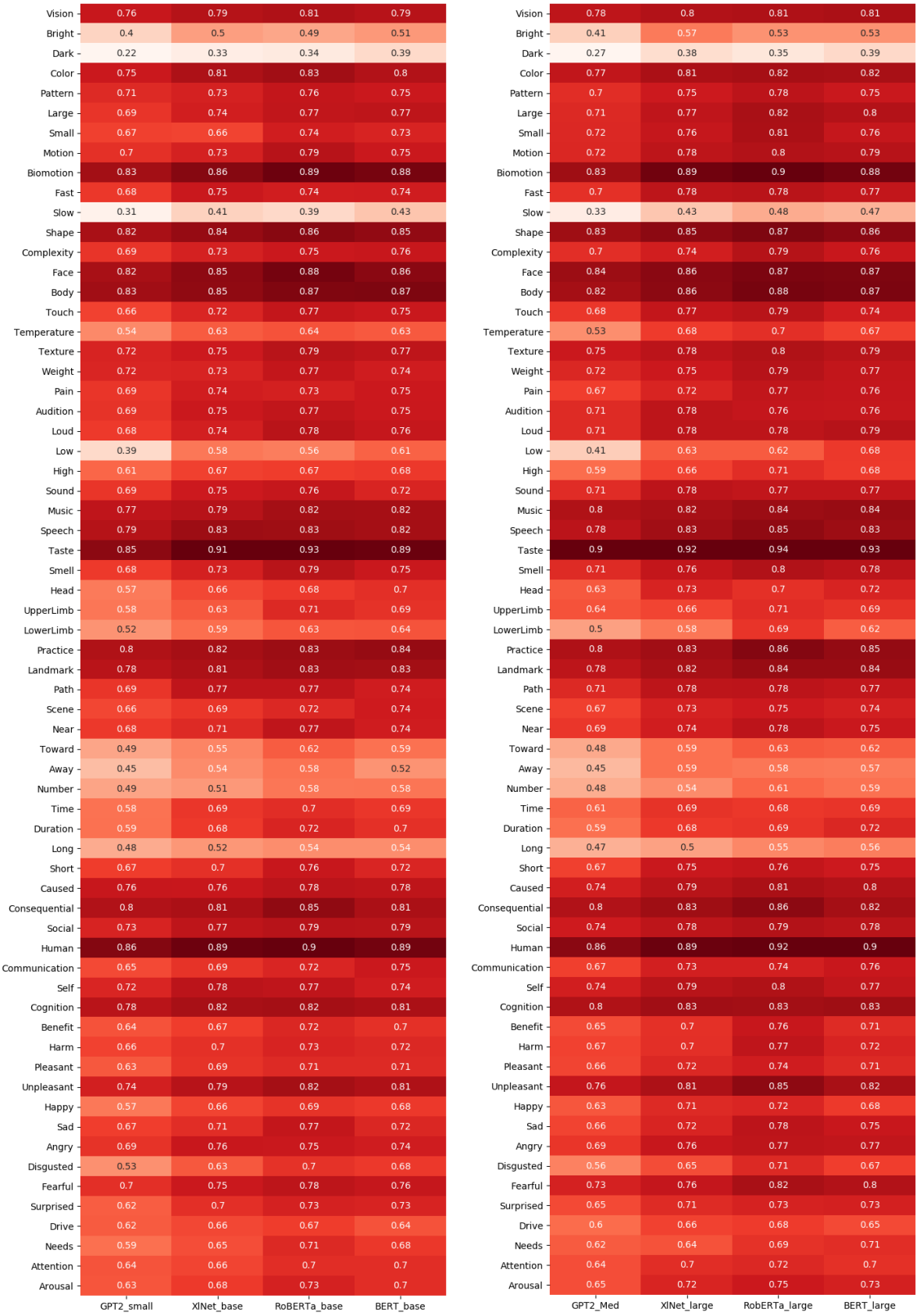
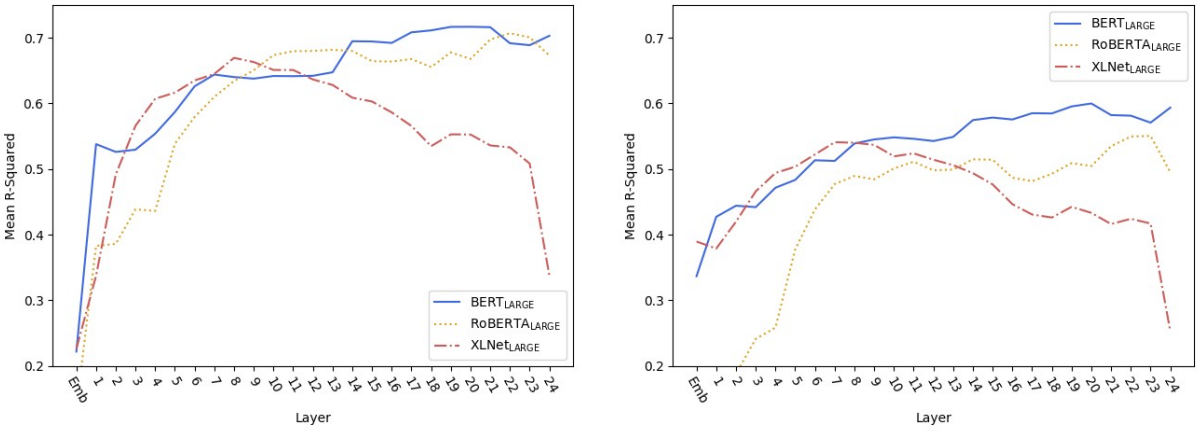


Figure c. All feature R-squared scores for the (a) small and (b) large models for selected sentences of Experiment 1b.



(a) (b)

Figure d. Model per-layer mean R-squared scores for Experiment 2 using (a) individual word-pair property embedding and (b) mean across word-pairs property embedding.

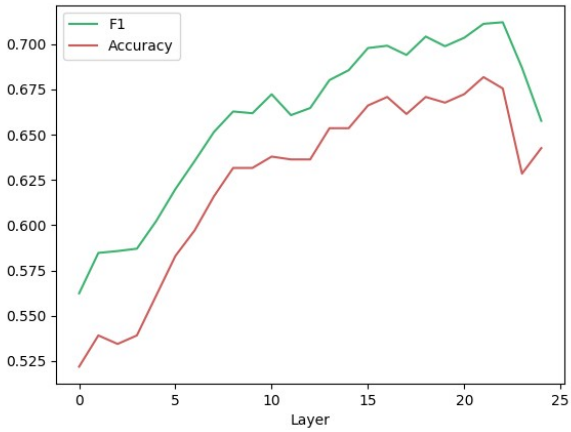
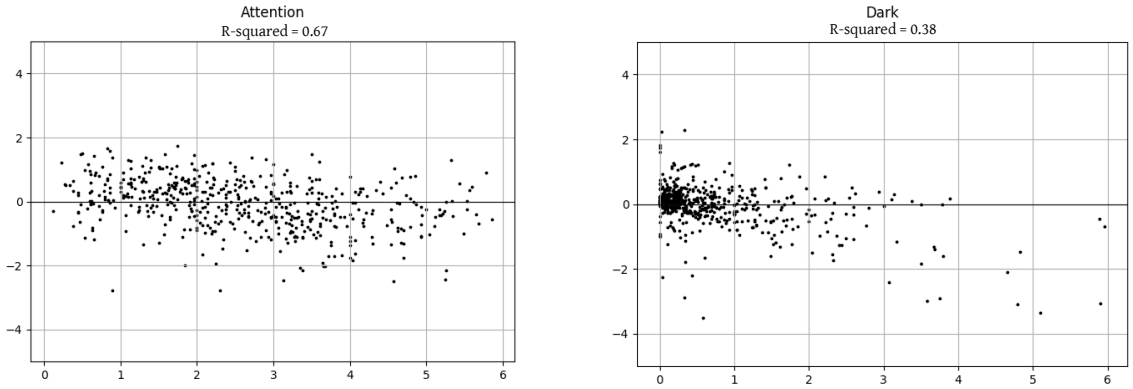


Figure e. Raw BERT$_{LARGE}$ Accuracy and F1 scores on WiC dataset



(a) (b)

Figure f. Residual plots for features (a) *Attention* and (b) *Dark*