THE UNIVERSITY OF CHICAGO

ANALYZING AND IMPROVING COMPOSITIONALITY IN NEURAL LANGUAGE MODELS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
LANG YU

CHICAGO, ILLINOIS
JUNE 2021

To my family.

"It is not enough to show how clever we are by showing how obscure everything is."

— J.L. Austin

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

Before starting my PhD, I could never image what a long and winding, yet rewarding, journey it would be. It has been my privilege to meet and work with those who have inspired and encouraged me down this road. Without their help, I could never become who I am today, let alone completing this dissertation. I only wish I list all of them here.

My advisor, Prof. Allyson Ettinger. I still remember the first time I talked with her at her office. I was going through the darkest time of my life, and I would like to thank her for saving me from the abyss of depression and self-denial. I appreciate the times when we stayed up together for deadlines, and the countless revisions she made on my scholarly papers. She has been my source of knowledge and insights, inspiring me to pursue my goals. Furthermore, Allyson is also a friend to me—she is a good listener and is extremely understanding of my occasional distress. Words fail me on how grateful I am for all her mentoring and guidance.

My committee members—Prof. Chen Yuxin and Prof. Rebecca Willett. I want to thank them for their valuable advice on my work. It has been a great pleasure working with them in the courses on machine learning. Also some other faculty members: Prof. John Goldsmith. Thanks for introducing me to computational linguistics, and being the advisor of my Master's project. Prof. Aaron Elmore. When I visited Chicago before accepting the offer, Aaron told me "PhD is a tough journey, but a rewarding one." How I wish I understood his words back then! Prof. Kevin Gimpel. I learned a lot from his Advanced NLP class, and I really appreciate the instructions he gave me on how to be a better PhD.

My friends at the University of Chicago: Li Ziying, Steven Basart, Yang Fan, Zheng Qinqing, Zhang Liwen, Tong Hao, Zhang Siyu, Feng Huiling, Yu Fangzhou. They have always kept me company, and I shall cherish the memory of those parties and boardgame nights with them. I also want to express my gratitude to Wu Qinxuan, Kanishka Misra and other friends at CompLing Lab—I have benefited much from our lab meetings; and to my friends at Facebook Seattle: Jeongmin Lee, Piers Wang, Harry Mavroforakis, Aaron Schlenker, Wang

Feng, Michael Jiang and Ahmed Magdy. My summer internships were unforgettable and rewarding thanks to them. Their supports and encouragements inspired me during and after my internships.

Finally, I must thank my parents for their unconditional love and understanding. It is their company and support that help me survive the ups and downs. And most of all, my wife and friend, Carol Fan, who has shared the joys and tears with me over the last six years, and to whom I also own not a small intellectual debt. I dedicate this journey to them.

ABSTRACT

Deep transformer models have pushed performance on NLP tasks to new limits, suggesting sophisticated treatment of complex linguistic inputs. However, we have limited understanding of how these models handle representation of input sequences, and whether this reflects sophisticated composition of meaning like that done by humans. In this dissertation, we take steps to analyze and improve compositionality in natural language models.

We present systematic analysis of phrasal representations in state-of-the-art pre-trained transformers. We use tests leveraging human judgments of phrase similarity and meaning shift, and compare results before and after control of word overlap, to tease apart lexical effects versus composition effects. We find that phrase representation in these models relies heavily on word content, with little evidence of nuanced composition. We also identify variations in phrase representation quality across models, layers, and representation types, and make corresponding recommendations for usage of representations from these models.

Motivated by the observations of pre-trained transformers, we explore directions of improving compositionality in neural language models. We first investigate the impact of fine-tuning on the capacity of contextualized embeddings to capture phrase meaning information beyond lexical content. Specifically, we fine-tune models on an adversarial paraphrase classification task with high lexical overlap, and on a sentiment classification task. After fine-tuning, we assess phrasal representations in controlled settings following prior work. We find that fine-tuning largely fails to benefit compositionality in these representations, though training on sentiment yields a small, localized benefit for certain models. In follow-up analyses, we identify confounding cues in the paraphrase dataset that may explain the lack of composition benefits from that task, and we discuss factors underlying the localized benefits from sentiment training. We then inspect a model with compositional architecture and show that the model shows weak compositionality despite incorporating explicit composition structure.

CHAPTER 1

INTRODUCTION

Neural language models have been pushing the state-of-the-art in a variety of natural language processing (NLP) tasks. The constant advances and seemingly super-human performance in well-defined tasks suggest that these models may be succeeding at composition of complex meanings, which is an essential component of language understanding. However, these neural language models are opaque—they are sensitive to disturbances in input, and significantly underperform humans on various adversarial datasets.

In this dissertation, I will explore how language models handle representations of linguistic units, and whether they master language understanding. Specifically, I will focus on *composition*—a model's capacity to combine meaning units into more complex units. The effort of assessing language models to be covered in this dissertation includes: 1) I will propose a set of model-independent tasks aiming at teasing apart lexical content encoding from nuances of composition. With careful control of lexical content, compositional information is isolated from lexical content; 2) I will apply the proposed tasks to state-of-the-art pre-trained transformers (Vaswani et al., 2017); 3) I will futher investigate the possibilities of improving compositionality in transformers through fine-tuning; 4) I will complement the work on transformers with another analysis on Recurrent neural network grammars (RNNG) (Dyer et al., 2016)—a model with compositional architecture.

1.1 Analyzing composition in language models

A notable difficulty of analyzing composition in representations is the mixed effect of lexical content encoding and nuances of composition. Furthermore, as discussed by Poliak et al. (2018); Gururangan et al. (2018); Ettinger et al. (2018), superficial clues and/or biases can to a large extent inflate models' performance on downstream tasks. For instance, in paraphrase

identification task, word overlap between sentence pairs can be an indicative clue that the model is able to pick up, which significantly trivializes the task. To tackle the aforementioned problems, we propose tests leveraging human judgments of phrase similarity and meaning shift, and compare results before and after control of word overlap, to tease apart lexical effects versus composition effects. Deep transformer models have pushed performance on NLP tasks to new limits, suggesting sophisticated treatment of complex linguistic inputs. However, we have limited understanding of how these models handle representation of phrases, and whether this reflects sophisticated composition of phrase meaning like that done by humans. In this dissertation, I will mainly focus on analyzing and improving compositionality in transformer-type models. In Chapter 4, I will present systematic analysis of phrasal representations in state-of-the-art pre-trained transformers. We find that phrase representation in these models relies heavily on word content, with little evidence of nuanced composition. We also identify variations in phrase representation quality across models, layers, and representation types, and make corresponding recommendations for usage of representations from these models. However, lacking compositional information can be a limitation imposed by the transformer architecture. In Chapter 6, I will explore the compositionality in a model with explicit compositional architecture. We find that with the removal of biases and superficial cues, the model still shows very little nuanced composition despite incorporating compositional structures based on parsing trees.

1.2 Improving compositionality in language models

Although *composition* is an indispensable component of language understanding, when testing for composition in pre-trained transformer representations, we find that these representations reflect word content of phrases, but do not show signs of more sophisticated humanlike composition beyond word content. Motivated by the findings that these representations reflect heavy influences of lexical content, in Chapter 5, I will present a follow-up work on

improving compositionality of pre-trained transformers. We investigate the impact of fine-tuning on the capacity of contextualized embeddings to capture phrase meaning information beyond lexical content. Specifically, we fine-tune models on an adversarial paraphrase classification task with high lexical overlap, and on a sentiment classification task. After fine-tuning, we analyze phrasal representations in controlled settings following prior work. We find that fine-tuning largely fails to benefit compositionality in these representations, though training on sentiment yields a small, localized benefit for certain models. In follow-up analyses, we identify confounding cues in the paraphrase dataset that may explain the lack of composition benefits from that task, and we discuss factors underlying the localized benefits from sentiment training.

1.3 Composition in models with explicit composition structure

With the observations of transformers' reliance on lexical content and missing compositional information, possibility remains that the lack of compositionality results from the limitation of the model architecture. In Chapter 6, I will investigate RNNG (Dyer et al., 2016)—a model with hierarchical composition structure guided by syntactic parsing trees. We experiment with the representation generated by vanilla recurrent composition, and the one produced by hierarchical structure. We apply two sets of tasks to assess composition in models' embeddings. The first set is sentence probing tasks proposed in Ettinger et al. (2018). It utilizes a specialized sentence generation system to generate large, annotated sentence sets. With the generation system, they are able to control superficial cues of word content and word order, so that bag-of-word model is not able to achieve above-chance performance. The other evaluation set is the similarity correlation and paraphrase classification proposed in Chapter 4. We show that with the explicit compositional structure, the model does not do composition more than transformers. The model demonstrates strong performance on lexical probing tasks, fails on tasks require systematic learning of syntactic information. In addition,

even though RNNG shows non-trivial alignment with human judgment under normal setting, performance degradation is still significant in controlled tests. However, the model is less sensitive to lexical content removal, suggesting less reliance on word overlap information.

1.4 This dissertation

1.4.1 Contributions

The contributions described in this dissertation are threefold.

In Chapter 3-4 I will propose a set of model-independent evaluation tasks to assess phrasal representation and composition. The tasks aims at capturing correspondence of phrase representation with human judgment. Specifically, the evaluation consists of similarity correlation, paraphrase classification and a qualitative analysis called "landmark experiment" (Kintsch, 2001). I will then report analyses on the state-of-the-art transformers. Across all models, there is non-trivial alignment with human judgment, but it seems to rely on lexical information. With careful control of lexical content and removal of superficial cues, we observe severe performance drop in both similarity correlations and paraphrase classifications. We conclude that pre-trained transformers lack sophisticated phrase composition beyond word content encoding.

In Chapter 5 I will investigate the interplay between fine-tuning transformer models and composition. Inspired by the finding that pre-trained models show heavy reliance on lexical content, we examine whether models will show better evidence of composition after fine-tuning on tasks that are good candidates for requiring composition: an adversarial paraphrase dataset forcing models to classify paraphrases with high lexical overlap, and a sentiment dataset with fine-grained phrase labels to promote composition. I report analyses based on evaluation tasks proposed in Chapter 4. Additionally, I will present fine-grained analyses on model changes and impact of different fine-tuning datasets, shedding light on

understanding the process of fine-tuning language models. We find that fine-tuning has limited benefit on improving compositionality, and I will present explanations on why it fails.

In Chapter 6 I will focus on analyzing a language model with compositional structure. We apply the sentence probing tasks proposed in Ettinger et al. (2018) and tasks proposed in Chapter 4. We find that even with the presence of hierarchical composition structure, the model still shows little nuanced composition information.

1.4.2 Overview

The remainder of this dissertation is organized as follows: in Chapter 2 I will introduce previous work that this dissertation builds on. In Chapter 3 I will discuss a variety of token representations I investigate for analyzing compositional information. In Chapter 4 I will elaborate on tasks we propose for evaluating phrasal representations and teasing apart compositional information from lexical encoding. This chapter will focus on investigating pre-trained transformer models. In Chapter 5 I will further report the possibilities of improving models' compositionality via fine-tuning. In Chapter 6 I will complement Chapter 3-5 with an analysis of a model incorporating explicit compositional architecture. In Chapter 7 I will conclude the dissertation and discuss potentials of future directions.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, I discuss background and previous work related to the main body of work discussed in this dissertation. Section 2.1 briefly discusses background of text representation learning and language modeling. Section 2.2 gives overview of previous work on probing neural models. Section 2.3 discusses previous approaches on investigating composition, which is a central topic of this dissertation.

2.1 Text representation learning and language modeling (LM)

The work in this dissertation focuses on analyzing text representations. In this section we review the progress of learning text embedding and language modeling.

The early work on text representations explores latent semantic analysis (LSA) and learning word vectors. Prominent work on LSA investigates using singular value decomposition (Bellegarda, 2000), Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Tam and Schultz, 2005; Mrva and Woodland, 2006) and HMM-LDA (Griffiths et al., 2004; Hsu and Glass, 2006). Notably attempts on mapping from individual words into word vectors include Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Values in these word vectors are generally trained based on semantic similarities (Rong, 2014). A notable characteristic that makes these embeddings attractive is the alignment between algebraic property and semantic meaning. A famous example is:

$$\boldsymbol{emb}(\text{king}) - \boldsymbol{emb}(\text{queen}) + \boldsymbol{emb}(\text{woman}) = \boldsymbol{emb}(\text{man})$$

where $\boldsymbol{emb(\cdot)}$ is the word vector of the corresponding word. When it comes to evaluating word embeddings, popular tasks include WordSim353 (Finkelstein et al., 2001) and SimLex-999 (Hill et al., 2015), which both evaluate based on semantic similarity.

A natural generalization of word embedding is sentence embedding. The idea of sentence embedding is to encode sentence text into a single vector, where important information is retrievable. Among different sentence embeddings, notable ones include: Doc2Vec (Le and Mikolov, 2014)—dense vectors are trained to predict words in the sentence/document; InferSent (Conneau et al., 2017)—the model is trained on Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) and Skip-thought (Kiros et al., 2015)—the encoder is trained to reconstruct the surrounding sentences.

Most early encoders are built on the assumption that words/sentences/paragraphs are characterized by their surroundings. And the resulting representations are context-independent. To tackle this problem, the idea of contextualized embeddings gains attention. One line of work approaches the problem by re-embedding existing word embeddings through neural models (McCann et al., 2017; Salant and Berant, 2018; Shi et al., 2019). ELMo (Peters et al., 2018a) vectors are derived from a bidirectional LSTM trained with language model objective, which captures context-dependent aspects of word meaning.

Since the introduction of the self-attention mechanism (Vaswani et al., 2017), transformer models become a new milestone. Universal Sentence Encoder (Cer et al., 2018) demonstrates promising performance with transformer and Deep Averaging Network (DAN) (Iyyer et al., 2015). GPT (Radford et al., 2018, 2019; Brown et al., 2020) and BERT (Devlin et al., 2019) pushed performance of a wide variety of downstream NLP tasks to new limit. And it becomes a paradigm to pre-train deep neural language models on general language modeling task, and fine-tune on downstream tasks. Following BERT, numerous variants are proposed. Notable work includes: RoBERTa (Liu et al., 2019) (builds on BERT, and makes changes to pre-training tasks and hyper-parameters), DistilBERT (Sanh et al., 2019) (a lightweight, fast transformer model with 40% than Bert base model and preserve competitive performance on The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018)), XLNet (Yang et al., 2019b) (builds on Transformer-XL (Dai et al., 2019) and uses

an autoregressive method to learn bidirectional contexts), XLM-RoBERTa (Conneau et al., 2019) (a large multilingual version of RoBERTa), BART (Lewis et al., 2020) (a denoising autoencoder based on transformer-based architecture) and SentiBERT (Yin et al., 2020) (a variant effectively captures sentiment semantics). In this dissertation, I will investigate some of the foregoing transformer models. In addition to language modeling tasks, BERT and its variants have been explored for other tasks. Li et al. (2020); Garg and Ramakrishnan (2020) explores adversarial samples generation with BERT. Zhang et al. (2019a) proposes using BERT as an automatic evaluation metric for text generation.

A common practice to evaluate the quality of word embeddings is to correlate embedding similarities with semantic/syntactic similarities (Finkelstein et al., 2001; Gerz et al., 2016; Hill et al., 2015; Conneau and Kiela, 2018). When it comes to evaluating LM models and sentence encoders, various tasks and datasets have been proposed. SentEval (Conneau and Kiela, 2018) presents a toolkit to evaluate sentence representations, which contains a variety of tasks including binary and multi-class classification, natural language inference and sentence similarity. XNLI (Conneau et al., 2018b) proposes a dataset of 15 languages, focusing on cross-lingual language understanding. Natural Language Decathlon (decaNLP) (McCann et al., 2018) encompasses a set of 10 tasks, formulating as question answering over context. GLUE (Wang et al., 2018) [1] and SuperGLUE (Wang et al., 2019) [2] are popular benchmarks for general-purpose language understanding. Despite showing improved performance on downstream tasks, analyses show that these contextualized embeddings have various limitations, for instance, gender bias (Zhao et al., 2019; Basta et al., 2019) and lack of semantic information (Tenney et al., 2019) and social bias (May et al., 2019; Kurita et al., 2019). The work presented in this thesis builds on the above works to evaluate representation quality and compositionality via correlation with human judgment. And we propose additional controlled

1. `https://gluebenchmark.com`

2. `https://super.gluebenchmark.com`

settings to isolate superficial cues that inflate models' performance. Extensive analysis work has specifically focused on transformer-type models, which will be covered in Section 2.2.

2.2 Interpretability of neural models

Along with continuous advances in NLP tasks, it attracts more and more attention to open up black-box models, and understand why models work/not work. One line of work interpret existing models by probing and analyzing outputs/representations/internals of these models. Another branch of effort attempts to build interpretable neural models from ground up, making output directly explainable. The work to be covered in later chapters closely relates to previous work on probing and interpreting neural models. In this section, I will review notable work that this dissertation builds on.

While variation and improvement of state-of-the-art models report surprisingly good performance, the community also notices failures of neural NLP models. Poliak et al. (2018); Gururangan et al. (2018); Chen et al. (2016) identify biases and uncontrolled cues exist in popular datasets that can inflate the performance. Geva et al. (2019) report annotator bias in crowd-sourcing dataset also harms the soundness of NLP tests. Ribeiro et al. (2019) open up a new perspective on QA tasks—regardless of correctness, models should produce consistent and coherent answers to questions on the same fact. They conclude that current models fall short on consistency, despite reporting strong performance on QA datasets. Gardner et al. (2020) reach a similar conclusion to this dissertation where systematic gaps (annotation artifacts etc.) can trivialize seemingly hard tasks.

All foregoing works imply that solely looking at evaluation metrics of neural models is not sufficient to determine the quality of neural models. To tackle the problem, various methods are proposed to evaluate neural models beyond metrics, including: correlating input features with model outputs (Belinkov and Bisk, 2017; Wallace et al., 2019b), approximating local decision boundaries with explainable linear models (Ribeiro et al., 2016) and investigating

important training samples (Yeh et al., 2018; Koh and Liang, 2017). Instead of evaluating models in a different way, another line of work (Andreas et al., 2015; Bogin et al., 2020; Lei et al., 2016; Narang et al., 2020) builds models with interpretability in the structure.

Besides above approaches, another popular trend is to directly interpret model predictions. Significant advantages of this method are that it is model-agnostic, lightweight to compute and faithful to underlying model. In terms of interpretation methods, a wide variety of directions have been explored, among which two approaches are frequently used:

1. Correlate importance of input features with model output. Murdoch et al. (2018) introduce contextual decomposition (CD) to interpret predictions made by LSTMs. Some works use gradient-based approaches to generate saliency maps to interpret image classification models (Simonyan et al., 2013; Shrikumar et al., 2017) and NLP models (Han et al., 2020). In addition to vanilla gradient based interpretation, many other variants are proposed—SmoothGrad (Smilkov et al., 2017) and Integrated Gradients (Sundararajan et al., 2017).

2. Systematically introduce perturbation to input as explanations. Anchors (Ribeiro et al., 2018a) and Universal Adversarial Triggers (Wallace et al., 2019a) are two notable attempts to find global decision rules by searching consistent perturbation patterns that affect models' output. Leave-one-out (Li et al., 2016)—define importance as drop in confidence when an input token is erased. Input Reduction (Feng et al., 2018) approaches the problem from a different direction where they iteratively remove least important input tokens without changing output of a model. Subramanian et al. (2019) further applies Input Reduction to models on Visual Question Answering (VQA) dataset. Additionally, several other approaches to generate adversarial examples have been proposed: HotFlip (Ebrahimi et al., 2018) targets at adversarial generation against character-level neural classifiers; SEARs (Ribeiro et al., 2018b) explores semantic-preserving perturbations that induce changes in the model's predictions; SCPN (Iyyer

et al., 2018) generates adversarial examples given a sentence and target syntactic form.

This dissertation contributes to a growing body of work on analysis of neural network models. Much work has studied recurrent neural network language models (Linzen et al., 2016; Wilcox et al., 2018; Chowdhury and Zamparelli, 2018; Gulordava et al., 2018; Futrell et al., 2019) and sentence encoders (Adi et al., 2016; Conneau et al., 2018a; Ettinger et al., 2016). Extensive work has studied the nature of learned representations in NLP models (Adi et al., 2016; Conneau et al., 2018a; Ettinger et al., 2016; Durrani et al., 2020). The present work builds in particular on analysis of contextualized representations (Bacon and Regier, 2019; Tenney et al., 2019; Peters et al., 2018b; Hewitt and Manning, 2019; Klafka and Ettinger, 2020; Toshniwal et al., 2020). Chang and Chen (2019) proposes a framework of explaining information captured by contextualized word embeddings. Wu et al. (2020) analyzes contextual word representation based on similarities.

In particular, a majority part of this dissertation concentrates on analyzing transformers. Characteristics of transformers are major considerations when designing probing tasks: instead of incremental composition in recurrent networks, transformers maintain representations for every token in every layer. It lacks a clear aggregated representations of a text span. Figure 2.1 demonstrates a common setup for interpreting transformer models. A popular practice is to perform layer-wise analysis—applying tasks to contextualized representations/attention heads/internal parameters to every layer of the model. The layer-wise analysis provides insights on dynamics of the model, shedding lights on how representations/model internals evolve as layer progresses. As for the choice of tasks, two trends prevail:

1. Classification-based probing. This line of work often designs classification tasks with controls to probe behaviors of a model. The most common setup uses contextualized embeddings as the input to a classifier, and the classification accuracy reflects information encoded in the contextualized representations. Kim et al. (2019a) target at function word comprehension through a set of classification tasks. van Aken et al. (2019)

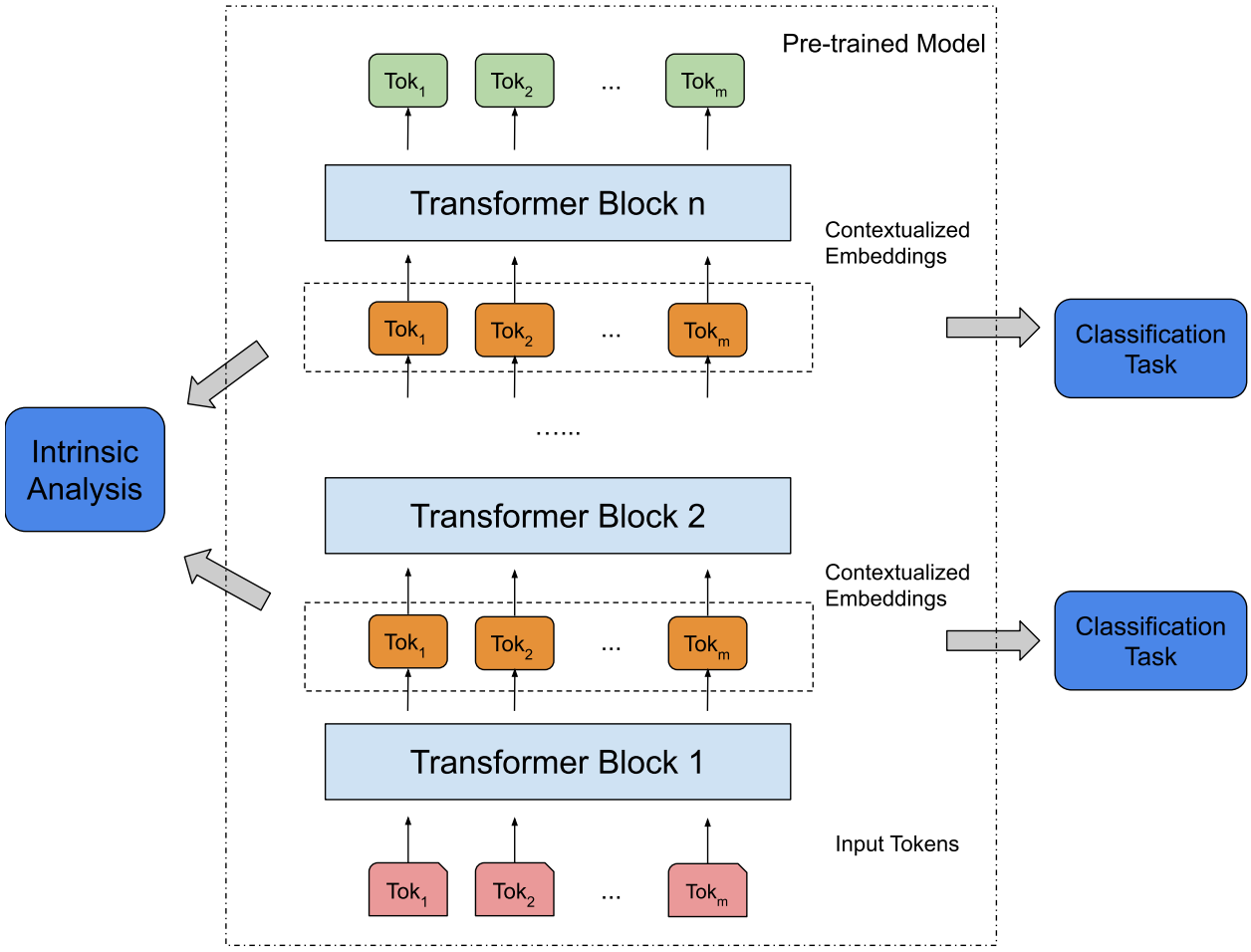

Figure 2.1: Common task setup for analyzing Transformers.

analyzes BERT's performance on QA tasks by applying probing tasks in a layer-wise fashion.

2. Intrinsic analysis. This branch analyzes models' internals directly—often with simply operations like cosine similarity, without additional classifiers. Specifically, Vig and Belinkov (2019); Clark et al. (2019) analyze attention mechanism, reporting correspondence between attention patterns and linguistic structures. Roberts et al. (2020); Raffel et al. (2020) report investigation on learned parameters of the models, exploring possibilities of transfer learning. There are also attempts exploring redundancy in transformers (Dalvi et al., 2020; Voita et al., 2019; Michel et al., 2019), suggesting

competitive performance can be reserved even majority of parameters are tuned.

Other than above popular trends, there have been other work on analyzing transformers. Ramnath et al. (2020) present layer-wise interpretation of BERT on Reading Comprehension based Question Answering (RCQA) through Integrated Gradients (Sundararajan et al., 2017). The evaluation that we use in this dissertation follows the paradigm of classification-based probing and correlation with similarity judgments.

Chapter 5 also builds on work subjecting trained NLP models to adversarial inputs, to highlight model weaknesses. One body of work approaches the problem by applying heuristic rules of perturbation to input sequences. PAWS (Zhang et al., 2019b)[3] and PAWSX (Yang et al., 2019a) are adversarial paraphrase datasets with high lexical overlap. The samples are generated via word swapping and back translation. State-of-the-art models report less than 40% accuracy without tuning on the dataset. In Chapter 5, I will present in-depth analysis of PAWS showing that superficial cues are not controlled, significantly trivializing the task. Universal Adversarial Triggers (Wallace et al., 2019a) explores gradient-based search to find input-agnostic sequences of tokens that trigger a model to produce a specific prediction. Jia and Liang (2017) propose an adversarial test scheme by inserting automatically generated sentences to samples in the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016). Another line of work uses neural models to construct adversarial examples or manipulate inputs in embedding space. Li et al. (2020) explores generating BERT adversarial samples using BERT. TEXTFOOLER (Jin et al., 2020) presents an model-independent algorithm to generate adversarial samples for text classification and textual entailment. Textbugger (Li et al., 2018) builds a framework to generate adversarial samples while preserving original meaning under both white-box and black-box settings.

Our work also contributes to efforts to understand impacts and outcomes of fine-tuning process. Miaschi et al. (2020); Mosbach et al. (2020); Merchant et al. (2020) compare linguistic

3. https://github.com/google-research-datasets/paws

knowledge learned by neural language models before and after fine-tuning, concluding that models lose general linguistic information during fine-tuning. Perez-Mayos et al. (2021) investigate the dynamics of syntax information during fine-tuning on various tasks (e.g. PoS tagging, dependency parsing and semantics-related tasks). To deal with the loss of generalizability during fine-tuning, Meta Fine-tuning (Wang et al., 2020) is proposed to train on multi-tasks in a group of similar tasks, mitigating learning gap between pre-training and fine-tuning to acquire transferable knowledge.

2.3 Composition in language models

Composition has drawn frequent attention in analysis of language models, as it is a fundamental component of language understanding. Several tasks have been proposed to evaluate composition in language models. Sentences Involving Compositional Knowledge (SICK)[4] (Marelli et al., 2014b) aims at composition of phrase and sentence meaning, which constitutes a task in SemEval-2014 (Marelli et al., 2014a). SICK contains ∼10,000 sentence pairs with semantic relatedness scores. Landmark test (Kintsch, 2001) also serves as a popular task on evaluating composition in early works. Question Answering via Sentence Composition (QASC) (Khot et al., 2020) is a QA dataset that explicitly requires model to compose facts from a large corpus in order to answer questions. It examines models' compositionality at sentence and paragraph level. COGS (Kim and Linzen, 2020) contains multiple systematic gaps that can only be solved by compositional generalization, such as generalizing syntactic structures.

In terms of methods for analyzing composition, one common practice relies on analysis of internal representations and downstream task behavior (Conneau et al., 2019; Nandakumar et al., 2019; McCoy et al., 2019). Some work adds careful controls on dataset to tease apart compositional information on sentence level (Ettinger et al., 2018; Dasgupta et al., 2018)

4. http://marcobaroni.org/composes/sick.html

and phrase level (Yu and Ettinger, 2020). Some work investigates compositionality via constructing non-linguistic synthetic datasets (Liška et al., 2018; Hupkes et al., 2018; Baan et al., 2019). Another line of work analyzes word interactions in neural networks' internal gates as the composition signal (Saphra and Lopez, 2020; Murdoch et al., 2018), extending the Contextual Decomposition algorithm proposed by Jumelet et al. (2019).

Similar to this dissertation, analyzing composition in transformers has also drawn frequent attentions. Staliūnaitė and Iacobacci (2020) explores the benefit of multitask learning in the context of a Conversational Question Answering (CoQA) task, which requires compositional semantics information. Parthasarathi et al. (2020) investigate the capability of transformers to compose multiple tasks within the same dialogue. Geva et al. (2020) focuses on understanding feed-forward layers in transformer language models. They conclude that it serves as a composition of key-value memories the model learned, and different layers pick up different textual patterns. Andor et al. (2019) explores the numerical composition of BERT on Discrete Reasoning Over Paragraphs (DROP) (Dua et al., 2019), and extend BERT with a lightweight extraction and composition layer.

In addition to analyzing composition in existing models, extensive effort has been made to incorporate composition in model architecture. Filimonov et al. (2020) propose a language modeling system that explicitly composes class-based models. Lin et al. (2019) introduce a dynamic composition mechanism to take reliability signals into consideration, and further improve performance on name tagging tasks. Recurrent Neural Network Grammars (RNNG) (Dyer et al., 2016) conducts composition of component representations hierarchically according to parsing tree of the input sequence. Parsing-Reading-Predict Networks (PRPN) (Shen et al., 2017) adopts a similar idea by simultaneously inducing syntactic structure and leveraging the inferred structure to learn a better language model.

The work presented in this thesis complement previous analysis work with a targeted and systematic study of a variety of models, focused on identifying both lexical and compositional

properties. We introduce controlled variants in addition to normal tasks to isolate signals of composition from the impact of other superficial cues that inflate models' performance.

Summary In this chapter, I review previous work that this dissertation closely relates to. Specifically, I discuss text representation learning and its evaluation in Section 2.1; progresses of NLP interpretability is covered in Section 2.2; lastly I present previous attempts on analyzing composition in Section 2.3.

CHAPTER 3

PHRASAL REPRESENTATION TYPES

A variety of approaches have been taken for representing sentences and phrases when all tokens output contextualized representations, as in our tested transformers. To clarify the phrasal information present in different forms of phrase representation, we experiment with a number of different combinations of token embeddings as representation types. In this dissertation, we experiment each of these representations at every layer of each model.

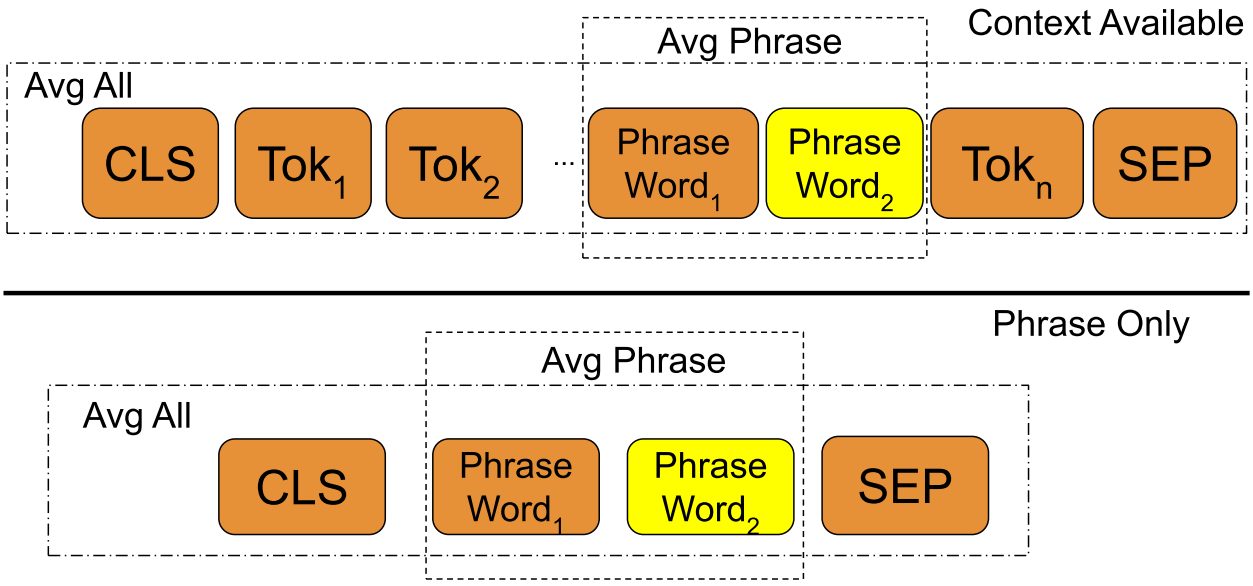

Figure 3.1: Example input sequences (BERT format). CLS is a special token at beginning of sequence. Tokens in yellow correspond to Head-Word. Avg-Phrase contains element-wise average of phrase word embeddings. Avg-All averages embeddings of all tokens.

Formally, let $[T_0, \cdots, T_k]$ be an input sequence of length $k+1$, with corresponding embeddings at ith layer $[\boldsymbol{e_0^i}, \cdots, \boldsymbol{e_k^i}]$. Assume the phrase spans the sequence $[a, b]$, where $0 \leq a \leq b \leq k$. Because two-word phrases are atypical inputs for these models, we experiment both with inputs of the two-word phrases alone ("phrase-only"), as well as inputs with the phrases embedded in sentences ("context-available"). This is illustrated in Figure 3.1 along with phrase representation types.

We test the following forms of phrase representation, drawn from each model and layer separately:

CLS Depending on specific models, this special token can be the first or last token of the input sequence (i.e. $\boldsymbol{e_0^i}$ or $\boldsymbol{e_k^i}$). In many applications, this token is used to represent the full input sequence.

Head-Word In each phrase, the head word is the semantic center the phrase. For instance, in the phrase "public service", "service" is the head word, expressing the central meaning of the phrase, while "public" is a modifier. Because phrase heads are not annotated in our datasets, we approximate the head by taking the embedding of the final word of the phrase. This representation is proposed as a potential representation of the whole phrase, if information is being composed into a central word:

$$\boldsymbol{p_{hw}^i} = \boldsymbol{e_b^i}$$

Avg-Phrase For this representation type we average the embeddings of the tokens in the target phrase (dashed box in Figure 3.1). This type of averaging of token embeddings is a common means of aggregate representation (Wieting et al., 2015).

$$\boldsymbol{p_{ap}^i} = \frac{1}{b-a+1}\sum_{x=a}^{b}\boldsymbol{e_x^i}$$

Avg-All Expanding beyond the tokens in "Avg-Phrase", this representation averages embeddings from the full input sequence.

$$\boldsymbol{p_{aa}^i} = \frac{1}{k+1}\sum_{x=0}^{k}\boldsymbol{e_x^i}$$

SEP With some variation between models, the SEP token is typically a separator for distinguishing input sentences, and is often the last token ($\boldsymbol{e^i_k}$) or second to last token ($\boldsymbol{e^i_{k-1}}$) of a sequence.

Other than aforementioned representation types, many representation types have been investigated by previous work (Toshniwal et al., 2020) including:

Endpoint is the concatenation of the boundary points of the phrase, which has shown effective in various tasks (Lee et al., 2016; Wadden et al., 2019) (e.g. SQuAD (Rajpurkar et al., 2016)):

$$\boldsymbol{p^i_{ep}} = [\boldsymbol{e^i_a}; \boldsymbol{e^i_b}]$$

Attention Pooling is a pooling method based on learned weights over the contextualized token embeddings (Tenney et al., 2019; Lee et al., 2017; Lin et al., 2017).

$$\boldsymbol{p^i_{att\_pool}} = \sum_{x=a}^{b} \alpha^i_x \boldsymbol{e^i_x}$$

$$t^i_x = \boldsymbol{v^i} \boldsymbol{e^i_x}$$

$$\alpha^i_x = softmax(t^i)_x$$

where $\boldsymbol{v^i}$ is a learned attention vector, and t^i_x is a weighted sum of the contextualized embeddings.

Max Pooling for all embeddings within the range of the phrase, max pooling takes the maximum value of the embeddings in each dimension to form the final embedding (Collobert et al., 2011; Conneau et al., 2017; Hashimoto et al., 2017).

Diff-Sum Diff-sum is a variant of Endpoint, which has been used by numerous previous work (Stern et al., 2017; Ouchi et al., 2018):

$$\boldsymbol{p}^{\boldsymbol{i}}_{\boldsymbol{ds}} = [\boldsymbol{e}^{\boldsymbol{i}}_{\boldsymbol{a}} + \boldsymbol{e}^{\boldsymbol{i}}_{\boldsymbol{b}}; \boldsymbol{e}^{\boldsymbol{i}}_{\boldsymbol{a}} - \boldsymbol{e}^{\boldsymbol{i}}_{\boldsymbol{b}}]$$

This dissertation does not include these representations for the reason that they do not have a clear correspondence between composed representation and its constituents. For above representations, each dimension of the final representation can be a combination of values from different individual constituents, which complicates the identification of composition process from individual words. We leave investigation on these representation types and other representations for larger linguistic units for future work.

CHAPTER 4

COMPOSITION IN PRE-TRAINED LANGUAGE MODELS

4.1 Introduction

A fundamental component of language understanding is the capacity to combine meaning units into larger units—a phenomenon known as *composition*—and to do so in a way that reflects the nuances of meaning as understood by humans. Transformers (Vaswani et al., 2017) have shown impressive performance in NLP, particularly transformers using pre-training, like BERT (Devlin et al., 2019) and GPT (Radford et al., 2018, 2019), suggesting that these models may be succeeding at composition of complex meanings. However, because transformers (like other contextual embedding models) typically maintain representations for every token, it is unclear how and at what points they might be combining word meanings into phrase meanings. This contrasts with models that incorporate explicit phrasal composition into their architecture, e.g. RNNG (Dyer et al., 2016; Kim et al., 2019b), recursive models for semantic composition (Socher et al., 2013), or transformers with attention-based composition modules (Yin et al., 2020).

In this chapter we take steps to clarify the nature of phrasal representation in transformers. We focus on representation of two-word phrases, and we prioritize identifying and teasing apart two important but distinct notions: how faithfully the models are representing information about the *words* that make up the phrase, and how faithfully the models are representing the nuances of the *composed phrase meaning* itself, over and above a simple account of the component words. To do this, we begin with existing methods for testing how well representations align with human judgments of meaning similarity: similarity correlations and paraphrase classification. We then introduce controlled variants of these datasets, removing cues of word overlap, in order to distinguish effects of word content from effects of more sophisticated composition. We complement these phrase similarity analyses with classic sense

selection tests of phrasal composition (Kintsch, 2001).

We apply these tests for systematic analysis of several state-of-the-art transformers: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019), XLNet (Yang et al., 2019b) and XLM-RoBERTa (Conneau et al., 2019). We run the tests in layerwise fashion, to establish the evolution of phrase information as layers progress, and we test various tokens and token combinations as phrase representations. We find that when word overlap is not controlled, models show strong correspondence with human judgments, with noteworthy patterns of variation across models, layers, and representation types. However, we find that correspondence drops substantially once word overlap is controlled, suggesting that although these transformers contain faithful representations of the lexical content of phrases, there is little evidence that these representations capture sophisticated details of meaning composition beyond word content. Based on the observed representation patterns, we make recommendations for selection of representations from these models. All code and controlled datesets are made available for replication and application to additional models. Datasets and code are available at `https://github.com/yulang/phrasal-composition-in-transformers`.

4.2 Testing phrase meaning similarity

Our methods begin with familiar approaches for assessing representations via meaning similarity: correlation with human phrase similarity judgments, and ability to identify paraphrases. The goal is to gauge the extent to which models arrive at representations reflecting the nuances of composed phrase meaning understood by humans. We draw on existing datasets, and begin by testing models on the original versions of these datasets—then we tease apart effects of word content from effects of more sophisticated meaning composition by introducing controlled variants of the datasets. The reasoning is that strong correlations with human similarity judgments, or strong paraphrase classification performance, could be

Normal Examples	
Source Phrase	Target Phrase & Score
average person	ordinary citizen (0.724)
	person average (0.518)
	country (0.255)
AB-BA Examples	
Source Phrase	Target Phrase & Score
law school	school law (0.382)
adult female	female adult (0.812)
arms control	control arms (0.473)

Table 4.1: Examples of correlation items. Numbers in parentheses are similarity scores between target phrase and source phrase. Upper half shows normal examples, and lower half shows controlled items.

influenced by artifacts that are not reflective of accurate phrase meaning composition per se. In particular, we may see strong performance simply on the basis of the amount of overlap in word content between phrases. To address this possibility, we create controlled datasets in which word overlap is no longer a cue to similarity.

As a starting point we focus on two-word phrases, as these are the smallest phrasal unit and the most conducive to these types of lexical controls, and because this allows us to leverage larger amounts of annotated phrase similarity data.

4.2.1 Phrase similarity correlation

We first evaluate phrase representations by assessing their alignment with human judgments of phrase meaning similarity. For testing this correspondence, we use the **BiRD** (Asaadi et al., 2019) dataset. BiRD is a bigram relatedness dataset designed to evaluate composition, consisting of 3,345 bigram pairs (examples in Table 4.1), with source phrases paired with numerous target phrases, and human-rated similarity scores ranging from 0 to 1.

In addition to testing on the full dataset, we design a controlled experiment to remove effects of word overlap, by filtering the dataset to pairs in which the two phrases consist of the same words. We refer to these pairs as "AB-BA" pairs (following terminology of the

authors of the BiRD dataset), and show examples in the lower half of Table 4.1.

We run similarity tests as follows: given a model M with layers L, for ith layer $l_i \in L$ and a source-target phrase pair, we compute representations of source phrase $p^i_{rep}(\text{src})$ and target phrase $p^i_{rep}(\text{trg})$, where rep is a representation type from Chapter 3, and we compute their cosine $\cos(p^i_{rep}(\text{src}), p^i_{rep}(\text{trg}))$. Pearson correlation $\boldsymbol{r_i}$ of layer l_i is then computed between cosine and human-rated score for all source-target pairs.

4.2.2 Paraphrase classification

Normal Examples	
Source Phrase	Target Phrase
are crucial	is absolutely vital (pos)
	was a matter of concern (neg)
	is an essential part (pos)
	are exacerbating (neg)
Controlled Examples	
Source Phrase	Target Phrase
communication infrastructure	telecommunications infrastructure (pos)
	data infrastructure (neg)

Table 4.2: Examples of classification items. Classification labels between target phrase and source phrase are in parentheses. Upper half shows normal examples, and lower half shows controlled items.

We further investigate the nature of phrase representations by testing their capacity to support binary paraphrase classification. This test allows us to explore whether we will see better alignment with human judgments of meaning similarity if we use more complicated operations than cosine similarity comparison. For the classification tasks, we draw on **PPDB 2.0** (Pavlick et al., 2015), a widely-used database consisting of paraphrases with scores generated by a regression model. To formulate our binary classification task, after filtering out low-quality paraphrases (discussed in Section 4.4), we use phrase pairs (source phrase, target phrase) from PPDB as positive pairs, and randomly sample phrases from the complete PPDB dataset to form negative pairs (source phrase, random phrase).

Because word overlap is also a likely cue for paraphrase classification, we filter to a controlled version of this dataset as well, as illustrated in Table 4.2. We formulate the controlled experiment here as holding word overlap between source phrase and target phrase to be exactly 50% for both positive and negative samples. Our choice of 50% word overlap in this case is necessary for construction of a sufficiently large, balanced classification dataset (AB-BA pairs in PPDB are too few to support classifier training, and AB-BA pairs are more likely to be non-paraphrases). Note, however, that by controlling word overlap to be exactly 50% for all phrase pairs, we still hold constant the *amount* of word overlap between phrases, which is the cue that we wish to remove. As an additional control, each source phrase is paired with an equal number of paraphrases and non-paraphrases, to avoid the classifier inferring labels based on phrase identity.

Formally, for each model layer l_i and representation type rep, we train

$$\text{CLF}^{i}_{rep} = \text{MLP}([\boldsymbol{pair^{i}_{rep}}])$$

where $\boldsymbol{pair^{i}_{rep}}$ represents embedding concatenations of each source phrase and target phrase:

$$\boldsymbol{pair^{i}_{rep}} = [\boldsymbol{p^{i}_{rep}(src)}; \boldsymbol{p^{i}_{rep}(trg)}]$$

The classifier is trained on binary classification of whether concatenated inputs represent paraphrases.

4.2.3 Feature importance analysis

With foregoing correlation and classification tasks, we investigate composition patterns of different token embeddings. However, when comparing different representations, which representation contains most information regarding paraphrase differentiation and how the pattern evolves across layers? To answer these questions, we formulate feature importance

analysis, targeting at analyzing relative amount of information contained in different tokens and their mappings. The main tool we use is LIME(Ribeiro et al., 2016)[1], which approximates complex black-box classifiers locally with interpretable linear models. For each layer of each model, we experiment with concatenating embeddings of all representation types as the input to train the classifier on the paraphrase classification task.

Formally, for each model layer l_i, we train

$$\text{CLF}^i = \text{MLP}([\boldsymbol{pair}^{\boldsymbol{i}}_{\boldsymbol{cls}}; \boldsymbol{pair}^{\boldsymbol{i}}_{\boldsymbol{sep}}; \boldsymbol{pair}^{\boldsymbol{i}}_{\boldsymbol{hw}}; \boldsymbol{pair}^{\boldsymbol{i}}_{\boldsymbol{aa}}; \boldsymbol{pair}^{\boldsymbol{i}}_{\boldsymbol{ap}}])$$

where $\boldsymbol{pair}^{\boldsymbol{i}}_{\boldsymbol{rep}}$ represents embedding concatenations of each source phrase and target phrase:

$$\boldsymbol{pair}^{\boldsymbol{i}}_{\boldsymbol{rep}} = [\boldsymbol{p}^{\boldsymbol{i}}_{\boldsymbol{rep}}(\boldsymbol{src}); \boldsymbol{p}^{\boldsymbol{i}}_{\boldsymbol{rep}}(\boldsymbol{trg})]$$

We train classifiers on paraphrase classification task for each layer of the model. LIME is then used to approximate the decision boundary of CLF_i, and normalize weights for each feature (in our case, token representations) in the vicinity of a specific sample. The weight of a specific representation reflects how much the presence of a representation contributes to the classification decision.

With LIME, we are able to probe into classifiers we trained, mitigating the potential discrepancy between strong classification performance and compositionality in models introduced in previous tasks. Another important insight we could have is to probe the classifier trained on controlled dataset. The feature importance reflects on which representation the classifier relies on most when word overlap information is removed. And thus the weights indicate which token representation, in which layer, contain most composed information.

1. https://github.com/marcotcr/lime

4.3 Polysemous disambiguation

In addition to aforementioned large-scale quantitative metrics, we also present fine-grained analysis proposed by (Kintsch, 2001; Mitchell and Lapata, 2008, 2010). The task focuses on models' ability to distinguish polysemous words by means of phrasal composition. We borrow two types of experiments presented in Kintsch (2001).

4.3.1 Landmark experiment

	horse ran	*color ran*
gallop	POS	NEG
dissolve	NEG	POS

Table 4.3: An example of landmark experiment of verb "run". Representations are expected to have higher cosine similarities between phrase and landmark word that are marked "POS".

Each test item consists of a) a central verb, b) two subject-verb phrases that pick out different senses of the verb, and c) two *landmark* words, each associating with one of the target senses of the verb. Table 4.3 shows an example with central verb "ran" and phrases "horse ran"/ "color ran". The corresponding landmark words are "gallop", which associates with "horse ran", and "dissolve", which associates with "color ran". The reasoning is that composition should select the correct verb meaning, shifting representations of the central verbs—and of the phrase as a whole—toward landmarks with closer meaning. For this example, models should produce phrase embeddings such that "horse ran" is closer to "gallop" and "color ran" is closer to "dissolve". We use the items introduced in (Kintsch, 2001), which consist of a total of 4 sets of landmark tests. We feed landmarks and phrases respectively through each transformer, without context, to generate corresponding representations p^i_{rep} for each layer l_i and representation type rep. Cosine similarity between each phrase-landmark pair is computed and compared against expected similarities.

4.3.2 Inference experiment

Source Sentence	Candidate 1	Candidate 2
the student washed the table.	the table was clean. (pos)	the student was clean. (neg)

Table 4.4: An example of inference experiment.

Each set consists of 3 sentences, where one source sentence is paired with two potential inferences. Reasonable sentence representation is expected to have high cosine similarity between source sentence and the positive candidate. In the example in Table 4.4, we evaluate contextualized representations for source sentence, candidate 1 and candidate 2 respectively, and calculate pair-wise cosine similarities.

In landmark experiment, all token representations discussed in Chapter 3 are examined, whereas only CLS is evaluated in inference experiment.

4.4 Experimental setup

Embeddings of each token are obtained by feeding input sequences through pre-trained contextual encoders. We investigate the "base" version of five transformers: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019), XLNet (Yang et al., 2019b) and XLM-RoBERTa (Conneau et al., 2019). For the models analyzed in this dissertation, we are using the implementation of (Wolf et al., 2019),[2] which is based on PyTorch (Paszke et al., 2019).

For correlation analysis, we first use the complete BiRD dataset, consisting of 3,345 phrase pairs.[3] We then test with our controlled subset of the data, consisting of 410 AB-BA pairs. For classification tasks, we first do preprocessing on PPDB 2.0,[4] filtering out pairs containing

2. https://github.com/huggingface/transformers

3. http://saifmohammad.com/WebPages/BiRD.html

4. http://paraphrase.org

hyperlinks, non-alphabetical symbols, and trivial paraphrases based on abbreviation or tense change. For our initial classification test, we use 13,050 source-target phrase pairs (of varying word overlap) from this preprocessed dataset. We then test with our controlled dataset, consisting of 11,770 source-target phrase pairs (each with precisely 50% word overlap). For each paraphrase classification task, 25% of selected data is reserved for testing. We use a multi-layer perceptron classifier with a single hidden layer of size 256 with ReLU activation, and a softmax layer to generate binary labels. We use a relatively simple classifier following the reasoning of Adi et al. (2016), that this allows examination of how easily extractable information is in these representations.

For both correlation and classification tasks, we experiment with phrase-only inputs and context-available (full-sentence) inputs. To obtain sentence contexts, we search for instances of source phrases in a Wikipedia dump, and extract sentences containing them. For a given phrase pair, target phrases are embedded in the same sentence context as the source phrase, to avoid effects of varying sentence position between phrases of a given pair. Because context sentences are extracted based on source phrases, our use of the same context for source and target phrases can give rise to unnatural contextual fit for target phrases. We consider this acceptable for the sake of controlling sentence position—and if anything, differences in contextual fit may aid models in distinguishing more and less similar phrases. The slight boost observed on the full datasets (for Avg-Phrase) suggests that the sentence contexts do provide the intended benefit from using input of a more natural size.

4.5 Results

In this section, we present results of transformers on similarity correlation and paraphrase classification, under both full and controlled settings. By comparing the performance on full and controlled datasets, we are able to isolate the effect of compositional information encoding from lexical content encoding. We will then complement these two tests with a

feature importance analysis (will be discussed in Section 4.6) and a qualitative analysis using tasks proposed in Kintsch (2001) (will be discussed in Section 4.7).

4.5.1 Similarity correlation

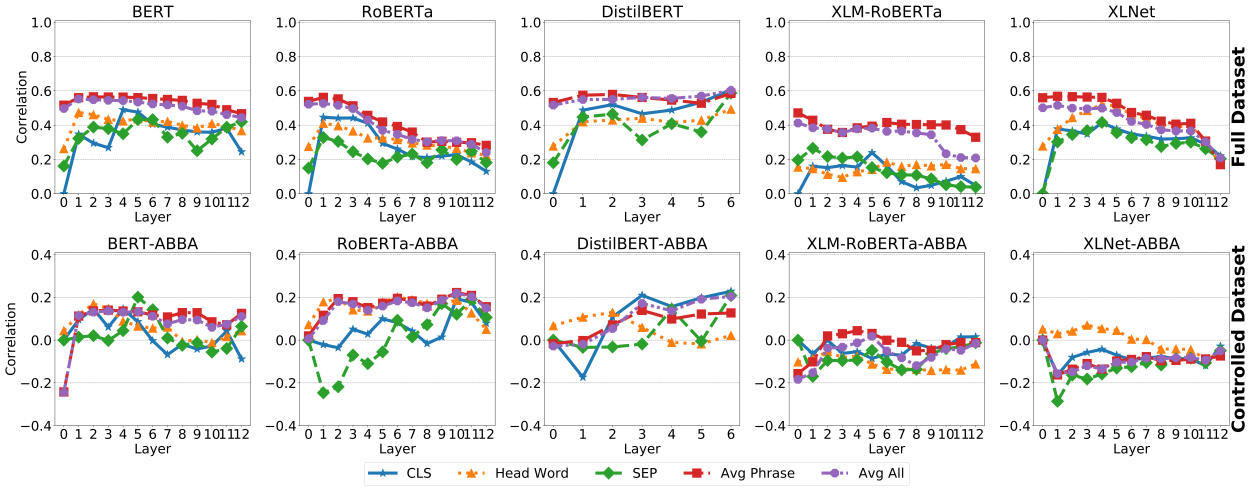

Figure 4.1: Correlation on BiRD dataset, phrase-only input setting. First row shows results on full dataset, and second row on controlled AB-BA pairs. Layer 0 corresponds to input embeddings passing to the model.

Full dataset The top row of Figure 4.1 shows correlation results on the full BiRD dataset for all models, layers, and representation types, with phrase-only inputs. Among representation types, Avg-Phrase and Avg-All consistently achieve the highest correlations across models and layers. In all models but DistilBERT, correlation of Avg-Phrase and Avg-All peaks at layer 1 and decreases in subsequent layers with minor fluctuations. Head-Word and SEP both show weaker, but non-trivial, correlations. The CLS token is of note with a consistent rapid rise as layers progress, suggesting that it quickly takes on properties of the words of the phrase. For all models but DistilBERT, CLS token correlations peak in middle layers and then decline.

Model-wise, XLM-RoBERTa shows the weakest overall correlations, potentially due to the fact that it is trained to infer input language and to handle multiple languages. BERT

retains fairly consistent correlations across layers, while RoBERTa and XLNet show rapid declines as layers progress, suggesting that these models increasingly incorporate information that deviates from human intuitions about phrase similarity. DistilBERT, despite being of smaller size, demonstrates competitive correlation. The CLS token in DistilBERT is notable for its continuing rise in correlation strength across layers. This suggests that DistilBERT in particular makes use of the CLS token to encode phrase information, and unlike other models, its representations retain the relevant properties to the final layer.

Controlled dataset Turning to our controlled AB-BA dataset, we examine the extent to which the above correlations indicate sophisticated phrasal composition versus effective encoding of information about phrases' component words. The bottom row of Figure 4.1 shows the correlations on this controlled subset. We see that performance of all models drops significantly, often with roughly zero correlation. Avg-All and Avg-Phrase no longer dominate the correlations, suggesting that these representations capture word information, but not higher-level compositional information. XLM-RoBERTa and XLNet show particularly low correlations, suggesting heavier reliance on word content. Notably, the CLS tokens in RoBERTa and DistilBERT stand out with comparatively strong correlations in later layers. This suggests that the rise that we see in CLS correlations for DistilBERT in particular may correspond to some real compositional signal in this token, and for this model the CLS token may in fact correspond to something more like a representation of the meaning of the full input sequence. The Avg-Phrase representation for RoBERTa also makes a comparatively strong showing.

Difference between full and controlled datasets Figure 4.2 shows the layer-wise drop in correlation of each model between full and controlled dataset. The performance gap reflects the extent of reliance on lexical content. Model-wise, RoBERTa in later layers demonstrates less significant drop. However, it is worth noting that correlation in these layers is relatively

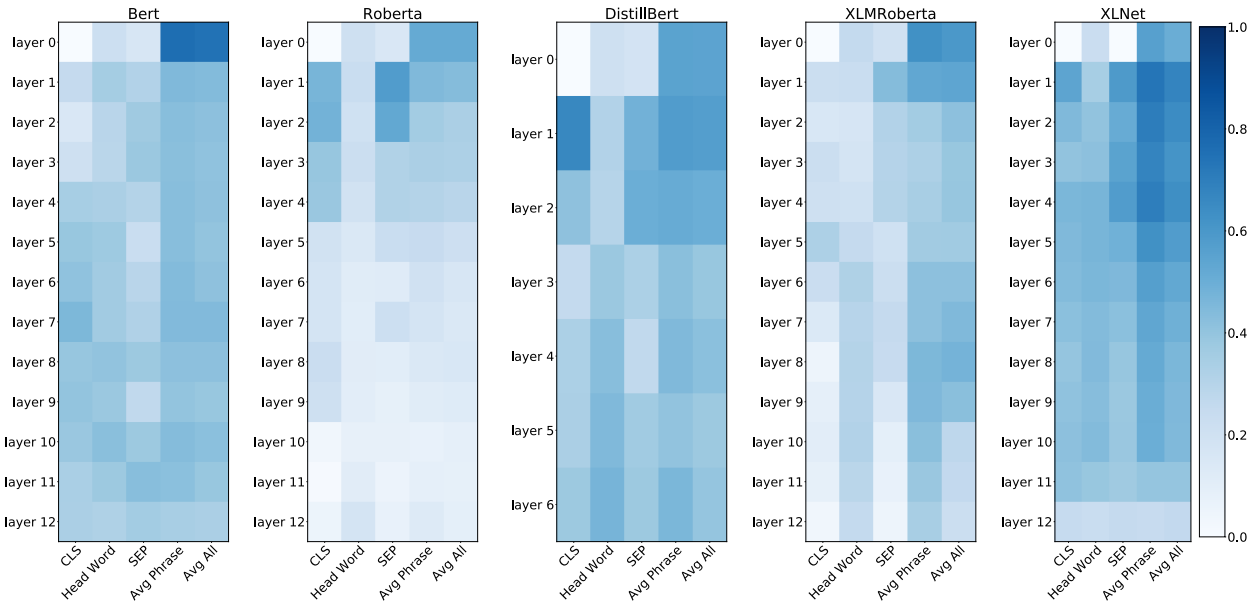

Figure 4.2: Correlation difference on BiRD between full and controlled dataset, phrase-only input setting. Layer 0 corresponds to input embeddings passing to the model.

weak on full dataset compared to other models, suggesting it contains information rather than maintaining lexical information from lower layers. Among representation types, the drop in CLS and SEP is less prominent compared to representations directly correspond to phrase tokens (i.e. Head Word, Avg-Phrase, Avg-All). It is consistent with the belief that CLS in particular captures higher level information of the entire input sequence. Layer-wise, lower layers show heavier influence of lexical content, and higher layers are more robust against the removal of word overlap cues.

Including sentence context Figure 4.3 shows the correlations when target phrases are embedded as part of a sentence context, rather than in isolation. As can be expected, Avg-Phrase is now consistently the highest in correlation on the full dataset—other tokens are presumably more impacted by the presence of additional words in the context. We also see that the Avg-Phrase correlations no longer drop so dramatically in later layers, suggesting that when given full sentence inputs, models retain more word properties in later layers than when given only phrases. This general trend holds also for Avg-All and Head-Word

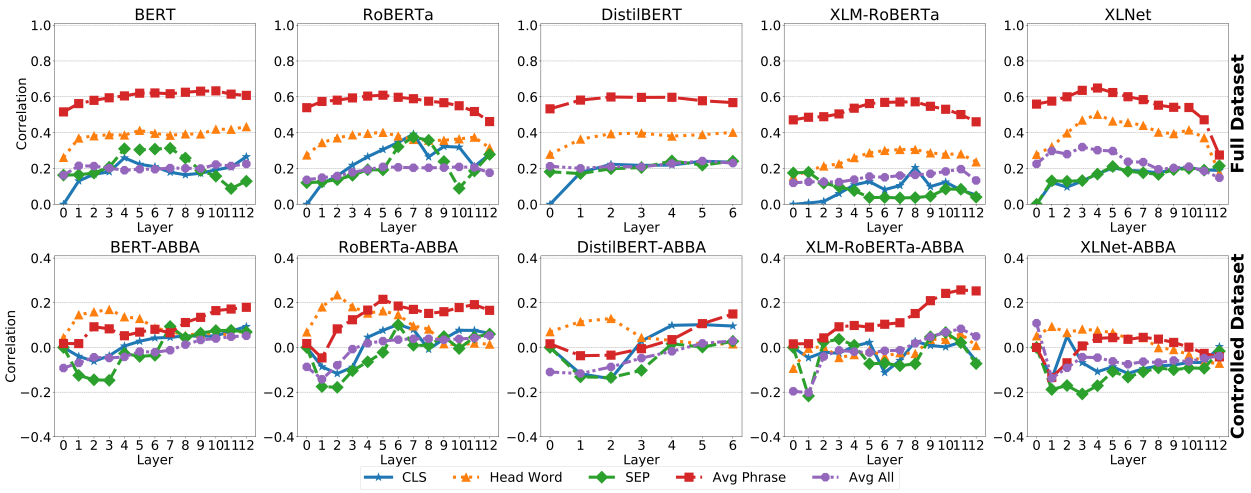

Figure 4.3: Correlation on BiRD dataset with phrases embedded in sentence context (context-available input setting).

representations.

In the AB-BA setting, we see that presence of context does boost overall correlation with human judgment. Of note is XLM-RoBERTa's Avg-Phrase, which without sentence context has zero correlation in the AB-BA setting, but which with sentence context reaches our highest observed AB-BA correlations in its final layers. However, even with context, the strongest correlation across models is still less than 0.3. It is still the case, then, that correlation on the controlled data degrades significantly relative to the full dataset. This indicates that even when phrases are input within sentence contexts, phrase representations in transformers reflect heavy reliance on word content, largely missing additional nuances of compositional phrase meaning.

Difference between full and controlled datasets Figure 4.4 shows the correlation difference under context available setting. Compared to the result under phrase-only setting in Figure 4.2, the overall drop is mitigated with the presence of context. Among all representation types, CLS and SEP are of note having less degradation compared to phrase-only input. The only exception is Avg-phrase, where context words push it to focusing more on lexical information, and thus we see a more notable correlation drop. Model-wise, BERT, RoBERTa

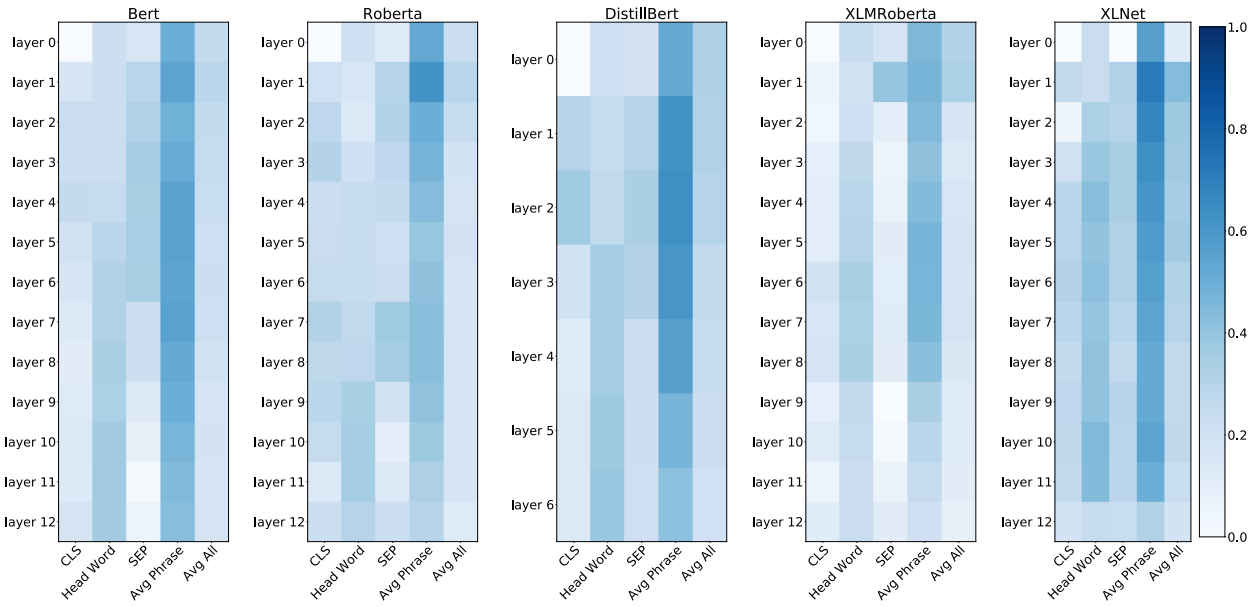

Figure 4.4: Correlation difference on BiRD between full and controlled dataset, context-available input setting.

and XLM-RoBERTa benefit more from contexts, whereas XLNet still shows heavy reliance on lexical content, especially for Head-word representation.

4.5.2 Paraphrase classification

Full dataset Results for our full paraphrase classification dataset, with phrase-only inputs, are shown in the top row of Figure 4.5. Accuracies are overall very high, and we see generally similar patterns to the correlation tasks. Best accuracy is achieved by using Avg-Phrase and Avg-All representations. RoBERTa, XLM-RoBERTa, and XLNet show decreasing correlations for top-performing representations in later layers, while BERT and DistilBERT remain more consistent across layers. Performance of CLS requires a few layers to peak, with top performance around middle layers, and in some models shows poor performance in later layers. SEP shows unstable performance compared to other representations, especially in DistilBERT and RoBERTa.

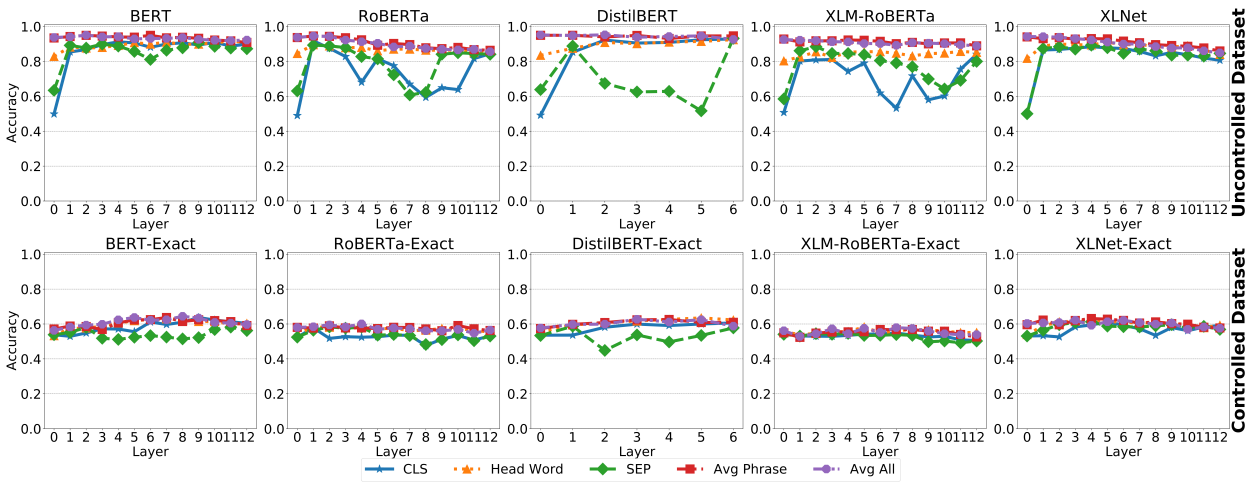

Figure 4.5: Classification accuracy on PPDB dataset (phrase-only input setting). First row shows classification accuracy on original dataset, and second row shows accuracy on controlled dataset.

Controlled dataset The bottom row of Figure 4.5 shows classification accuracy when word overlap is held constant. Consistent with the drop in correlations on the controlled AB-BA experiments above, classification performance of all models drops down to only slightly above chance performance of 50%. This suggests that the high classification performance on the full dataset relies largely on word overlap information, and that there is little higher-level phrase meaning information to aid classification in the absence of the overlap cue. We see in some cases a very slight trend such that classification accuracy increases a bit toward middle layers—so to the extent that there is any compositional phrase information being captured, it may increase within representations in the middle layers. Overall, the consistency of these results with those of the correlation analysis suggests that the apparent lack of accurate compositional meaning information in our tested phrase representations is not simply a result of cosine correlations being inappropriate for picking up on correspondences.

Difference between full and controlled datasets Figure 4.6 shows the classification accuracy changes of all models on full and controlled PPDB dataset. The accuracy changes are more consistent as layer progresses compared to correlation changes. One potential reason

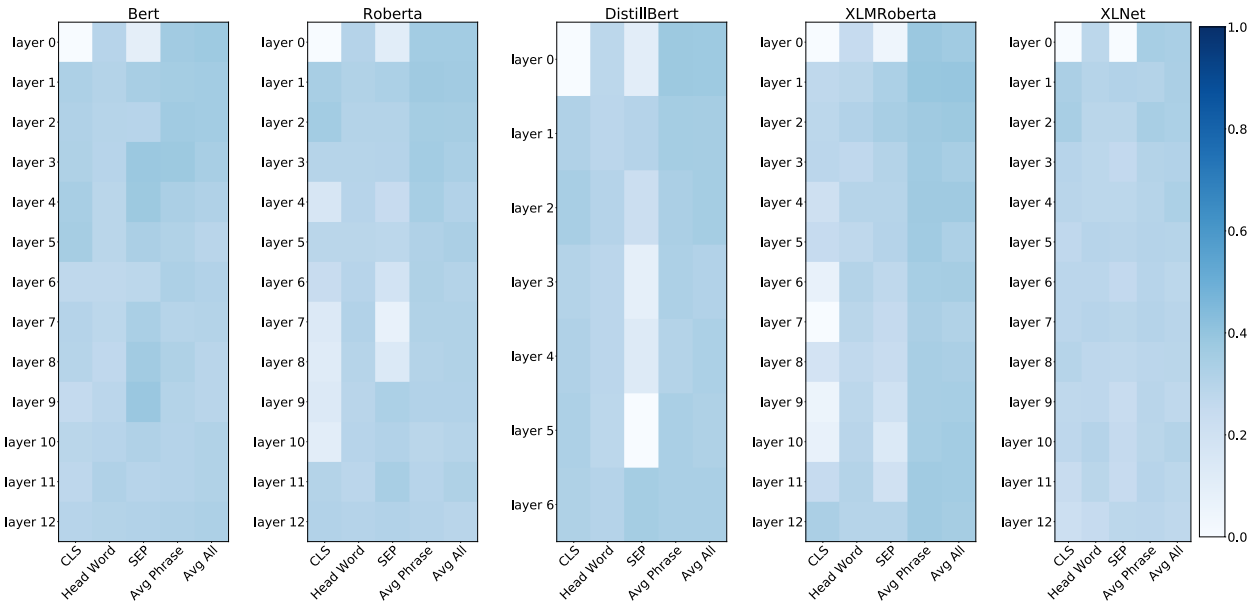

Figure 4.6: Accuracy difference on PPDB between full and controlled dataset, phrase-only input setting. Layer 0 corresponds to input embeddings passing to the model.

is that MLP classifier masks minor fluctuations in information encoded in contextualized embeddings, whereas cosine similarity is more sensitive to these fluctuations. However, the overall trends we observe are consistent with the similarity correlation task: representation-wise, CLS and SEP show relatively less influence of lexical information, whereas Avg-Phrase and Avg-All show more accuracy drop in early layers. The finding supports our conclusion from similarity correlation, where representations in lower layers of transformers maintain more low level lexical information and CLS maintain more higher-level information. Model-wise, DistilBERT and RoBERTa demonstrate more robustness against the removal of word overlap. And layer-wise, later layers show less significant accuracy drop, indicating some higher level compositional information is captured.

Including sentence context Figure 4.7 shows the classification results for representations of phrases embedded in sentence contexts. The patterns largely align with our observations from the correlation task. Performance on the full dataset is still high, with Avg-Phrase now showing consistently highest performance, being least influenced by the presence of new context

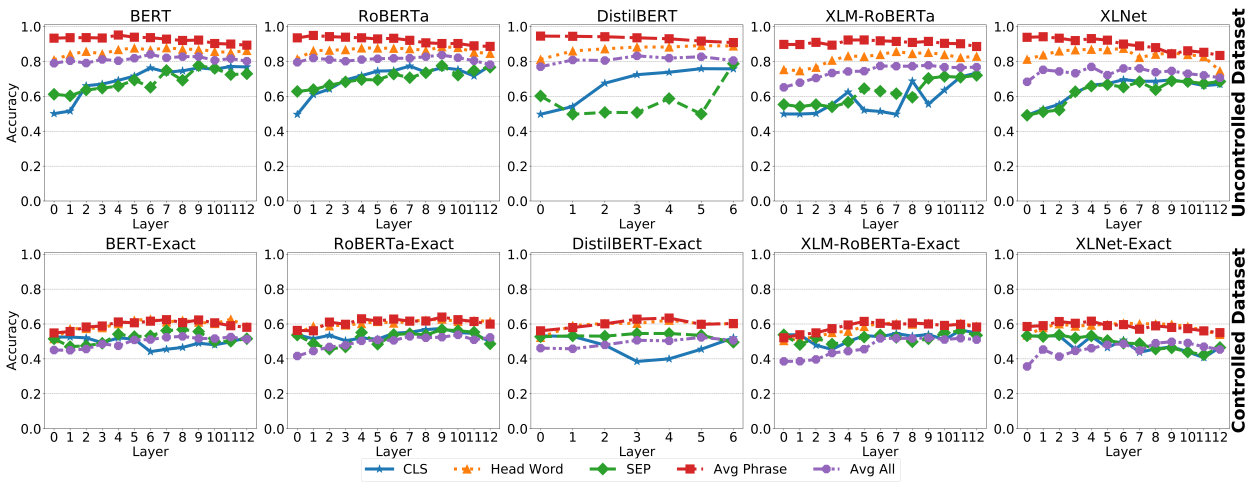

Figure 4.7: Classification accuracy on PPDB dataset with phrases embedded in sentence context. First row shows classification accuracy on original dataset, and second row shows accuracy on controlled dataset.

words. In the controlled setting, we see the same substantial drop in performance relative to the full dataset—there is very slight improvement over the phrase-only representations, but the highest accuracy among all models is still around 0.6. Thus, the inclusion of sentence context again does not provide any additional evidence for sophisticated compositional meaning information in the tested phrase representations.

Difference between full and controlled datasets Figure 4.8 presents the accuracy difference on PPDB dataset under full and controlled settings. Similar to context-available correlation task, overall accuracy drops are mitigated with context words available, which benefits from the slight improvement on controlled task. Notably, CLS and SEP show minor drops, especially in BERT, DistilBERT and XLM-RoBERTa. It is worth noting that later layers SEP in DistilBERT and XLM-RoBERTa show a different trend where last few layers have more significant drop than early layers. It implies that the presence of context makes the contextualized embeddings in later layers to maintain more lexical information. We find similar trend in XLNet, where early layers have less accuracy changes but later layers suffer more. The trend is also a result of more significant improvement in later layers under full

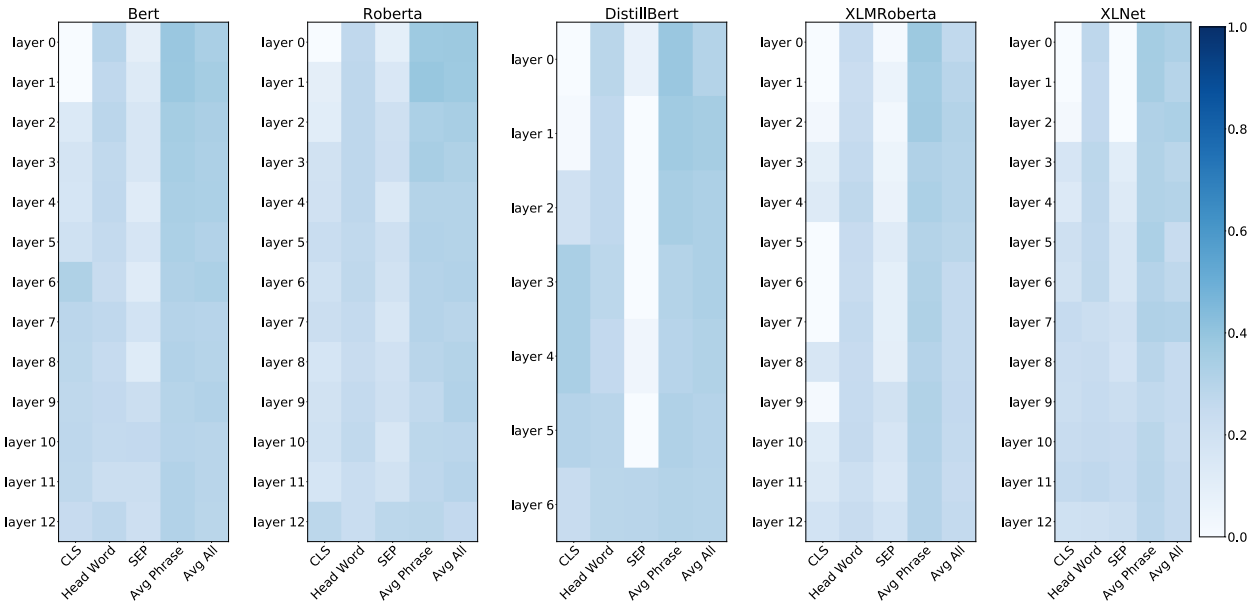

Figure 4.8: Accuracy difference on PPDB between full and controlled dataset, context-available input setting.

dataset setting. As discussed earlier, the context words boost the performance of later layers by pushing models to maintain lexical information, rather than compositional information.

4.6 Feature importance analysis

In this section, we present the analysis using LIME. Figure 4.9 shows the importance analysis on classifiers trained on PPDB classification tasks, and Table 4.5 shows the raw importance of BERT representations.

Among representation types, feature importance shows that classifiers prefer Head-Word as the most important representation for the binary classification. Even though accuracy of using Head-Word only is not as good as Avg-Phrase and Avg-All in correlation and classification tasks (as discussed in Section 4.5), Head-Word is considered more informative when all representation types are concatenated. It suggests that with information about phrases available (from Avg-Phrase and Avg-All), Head-Word contains most indicative information on identifying paraphrases. It is not surprising that Avg Phrase is selected as

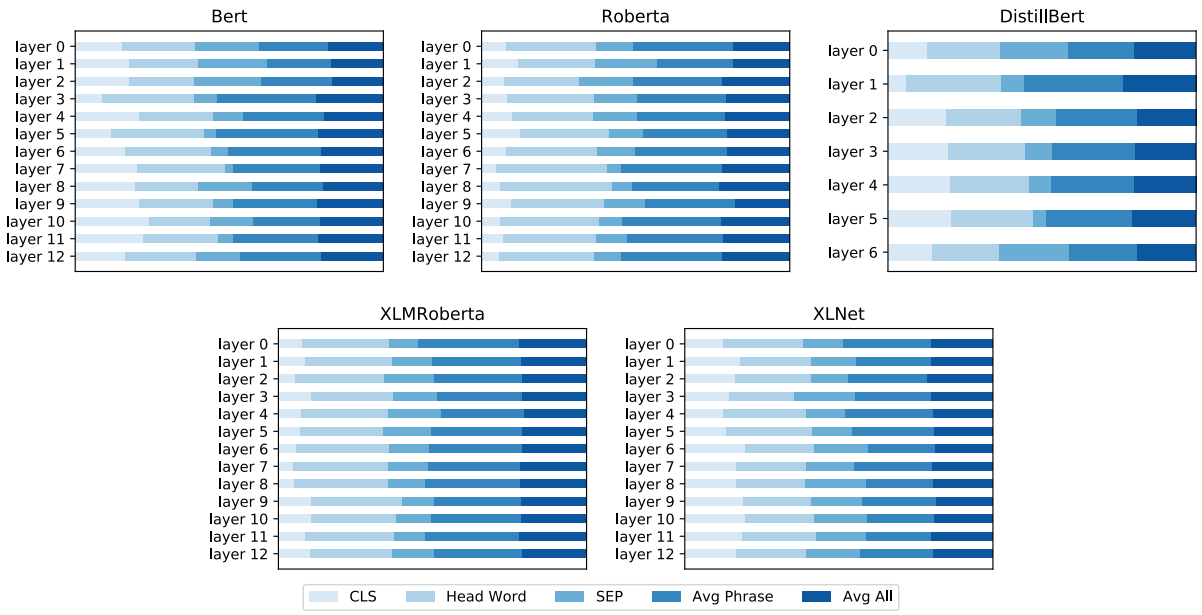

Figure 4.9: LIME experiments. Feature importance analysis of classifiers trained on PPDB classification tasks. Feature weights are normalized for each sample.

the second important features, with almost same weight as Head Word. Whereas SEP is assigned almost no weight, which accords with our previous experiments. Another notable observation is that CLS in XLM-RoBERTa is assigned significantly less weights compared to CLS in other models. It suggests that CLS in XLM-RoBERTa contains little information, which explains the fluctuation and weak performance we see in Figure 4.5.

Moving to the dynamics of weights as layer progresses, we see that proportion of weights for CLS peaks at middle layers (layer 4 for BERT, layer 5 for RoBERTa). We find a similar pattern on correlation task where the performance of CLS reaches highest at middle layer. The LIME experiment support the previous finding from another angle, that CLS requires some layer of composition to include most information about input phrases, and as layer progresses, CLS embeddings incorporate other information and lose the phrasal encoding.

Layer	CLS	Head-Word	SEP	Avg-Phrase	Avg-All
0	0.15	0.24	0.21	0.22	0.18
1	0.18	0.23	0.22	0.21	0.17
2	0.18	0.21	0.22	0.23	0.17
3	0.09	0.30	0.07	0.32	0.22
4	0.21	0.24	0.10	0.26	0.19
5	0.12	0.30	0.04	0.33	0.21
6	0.16	0.28	0.06	0.30	0.20
7	0.20	0.28	0.03	0.28	0.21
8	0.20	0.20	0.18	0.23	0.20
9	0.21	0.24	0.06	0.27	0.22
10	0.24	0.20	0.14	0.22	0.20
11	0.22	0.24	0.05	0.27	0.21
12	0.16	0.23	0.14	0.26	0.20

Table 4.5: Feature weights of classifiers trained on normal PPDB classification with BERT representations.

4.7 Qualitative analysis: sense disambiguation

The above analyses rely on testing models' sensitivity to meaning similarity between two phrases. In this section we complement these analyses with another test aimed at assessing phrasal composition: testing models' ability to select the correct senses of polysemous words in a composed phrase, as proposed by (Kintsch, 2001).

4.7.1 Landmark experiment

Figure 4.10 shows the percentage of phrases that fall closer to the correct landmark word than to the incorrect one, averaged over 16 phrase-landmark word pairs. We see strong overall performance across models, suggesting that the information needed for this task is successfully captured by these models' representations. Additionally, we see that the patterns largely mirror the results above for correlation and classification on uncontrolled datasets. Particularly, Avg-Phrase and Avg-All show comparatively strong performance across models. RoBERTa and XLNet show stronger performance in early layers, dropping off in later layers, while BERT and DistilBERT show more consistency across layers. XLM-RoBERTa and

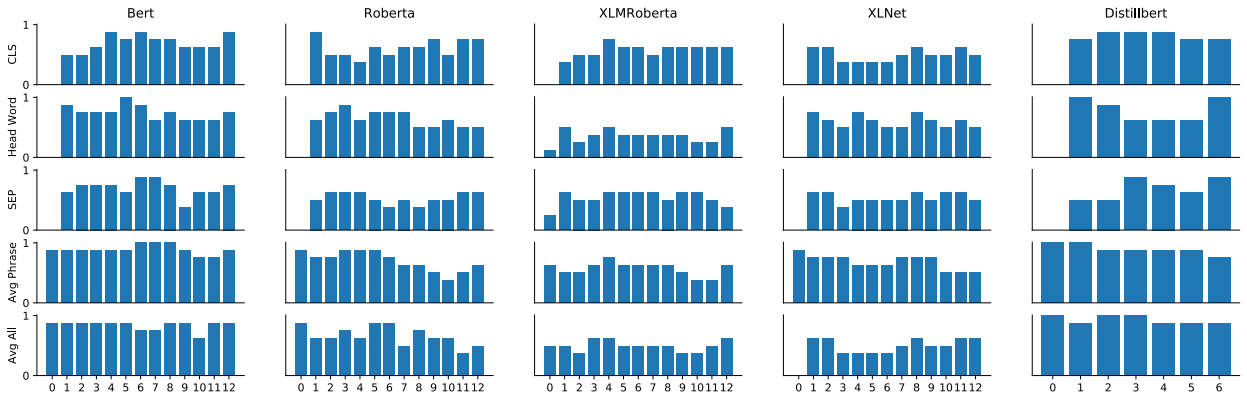

Figure 4.10: Landmark experiments. Y-axis denotes the percentage of samples that are shifted towards the correct landmark words in each layer. Missing bars occur when representations are independent of input at layer 0, such that cosine similarity between phrases and landmarks will always be 1.

XLNet show lower performance overall.

For this verb sense disambiguation analysis, the Head-Word token is of note because it corresponds to the central verb of interest, so its sense can only be distinguished by its combination with the other word of the phrase. XLM-RoBERTa has the weakest performance with Head-Word, while BERT and DistilBERT demonstrate strong disambiguation with this token. As for the CLS token, RoBERTa produces the highest quality representation at layer 1, and BERT outperforms other models starting from layer 6, with DistilBERT also showing strong performance across layers.

Notably, the observed parallels to our correlation and classification results are in alignment with the uncontrolled rather than the controlled versions of those tests. So while these parallels lend further credence to the general observations that we make about phrase representation patterns across models, layers, and representation types, it is worth noting that these landmark composition tests may be susceptible to lexical effects similar to those controlled for above. Since these test items are too few to filter with the above methods, we leave in-depth investigation of this question to future work.

4.7.2 Inference experiment

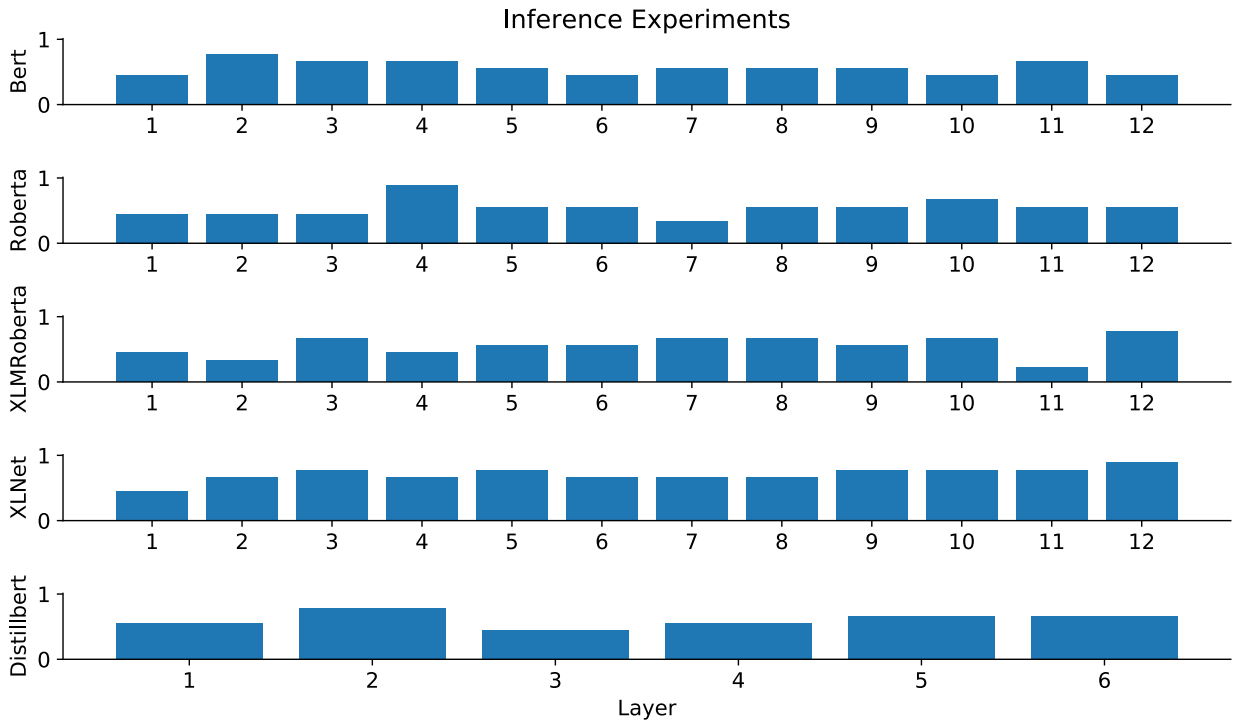

Figure 4.11: Accuracy on inference experiments. Y value denotes the percentage of CLS representations that reside closer to the correct inference candidate in terms of cosine similarity distance.

Figure 4.11 shows the accuracy of CLS representations that reside closer to the correct candidate inference in cosine distance space for each model. Overall, BERT consistently outperforms other models in lower layers while XLNet shows strong performance in deeper layers. RoBERTa shows a similar pattern as BERT, where the performance peaks at early layers, and degrades as layer continues. Although the task setup is similar to natural language inference task, the performance we observe largely mirrors the trend of full correlation and classification tasks. It is similar to what we see in landmark experiment, which indicates that these tests that are traditionally believed to be composition tasks, might require more lexical knowledge rather than higher-level compositional information. We speculate that there are spurious cues that make the models able to infer correct candidate without understanding

the input sentences. One potential cue can be the connection between subject of the source sentence and that of the correct candidate. We leave closer investigation for future work.

4.8 Discussion

The analyses reported above yield two primary takeaways. First, they shed light on the nature of these models' phrase representations, and the extent to which they reflect word content versus phrasal composition. At many points in these models there is non-trivial alignment with human judgments of phrase similarity, paraphrase classification, and verb sense selection. However, when we control our correlation and classification tests to remove the cue of word overlap, we see little evidence that the representations reflect sophisticated phrase composition beyond what can be gleaned from word content. While we see strong performance on classic sense selection items designed to test phrase composition, the observed results largely parallel those from the uncontrolled versions of the correlation and classification analyses, suggesting that success on this landmark test may reflect lexical properties more than sophisticated composition. Given the importance of systematic meaning composition for robust and flexible language understanding, based on these results we predict that we will see corresponding weaknesses as more tests emerge for these models' handling of subtle meaning differences in downstream tasks.

Our systematic examination of models, layers and representation types yields a second takeaway in the form of practical implications for selecting and extracting representations from these models. For faithful representations of word content, Avg-Phrase is generally the strongest candidate. If only the phrase is embedded, drawing from earlier layers is best in RoBERTa, XLM-RoBERTa, and XLNet, while middle layers are better in BERT, and later layers in DistilBERT. If the phrase is input as part of a sentence, middle layers are generally best across models. Though the CLS token is often interpreted to represent a full input sequence, we find it to be a poor phrase representation even with phrase-only input,

with the notable exception of the final layer of DistilBERT.

As for representations that reflect true phrase meaning composition, we have established that such representations may not currently be available in these models. However, to the extent that we do see weak evidence of potential compositional meaning sensitivity, this appears to be strongest in DistilBERT's CLS token in final layers, in RoBERTa's Avg-Phrase representation in later layers, and in XLM-RoBERTa's Avg-Phrase representation from later layers *only* when the phrase is contained within a sentence context.

4.9 Conclusions

We have systematically investigated the nature of phrase representations in state-of-the-art transformers. Teasing apart sensitivity to word content versus phrase meaning composition, we find strong sensitivity across models when it comes to word content encoding, but little evidence of sophisticated phrase composition. The observed sensitivity patterns across models, layers, and representation types shed light on practical considerations for extracting phrase representations from these models.

Future work can apply these tests to a broader range of models, and continue to develop controlled tests that target encoding of complex compositional meanings, both for two-word phrases and for larger meaning units. We hope that our findings will stimulate further work on leveraging the power of these generalized transformers while improving their capacity to capture compositional meaning.

Majority of the work in this chapter is published in Yu and Ettinger (2020).

CHAPTER 5

INTERPLAY BETWEEN FINE-TUNING AND COMPOSITION

5.1 Introduction

Transformer neural language models like BERT (Devlin et al., 2019), GPT (Radford et al., 2018, 2019) and XLNet (Yang et al., 2019b), have improved the state-of-the-art in many NLP tasks since their introduction. The versatility of these pre-trained models suggests that they may acquire fairly robust linguistic knowledge and capacity for natural language "understanding". However, an emerging body of analysis (Niven and Kao, 2019; Kim and Linzen, 2020; Ettinger, 2020) demonstrates a level of superficiality in handling of language.

As presented in the previous chapter, when testing for composition in pre-trained transformer representations, these representations reflect word content of phrases, but do not show signs of more sophisticated humanlike composition beyond word content. In this chapter we perform a direct follow-up of that study, asking whether models will show better evidence of composition after fine-tuning on tasks that are good candidates for requiring composition: 1) the Quora Question Pairs dataset in Paraphrase Adversaries from Word Scrambling (PAWS-QQP) (Zhang et al., 2019b), an adversarial paraphrase dataset forcing models to classify paraphrases with high lexical overlap, and 2) the Stanford Sentiment Treebank (Socher et al., 2013), a sentiment dataset with fine-grained phrase labels to promote composition. We base our analysis on the tests proposed in the previous chapter, which rely on alignment with human judgments of phrase pair similarities, and leverage control of lexical overlap to target compositionality. We fine-tune and test the same models, for optimal comparison.

We find that across the board, fine-tuning on PAWS does not improve compositionality—if anything, performance on composition metrics tends to degrade. Composition performance also remains low after training on SST, but we do see some localized improvements for certain models. Analyzing the PAWS dataset, we find reliable superficial cues to paraphrase labels

(distance of word swap), explaining in part why fine-tuning on that task might fail to improve compositionality—and reinforcing the need for caution in interpreting difficulty of NLP tasks. We also discuss the contribution of variation in size of labeled phrases in SST, with respect to the benefits that result from fine-tuning on that task. All experimental code and data will be made available for further testing.

5.2 Fine-tuning pre-trained transformers

In response to the weaknesses observed by (Yu and Ettinger, 2020), we select two different datasets with promising characteristics for addressing these weaknesses. We fine-tune on these tasks, then perform layer-wise testing on contextualized representations from the fine-tuned models, comparing against results on the pre-trained models. Here we describe the two fine-tuning datasets.

5.2.1 PAWS: fine-tuning on high word overlap

Sentence 1	**Sentence 2**	**Label**
There are also specific discussions , public profile debates and project discussions .	There are also public discussions , profile specific discussions , and project discussions .	0
She worked and lived in Stuttgart , Berlin (Germany) and in Vienna (Austria) .	She worked and lived in Germany (Stuttgart , Berlin) and in Vienna (Austria) .	1

Table 5.1: Example pairs from PAWS-QQP. Both positive and negative pairs have high bag-of-words overlap.

The core of the (Yu and Ettinger, 2020) finding is that model performance on the selected composition tests degrades significantly when cues of lexical overlap are controlled. It stands to reason, then, that a model trained to discern meaning differences under conditions of high lexical overlap may improve on these overlap-controlled composition tests. This drives our selection of the Paraphrase Adversaries from Word Scrambling (PAWS) dataset (Zhang et al.,

2019c), which consists of sentence pairs with high lexical overlap. The task is formulated as binary classification of whether two sentences are paraphrases or not. State-of-the-art pre-trained models achieve only $< 40\%$ accuracy before training on the dataset (Zhang et al., 2019b). Table 5.1 shows examples from this dataset. Due to the high lexical overlap, we might expect that in order to achieve non-trivial accuracy on this task, models must attend to more sophisticated meaning information than simple word content.

5.2.2 SST: fine-tuning on hierarchical labels

Another dataset that has been associated with training and evaluation of phrasal composition is the Stanford Sentiment Treebank, which contains syntactic phrases of various lengths, together with fine-grained human-annotated sentiment labels for these phrases. Because this dataset contains annotations of composed phrases of various sizes, we can reasonably expect that training on this dataset may foster an increased sensitivity to compositional phrase meaning. We formulate the fine-tuning task as a 5-class classification task following the setup in (Socher et al., 2013). The models are trained to predict sentiment labels given phrases as input.

5.3 Representation evaluation

For optimal comparison of the effects of fine-tuning on the above tasks, we replicate the tests, representation types, and models reported on by (Yu and Ettinger, 2020). Here we briefly describe these methods. For more details on the evaluation dataset and task setup, please refer to (Yu and Ettinger, 2020).

5.3.1 Evaluation tasks

(Yu and Ettinger, 2020) propose two analyses for measuring composition, which we apply to our fine-tuned models: similarity correlations and paraphrase classification. They focus on two-word phrases, using the BiRD (Asaadi et al., 2019) bigram relatedness dataset for similarity correlations, and the **PPDB 2.0** (Pavlick et al., 2015) paraphrase database for paraphrase classification.

For both task types, (Yu and Ettinger, 2020) compare between "normal" and "controlled" tests, with the latter filtering the data to control word overlap, such that amount of word overlap can no longer be used as a cue to improve performance.

It is on these controlled settings that (Yu and Ettinger, 2020) observe the significant drop in performance, concluding that model representations lack the compositional knowledge to discern phrase meaning beyond word content.[1]

5.3.2 Representation types

Following (Yu and Ettinger, 2020), for each input phrase we test as a potential representation 1) CLS token, 2) average of tokens within the phrase (Avg-Phrase), 3) average of all sentence tokens (Avg-All), 4) embedding of the second word of the phrase, intended to approximate the semantic head (Head-Word), and 5) SEP token. We test each of these representations at every layer of each model.

5.4 Experimental setup

We fine-tune and analyze the same models that (Yu and Ettinger, 2020) test in pre-trained form: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al.,

1. Like (Yu and Ettinger, 2020), we also test both phrase-only input (encoder input consists only of two-word phrase plus special CLS/SEP tokens), as well as inputs in which phrases are embedded in sentence contexts.

2019), XLNet (Yang et al., 2019b) and XLM-RoBERTa (Conneau et al., 2019). In each case, the pre-trained "base" version is used as the starting point for fine-tuning. We use the implementation of (Wolf et al., 2019)[2] based on PyTorch (Paszke et al., 2019).

We fine-tune these models on the two datasets described in Section 5.2. The Quora Question Pairs dataset in Paraphrase Adversaries from Word Scrambling (PAWS-QQP)[3] consists of a training set with 11,988 sentence pairs, and a dev/test set with 677 sentence pairs. Tuning on PAWS-QQP is formulated as binary classification. Sentences are passed as input and models are trained to predict whether two input sentences are paraphrases or not. Models are trained on the training set, and validated on the dev/test set for convergence.

The Stanford Sentiment Treebank (SST)[4] (Socher et al., 2013) contains 215,154 phrases. 15% of the data is reserved for validation. The fine-tuning task is formulated as 5-class classification on sentiment labels, where models are given phrases as input, and asked to predict sentiment. In both tasks, the Adam optimizer (Kingma and Ba, 2014) with default weight decay is used. Models are trained until convergence on the validation set.

The evaluation tasks consist of correlation analysis and paraphrase classification. For correlation in the "original" setting, we use the complete BiRD dataset, containing 3,345 phrase pairs.[5] In the controlled setting from (Yu and Ettinger, 2020), the data consists of 410 "AB-BA" mirror-image pairs with 100% word overlap (e.g., *law school* / *school law*). For the classification tasks, we use the preprocessed data released by (Yu and Ettinger, 2020).[6] We collect 12,036 source-target phrase pairs from the preprocessed dataset for our uncontrolled classification setting, and for the controlled classification setting, we collect 11,772 phrase pairs with exactly 50% word overlap in each pair.

2. https://github.com/huggingface/transformers

3. https://github.com/google-research-datasets/paws

4. https://nlp.stanford.edu/sentiment/treebank.html

5. http://saifmohammad.com/WebPages/BiRD.html

6. https://github.com/yulang/phrasal-composition-in-transformers

5.5 Results after fine-tuning

5.5.1 Full datasets

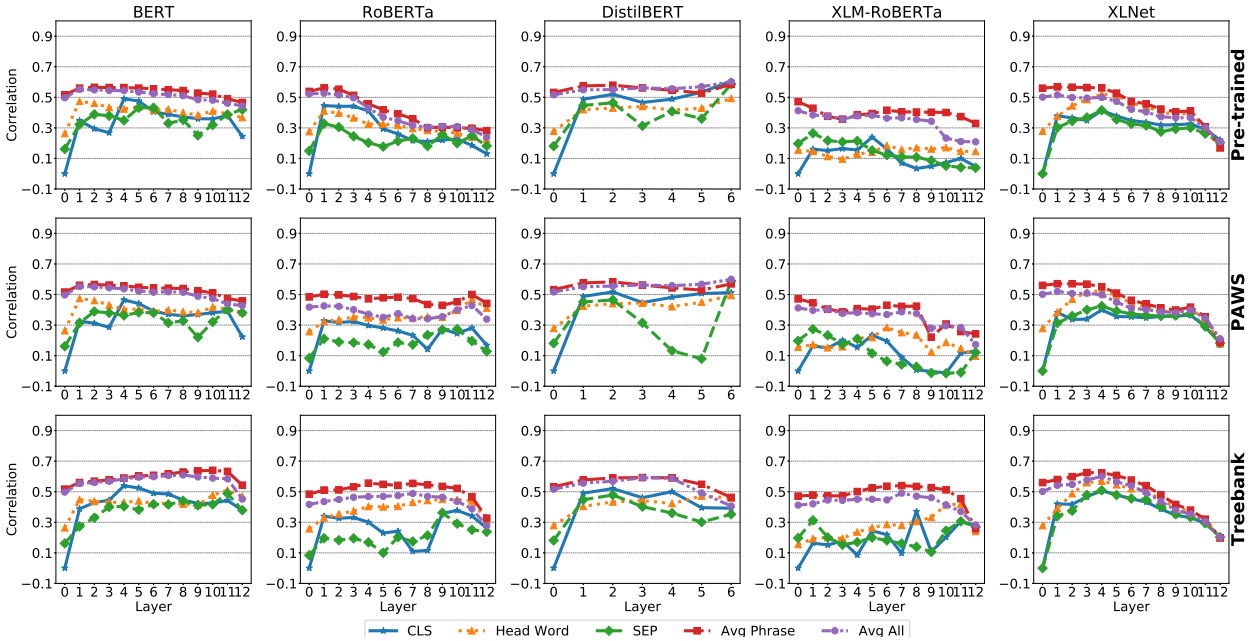

Figure 5.1: Correlation on BiRD dataset with phrase-only input. First row shows correlation of pre-trained models, second row shows models tuned on PAWS-QQP, and last row shows models tuned on Sentiment Treebank. X-axis of each subplot corresponds to layer index, and Y-axis corresponds to the correlation value. Layer 0 corresponds to input embeddings passing to the model.

Similarity Correlation Figure 5.1 presents the original results from (Yu and Ettinger, 2020) on pre-trained models, alongside our new results after fine-tuning, on the full BiRD dataset. Since this is prior to the control of word overlap, these correlations can be expected to reflect effects of lexical content encoding, without yet having isolated effects of composition. We see that overall, all models benefit from fine-tuning in this setting, with consistent improvements in peak correlations. For a given representation type, improvements are generally stronger after fine-tuning on SST than on PAWS. Between representation types, Avg-Phrase and Avg-All remain consistently at the highest correlations after fine-tuning.

Additionally, we see that the steady decline in correlation at later layers in pre-trained BERT, RoBERTa and XLM-RoBERTa is mitigated after fine-tuning. Model-wise, we see the most significant improvement in RoBERTa, where the correlations become more consistent across layers for all representation types except SEP. As we discuss below, we take this as indication that the fine-tuning promotes more robust retention of word content information across layers, if not more robust phrasal composition.

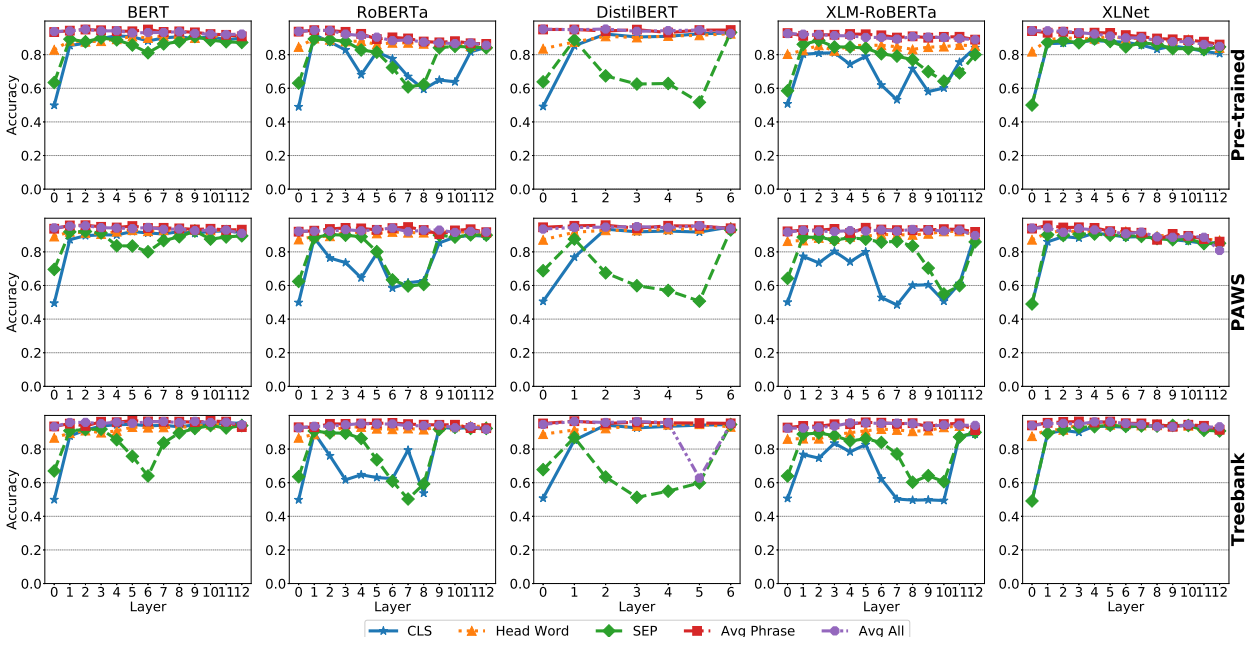

Figure 5.2: Accuracy on normal PPDB dataset with phrase-only input. First row shows accuracy of pre-trained models, second row shows models tuned on PAWS-QQP, and last row shows models tuned on Sentiment Treebank.

Paraphrase Classification Figure 5.2 shows the paraphrase classification accuracy on uncontrolled PPDB dataset. Overall, we see a similar pattern to the correlation task, where all models benefit from fine-tuning, and SST-tuned models demonstrate stronger peak performance compared to PAWS-tuned. Between representation types, Avg-Phrase and Avg-All show the strongest performance after fine-tuning. We find fine-tuning has the least impact on SEP across all models, and accuracy of SEP in middle layers of BERT is harmed by fine-tuning. It suggests that SEP contains little information on differentiating paraphrases,

which fine-tuning shows little changes in it. Layer-wise, fine-tuning mitigates the performance drop in later layers, which is also a notable pattern we observe in similarity correlation task. Model-wise, Avg-Phrase, Avg-All and Head-Word in later layers in RoBERTa and XLNet see the most prominent improvement from fine-tuning. CLS in XLM-RoBERTa demonstrates unstable fluctuation even after fine-tuning.

5.5.2 Controlled datasets

Above we see benefits of fine-tuning for performance on the full datasets—but the critical question here is whether correlations also show improved performance on the word-overlap controlled datasets, which better isolate effects of composition.

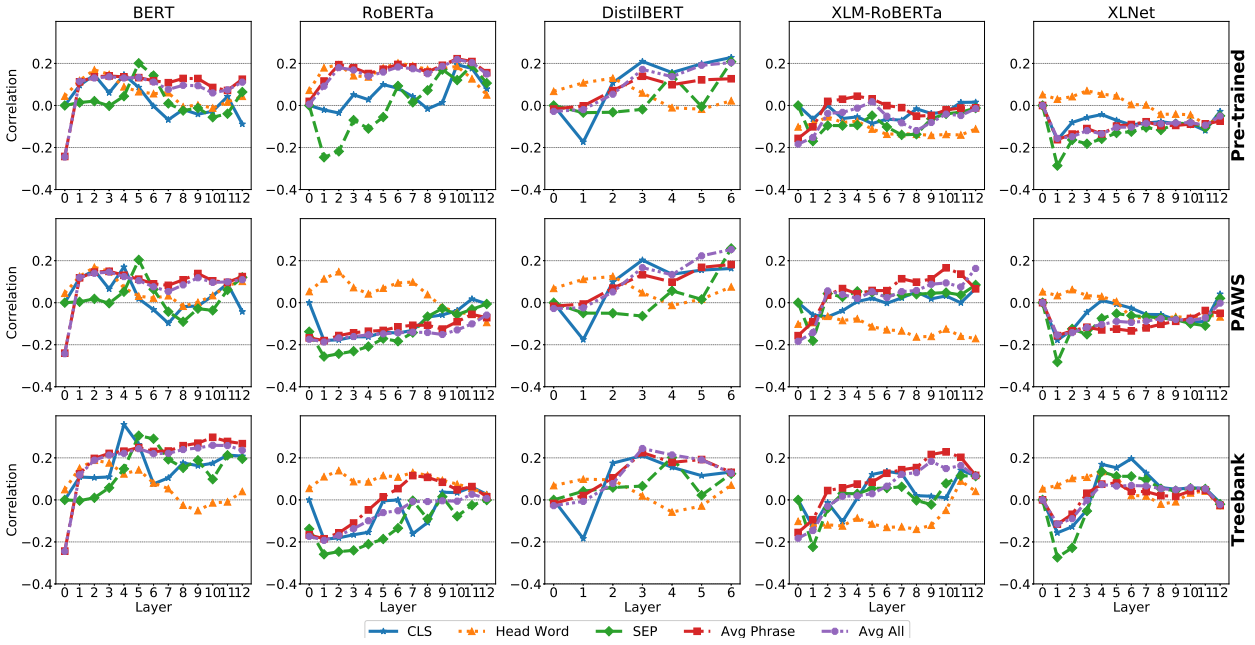

Figure 5.3: Correlation on controlled BiRD dataset (AB-BA setting) with phrase-only input. First row shows correlation of pre-trained models, second row shows models tuned on PAWS-QQP, and last row shows models tuned on Sentiment Treebank.

Figure 5.3 shows correlations for all models on the controlled AB-BA (full word overlap) correlation test. Figure 5.4 shows the results for the controlled paraphrase classification setting, where both paraphrase and non-paraphrase pairs have exactly 50% word overlap.

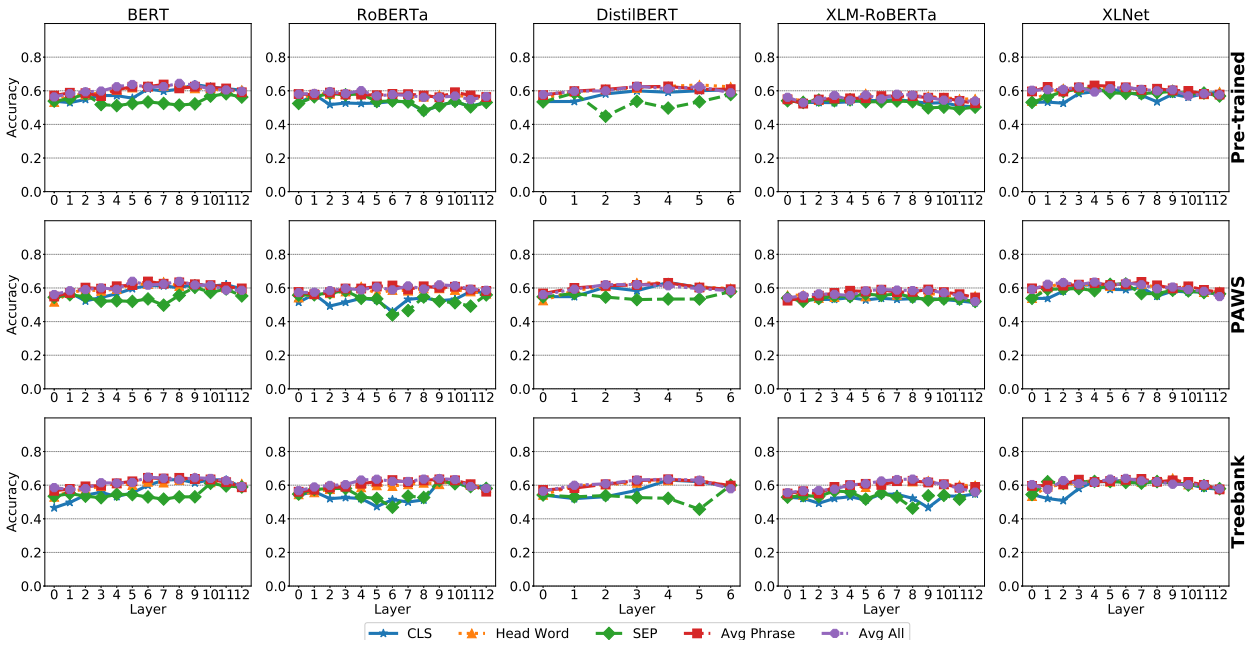

Figure 5.4: Accuracy on controlled PPDB dataset (exact 50% setting) with phrase-only input. First row shows accuracy of pre-trained models, second row shows models tuned on PAWS-QQP, and last row shows models tuned on Sentiment Treebank.

The first comparison to note is that between original and controlled settings, which allows us to establish the contributions of overlap information as opposed to composition. Comparing between Figure 5.1 and Figure 5.3, it is clear that fine-tuned models still show substantial drops in correlation when overlap cues are removed. The same goes for Figure 5.4 (by comparison to Figure 5.2)—we see that on the controlled dataset the fine-tuned accuracies hover just above chance-level performance both before and after fine-tuning, compared to over 90% accuracy on the uncontrolled dataset. This gap in performance between the original and controlled datasets mirrors the findings of (Yu and Ettinger, 2020), and suggests that even after fine-tuning, the majority of correspondence between model phrase representations and human meaning similarity judgments can be attributed to capturing of word content information rather than composition.

The second key comparison is between pre-trained and fine-tuned models within the overlap-controlled settings. While the prior comparison tells us that similarity correspondence

is still dominated by word content effects, this second comparison can tell us whether fine-tuning shows at least some boost in compositionality. Comparing performance of pre-trained and fine-tuned models in Figure 5.3, we see that fine-tuning on PAWS-QQP in fact consistently harms correlations in all models, except Avg-Phrase and Avg-All in XLM-RoBERTa. This is despite the fact that models achieve strong validation performance on PAWS-QQP (as shown in Table 5.2), suggesting that learning this task does little to improve composition. We will explore the reasons for this below.

In Figure 5.4, we see that fine-tuning does very slightly improves the best accuracies among models in the controlled datasets (around 3% increase in peak accuracy for SST and 2% for PAWS), but even so, the best accuracies among models continue to be only marginally above chance. This, too, fails to provide evidence of any substantial composition improvement resulting from the fine-tuning process.

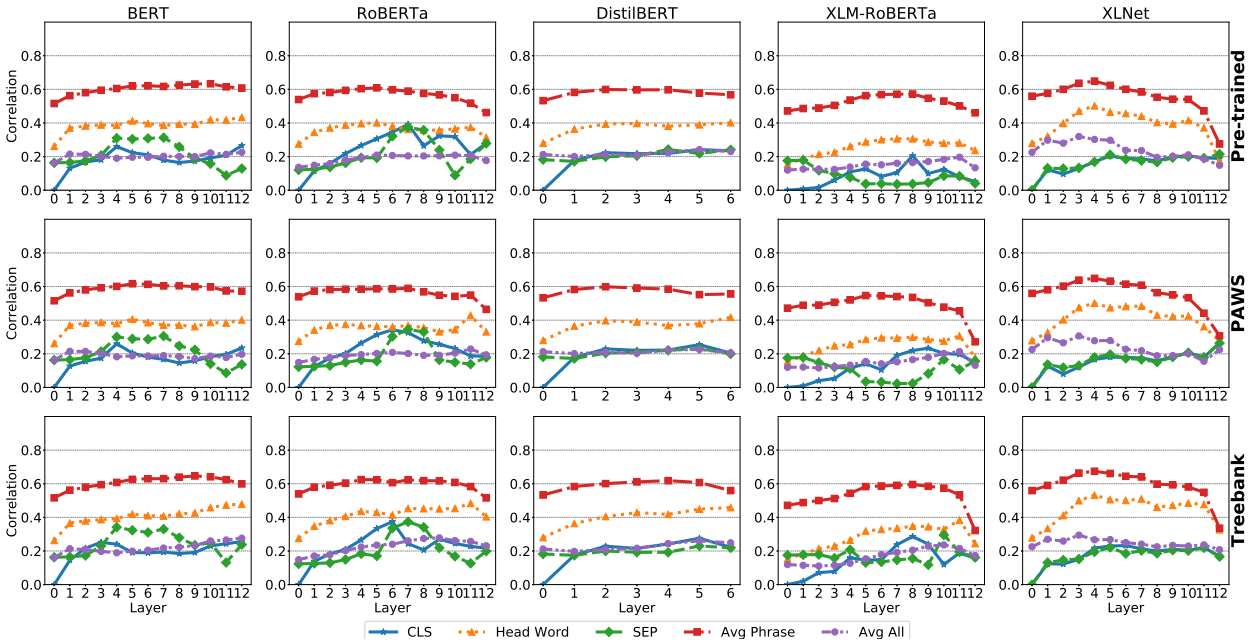

Figure 5.5: Correlation on full BiRD dataset with phrases embedded in context sentence (context-available input). First row shows correlation of pre-trained models,second row shows models tuned on PAWS-QQP, and last row shows models tuned on Sentiment Treebank. X-axis of each subplot corresponds to layer index, and Y-axis corresponds to the correlation value. Layer 0 corresponds to input embeddings passing to the model.

The story changes slightly when we turn to impacts of SST fine-tuning on correlations in Figure 5.3. While all correlations remain low after fine-tuning, we do see that correlations for BERT, XLM-RoBERTa and XLNet exhibit some non-trivial benefits from SST tuning. In particular, SST tuning consistently improves correlation among all representation types in BERT, boosting the highest correlation from ∼0.2 to ∼0.39. Between representation types, the greatest change is in the CLS token, with the most dramatic point of improvement being an abrupt correlation peak in the CLS token at BERT's fourth layer. We will discuss more below about this localized benefit from SST.

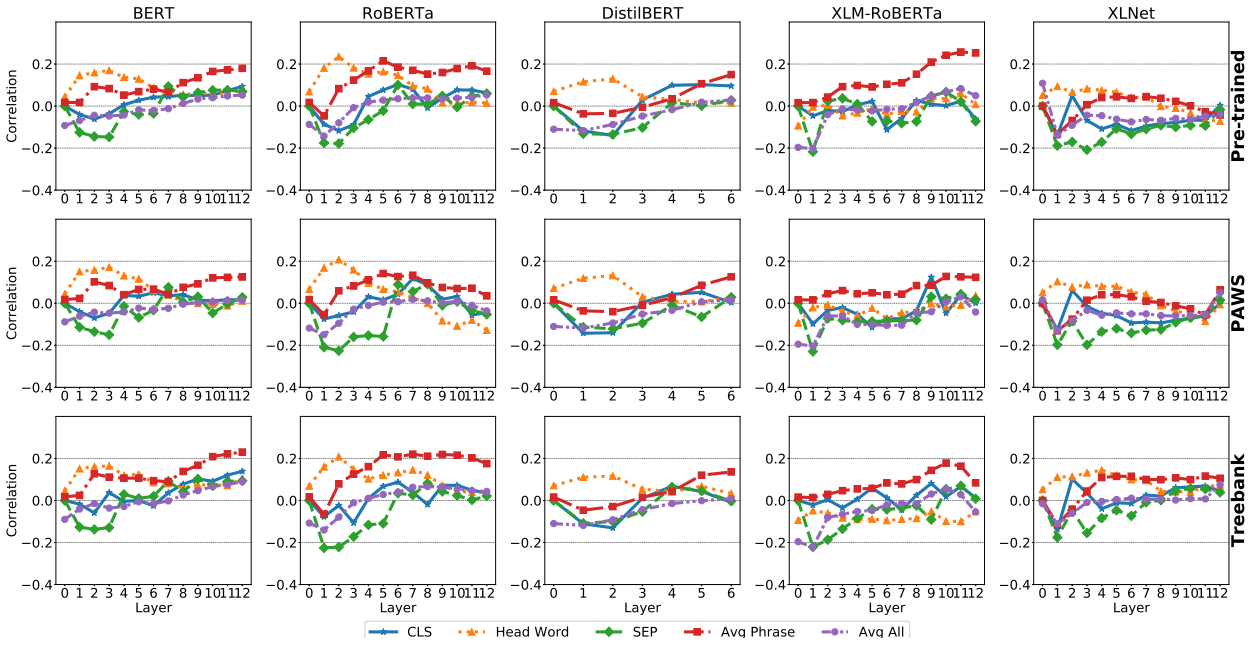

Figure 5.6: Correlation on controlled BiRD dataset (AB-BA setting) with phrases embedded in context sentence (context-available input). First row shows correlation of pre-trained models,second row shows models tuned on PAWS-QQP, and last row shows models tuned on Sentiment Treebank. X-axis of each subplot corresponds to layer index, and Y-axis corresponds to the correlation value. Layer 0 corresponds to input embeddings passing to the model.

A final important observation is that fine-tuning on either dataset harms the correlations for Avg-Phrase and Avg-All in RoBERTa under the controlled setting, by contrast to the general improvements seen for that and other models in the uncontrolled setting. This suggests that at least for that model, fine-tuning encourages retention or enhancement of

lexical information, but results in degradation of compositional phrase information.

5.5.3 Including sentence context

Following the setup discussed in Chapter 4, in addition to phrase-only inputs we also try embedding target phrases in sentence contexts. Figure 5.5 shows the result of similarity correlation with phrases embedded in context sentences. Compared to phrase-only setting in Figure 5.1, we see that presence of context words does boost overall correlation and accuracy, but does not alter the general trends. Avg-Phrase still dominates the correlation performance consistently across layers and models. Notably, the sharp correlation drop of Avg-Phrase and Head-Word in later layers are mitigated. It indicates that context words help Avg-phrase and Head-Word maintain more information about phrase similarity. Another trend is that correlation of Avg-All drops consistently, which is a result of the presence context words. Avg-All in this case incorporates information from context other than target phrases, leading to the consistent correlation degradation. Similar effect occurs to CLS, where it encodes more information not directly relevant to the phrase similarity task.

Comparing between pre-trained models and fine-tuned models, peak correlations are improved consistently after fine-tuning. Across all models, SST consistently has a greater improvement on peak performance. Model-wise, we find RoBERTa and BERT benefit more from fine-tuning on both tasks. Specifically, peak performance of BERT is boosted by $\sim 5\%$ in later layers, whereas RoBERTa peaks at middle layers, with around 3% correlation increase.

Figure 5.6 shows the impact of context on controlled BiRD dataset (AB-BA setting). Even with the presence of context words, models still show relatively weak performance on controlled tasks. Compared to phrase-only setting in Figure 5.3, we see that correlation of Avg-Phrase improves, especially in later layers. With context available, correlation no longer peaks in early layers as in phrase-only input, which implies that under context-available setting, it requires more layers of composition for these representations to capture phrase

similarity nuances. Similar to phrase-only, SST tuning consistently shows better improvement than PAWS. Since PAWS is specifically designed for paraphrase identification in sentence level, it is surprising that with context present, it under-performs SST, which consists of linguistic phrases of various length. In the next section, we will provide in-depth analysis on why PAWS largely fails. Another important observation is that fine-tuning mitigate the correlation drop in last layer of XLNet, and boost the peak correlation.

5.6 Analyzing impact of fine-tuning

The presented results suggest that despite compelling reasons to think that fine-tuning on the selected tasks may improve compositionality, these models mostly do not exhibit noteworthy benefits from fine-tuning. In particular, fine-tuning on the PAWS-QQP dataset mostly harms performance on the controlled datasets taken to be most indicative of compositionality. As for SST, the benefits are minimal, but in localized cases like BERT's CLS token, we do see signs of improved compositionality. In this section, we conduct further analysis on the impacts of fine-tuning, and discuss why tuned models behave as they do.

5.6.1 Failure of PAWS-QQP

Table 5.2 shows accuracy of fine-tuned models on the dev/test set of PAWS-QQP. The performance of BERT in the table is different from previous work mainly due to the fact that models in (Zhang et al., 2019b) are tuned on concatenation of QQP and PAWS-QQP dataset rather than PAWS only. It is clear that the models are learning to perform well on this dataset, but this does not translate to improved composition sensitivity.

We explore the possibility that this discrepancy may be caused by trivial cues arising during the construction of the dataset, enabling models to infer paraphrase labels without needing to improve their understanding of the meaning of the sentence pair (c.f., Poliak et al., 2018; Gururangan et al., 2018). Sentence pairs in PAWS are generated via word swapping and

Model	Accuracy(%)
BERT	80.13
RoBERTa	90.81
DistilBERT	81.98
XLM-RoBERTa	91.18
XLNet	88.24
Linear CLF	71.34

Table 5.2: Accuracy of fine-tuned models on the dev/test set of PAWS-QQP. Baseline is a linear classifier with relative swapping distance as the only input feature.

back translation to ensure high bag-of-words overlap (Zhang et al., 2019b). We hypothesize that models may be able to achieve high performance in this task based on distance of the word swap alone, without requiring any sophisticated meaning extraction.

To test this, given a sentence pair (s_1, s_2) with word counts l_1, l_2, respectively, we define "relative swapping distance" as

$$dist_{relative} = \frac{dist_{swap}}{max(l_1, l_2)}$$

where $dist_{swap}$ is defined as the index difference of the first swapping word in s_1 and s_2. For the example shown in the first row of Table 5.1, the first swapping word is "specific", with $dist_{swap} = 4$.

In the last plot of Figure 5.7, we show an association between relative swapping distance and paraphrase labels in the PAWS dev/test set: sentence pairs with small swapping distance tend to be positive samples, while large swapping distance associates with negative labels. Other plots in Figure 5.7 show the distribution of positive and negative predictions generated by each fine-tuned model with respect to relative swapping distance. We see a similar pattern, with models tending to generate negative labels when swapping distance is larger.

To verify the viability of this cue, we train a simple linear classifier on PAWS, with relative swapping distance as the only input feature. The results are reported as "Linear CLF" in Table 5.2. Even without access to the content of the sentences, we see that this simple model

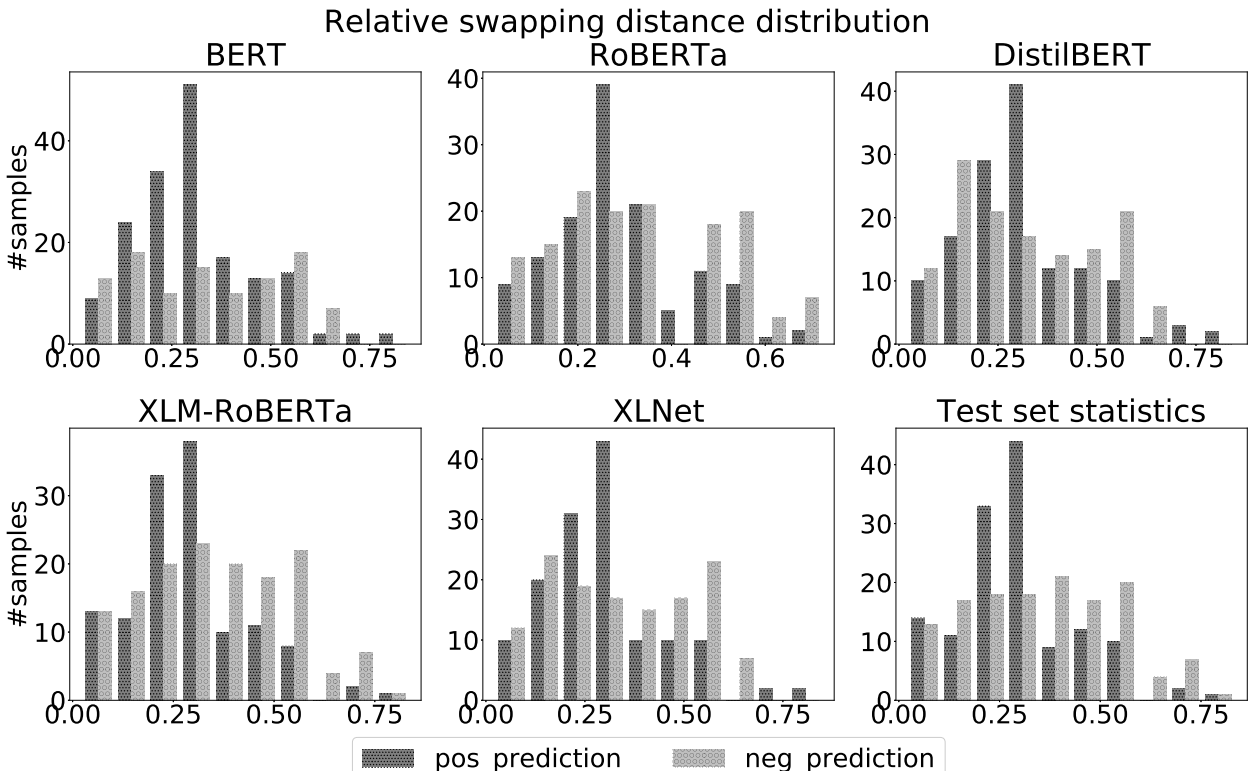

Figure 5.7: Distribution of positive and negative predictions made by tuned models. Last plot shows the statistics in the dev/test set. X-axis corresponds to the relative swapping distance, Y-axis shows the number of samples in the specific relative swapping distance bin.

is able to achieve non-trivial and comparably good classification accuracy on the dev/test set. The strong performance of the linear classifier and the distribution of predictions are consistent with the hypothesis that when we tune on PAWS-QQP, rather than forcing models to learn nuanced meaning in the absence of word overlap cues, we may instead encourage models to focus on lower-level information having little to do with the actual sentences, further degrading their performance on the composition tasks.

5.6.2 Localized impacts of SST

Fine-tuning on sentiment shows a bit of a different pattern—while it mostly shows only minor changes from pre-training, and the correlations and classification accuracies remain at decidedly low levels on the controlled settings, we do see in certain models some distinctive

changes in levels of similarity correlation as a result of tuning on SST. Notably, since these improvement patterns are seen in the similarity correlations but not in the classification accuracies, this suggests that these two tasks are picking up on slightly different aspects of phrasal compositionality. To investigate these effects further, we focus our attention on BERT, which shows the most distinctive improvement in correlations.

The obvious candidate for the source of the localized SST benefit is the dataset's inclusion of labeled syntactic phrases of various sizes. The benefits seen from SST-tuning suggest that this may indeed encourage models to gain finer-grained sensitivity to compositional impacts of phrase structure (at least those relevant for sentiment). To examine this further, we filter the SST dataset to subsets with phrases of the same length, from 2 to 6 words, and tune pre-trained BERT on each subset.

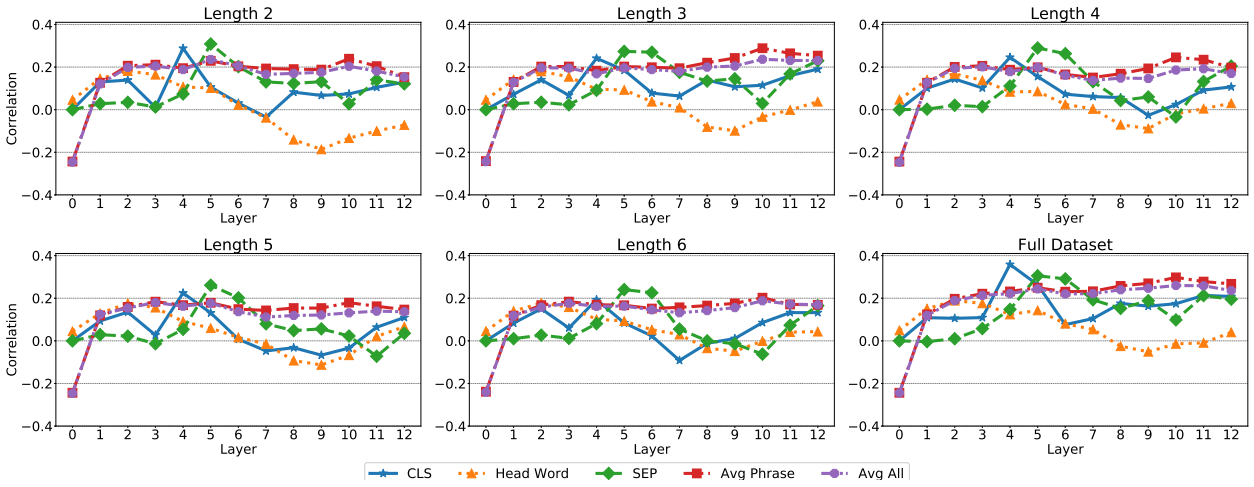

Figure 5.8: Layer-wise correlation of BERT fine-tuned on phrases of different lengths in SST.

Figure 5.8 shows the correlations for BERT, fine-tuned on each phrase length, on the overlap-controlled BiRD dataset. We see that tuning on the full dataset (mixed phrase lengths) gives the strongest fourth-layer boost in CLS correlation performance—but among the size subsets, a semblance of the fourth layer CLS peak is seen across phrase lengths, with length 2 training yielding the strongest peak, and length 6 training the smallest. This suggests an amount of size-based specialization—sentiment training on phrases of (or closer

to) length two has more positive impact on similarity correlations for our two-word phrases. However, we also see that phrases of other sizes contribute non-trivially to the ultimate correlation improvement from training on the full dataset. This is consistent with the notion that training on diverse phrase sizes encourages fine-grained attention to compositionality, while training on phrases of similar size may have slightly more direct benefit.

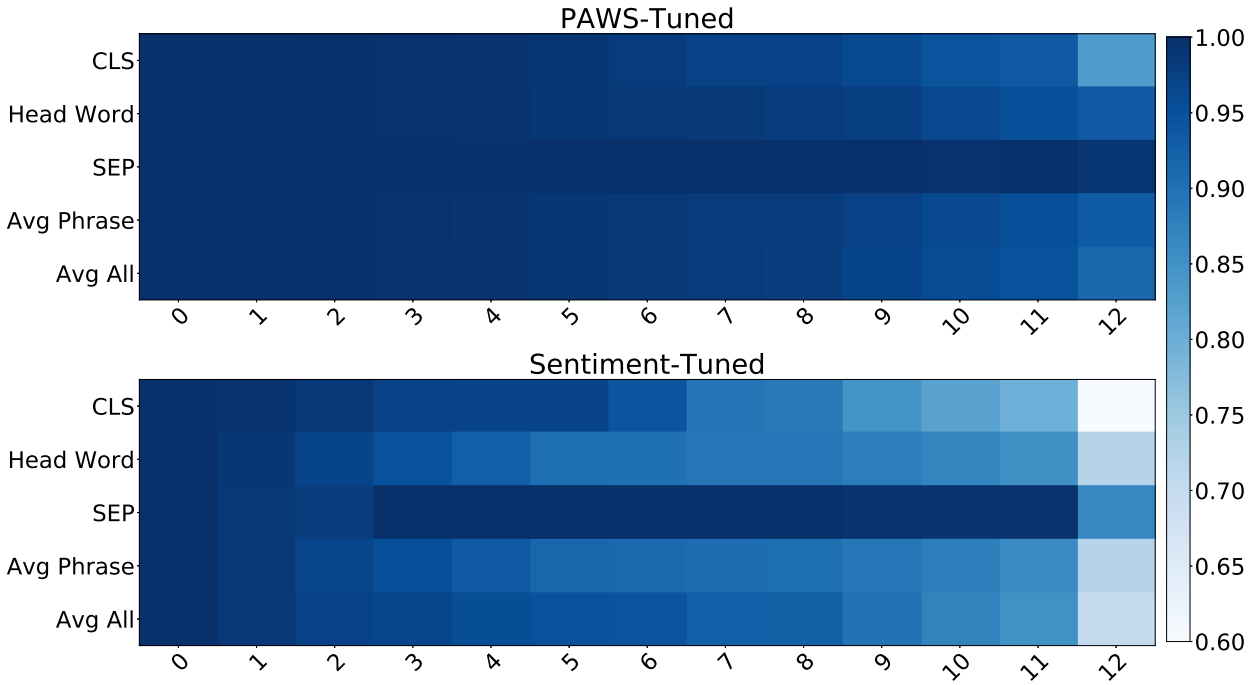

Figure 5.9: Average layer-wise embedding similarity between fine-tuned and pre-trained BERT. The upper half shows the comparison between PAWS-QQP tuned and pre-trained BERT. And the lower half presents Sentiment Treebank-tuned v.s. pre-trained. Embeddings are evaluated using full BiRD dataset for input.

5.6.3 Representation changes

For further comparison of fine-tuning effects between tasks, we analyze changes in BERT representations before and after fine-tuning. Figure 5.9 shows the results. We see substantial differences between tasks in terms of representation changes: while SST fine-tuning produces significant changes across representations and layers, PAWS fine-tuning leaves representations largely unchanged (further supporting the notion that this task can be solved fairly trivially).

We also see that after SST tuning, BERT's CLS token shows robust similarity to pre-trained representations until the fifth layer, followed by a rapid drop in similarity. This suggests that the fourth-layer correlation peak may be enabled in part by retention of key information from pre-training, combined with heightened phrase sensitivity from fine-tuning. We leave in-depth exploration of this dynamic for future work.

For PAWS tuning, similarity of all representation types decreases as layers progress, suggesting that fine-tuning has greater impact on later layers of transformers. And the only exception is SEP, where the impact of fine-tuning fades as layer progresses. It suggests that SEP plays very little role on fine-tuning tasks, and thus sees minor changes in deeper layers. In addition, the effect on tokens in later layers does not succeed in pushing later layers to focus more on higher-level information. As shown in the previous section, PAWS essentially makes models focus more on low-level lexical information in deeper layers.

For SST tuning, similarity evolution demonstrates a more diverse pattern: CLS shows decrease in similarity until layer 3, and a slight increase at layer 4. As the irregular variation matches the correlation peak in controlled BiRD task, we speculate that fine-tuning guides models to contain more nuances of composition in lower layers, and starting from layer 4, CLS starts to absorb information other than phrasal similarity (potentially global sentence information). Representations that directly correspond to phrase tokens (Avg-Phrase, Avg-All, Head-Word) show consistent decrease as layer progresses, suggesting fine-tuning alters deeper layers to contain more higher-level information. Like PAWS, SEP shows a contrary pattern of increasing similarity except last layer.

Overall, among all representation types, CLS shows the most notable changes, and SEP shows almost no changes except last layer, which is not surprising since CLS in last layer is used as the input for classification tasks during fine-tuning. And SST, to some extent, improves models' ability to capture compositional information, while PAWS essentially harms models' compositionality.

5.7 Discussion

The results of our experiments indicate that despite the promise of these two tasks for improving models' phrasal composition, fine-tuning on these tasks falls far short of resolving the composition weaknesses observed by (Yu and Ettinger, 2020). The majority of correspondence with human judgments can still be attributed to word overlap effects—disappearing once overlap is controlled—and improvements on the controlled settings are absent, extremely small, or highly localized to particular models, layers and representations. This outcome aligns with the increasing body of evidence that NLP datasets often do not require of models the level of linguistic sophistication that we might hope for—and in particular, our identification of a strong spurious cue in the PAWS dataset contributes to a number of findings emphasizing that NLP datasets often have artifacts that can inflate performance (Poliak et al., 2018; Gururangan et al., 2018; Kaushik and Lipton, 2018).

We do see a ray of promise in the small, localized benefits for certain models from tuning on SST. These improvements do not extend to all models, and are fairly small in the models that do see benefits—but as we discuss above, it appears that training on fine-grained syntactic phrase distinctions can indeed confer some enhancement of compositional meaning in phrase representations—at least when model conditions are amenable. Since sentiment information constitutes only a very limited aspect of phrase meaning, we anticipate that training on fine-grained phrase labels containing richer meaning information would be promising for promoting composition more robustly.

5.8 Conclusions

We have tested effects of fine-tuning on phrase meaning composition in transformer representations. Although we select tasks with promise to address composition weaknesses and reliance on word overlap, we find that representations in the fine-tuned models show little

improvement on controlled composition tests, or show only very localized improvements. Follow-up analyses suggest that PAWS-QQP has spurious cues that undermine learning of sophisticated meaning properties. However, results from SST tuning suggest that training on labeled phrases of various sizes may be effective for learning composition.

Future work should investigate how model properties interact with fine-tuning to produce improvements in particular models and layers—and should move toward phrase-level training with meaning-rich annotations, which we predict will be a promising direction for improving models' phrase meaning composition.

CHAPTER 6

COMPOSITION IN MODELS WITH EXPLICIT COMPOSITION STRUCTURE

6.1 Introduction

In previous chapters, we target our effort on Transformer-type models. We conclude that despite strong performance on downstream NLP tasks, these models show little sign of high level compositionality. We then select two tasks with promise to push models focusing more on compositional information and reduce reliance on lexical content. However, as discussed in Chapter 5, these efforts only show localized impact on pre-trained models, and models fail to capture higher-level compositional information despite improving on uncontrolled tasks. However, several possibilities remain to improve compositionality in language models:

1. Design a better fine-tuning task. As analyzed in Section 5.6, presence of spurious clues can trivialize the fine-tuning task, leaving little impact on improving compositionality. With careful control of lexical cues and a task requires rich compositional information, language models can potentially learn to focus composition of meaning. I will elaborate on this option in Chapter 7.

2. Incorporate explicit compositional architecture in model design. Possibility remains that the weak compositionality we observe in transformers is the limitation of the transformer architecture. In order to learn composition in language modelling, explicit composition structure in model design might be beneficial.

In this chapter, I will present further investigation on option 2. Specifically, I will investigate Recurrent Neural Network Grammars (RNNG) (Dyer et al., 2016)—a probabilistic model that explicitly models hierarchical relationships among words. With direct modeling of hierarchical structure of input sequences, we ask whether the model is able to capture

nuances of composition. To further isolate the impact of the compositional structure, we present in-depth analysis, by testing embeddings of RNNG from different model components respectively. We utilize two task sets: 1)the sentence probing tasks proposed by Ettinger et al. (2018), where workloads are generated with full control of lexical content, and biases of word pair order, word pair frequency are removed; and 2) similarity correlation and paraphrase classification proposed in Chapter 4. With normal and overlap controlled tasks, these evaluations are able to tease apart nuances of composition from lexical encoding.

We find that the model shows strong performance on tasks relying on lexical encoding. However, when it comes to tasks requiring higher level linguistic information (e.g. semantic role task), we see significant performance degradation, suggesting the failure of composing information beyond word content despite the presence of hierarchical composition component.

6.2 Analyzing composition in RNNG

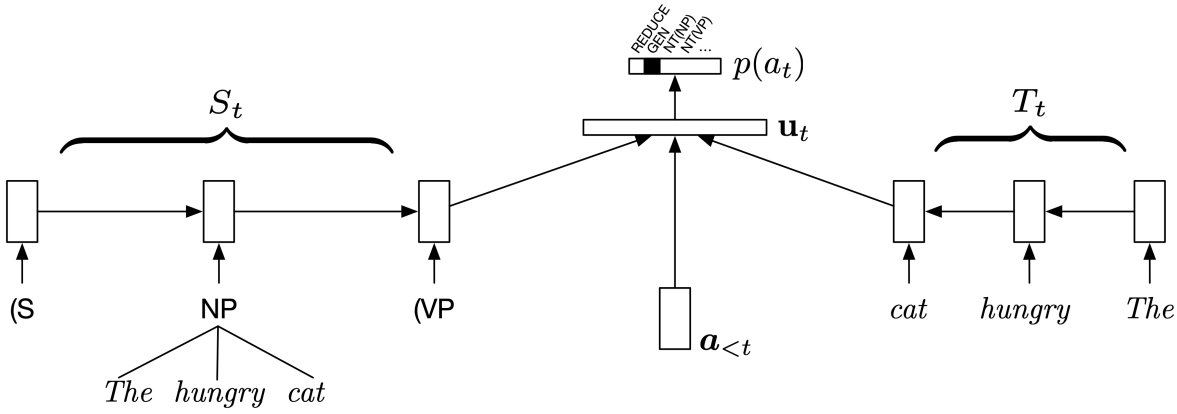

Figure 6.1: Illustration of model architecture from Dyer et al. (2016). At each time-step, the probability distribution of next action ($p(a_t)$) is computed based on three embeddings of different components: embeddings that represent the stack (S_t), output buffer of terminals (T_t) and history of actions ($a_{<t}$).

Figure 6.1 illustrates the state of RNNG on a certain time-step of input processing. At a certain time-step, the probability distribution of next action a_t is dependent on the repre-

sentations from three components: Stack (S_t) of partially completed syntactic constituents, history of actions ($a_{<t}$) and output buffer of terminals (T_t). The representations of action history and output buffer are encoded with standard RNN architectures, whereas embedding for Stack (S_t) is generated based on bidirectional LSTMs according to the parse tree structure. We consider the embedding for output buffer emb_{term} as embedding purely based on lexical tokens, and embedding for Stack emb_{stack} has potential for capturing compositional information, given the fact that accumulated representation is composed guided by syntactic parsing information. Since the embedding for action history is model-specific, in this chapter, we will focus on analyzing emb_{term} and emb_{stack}.

We complement the tasks discussed in Chapter 4 with a set of sentence probing tasks that consist of various tasks targeting at both lexical and compositional information. With these tasks, we are able to investigate the capability of word content encoding, as well as compositionality of the model.

6.2.1 Sentence probing tasks

The first set of tasks we use is sentence probing tasks proposed by Ettinger et al. (2018). Since RNNG is a model specifically designed to encode sentences, it produces clear accumulated sentence representation. The sentence representation makes it natural to apply sentence probing tasks in this chapter, in addition to evaluation tasks we use in previous chapters. Targeting at eliminating potential artifacts and superficial cues in the dataset, the sentence probing tasks utilize a specialized generation system to generates large number of examples with rich variations based on input lexical, semantic and syntactic constraints. Additionally, potential biases from lexical content and train/test split are controlled in the output dataset.

The set of tasks is formulated as classification probing tasks, which consists of the following tasks:

- **Content1Probe**: Given a sentence s and a verb v, it is formulated as a binary

classification task on whether the sentence contains the verb (or its conjugation).

- **Content2Probe**: Given a sentence s and a noun-verb pair n-v, it is formulated as a binary classification task on whether the sentence contains the noun-verb words.
- **Order**: Given a noun n, a verb v and a sentence s (containing both n and v), it is formulated as a binary classification on whether n occurs before v.
- **Negation**: Given a verb v and a sentence s (containing v, one negation and one other verb), the task is formulated as whether v is negated in s.
- **SemRole**: Given a noun n, a verb v and a sentence s (containing both n and v), binary classification on whether n is the agent to v.

Negation and **SemRole** have a specific focus on assessing composition in representations, while other tasks require more lexical knowledge to solve. In particular, **Negation** cannot be solved trivially by identifying the existence of negation word, and **SemRole** targets at systematic learning of semantic information.

6.2.2 Similarity correlation and paraphrase classification

In addition to the sentence probing tasks, we apply composition evaluation tasks discussed in Section 4.2. For similarity correlation with BiRD, we embed phrases on context sentences, and extract RNNG embeddings at the last time step of processing sentence as the embedding for the phrase. We also apply the controlled test as discussed in the early chapter. Similar to the setup in investigating transformers, We then correlate the cosine similarity between source and target phrases (embedded in sentences) with human judgment scores. For paraphrase classification, we follow the random sampling and controlled schemes discussed earlier. For both normal and controlled test, phrase pairs are embedded in context sentences, and embeddings at last time steps are extracted as representations. Representations of each phrase pair are then concatenated as the input to the classifier.

By applying aforementioned tasks, we are able to tease apart the capability of lexical encoding from composition of input sequence meaning.

6.3 Experimental setup

For the RNNG inspected in this chapter, we use the pre-trained generative model published by the author.[1] We experiment with three different versions of sentence embeddings from RNNG: Term, where we only use embeddings of the output buffer of terminals ($\boldsymbol{emb}_{term}$); Stack, where we only use embeddings of stack ($\boldsymbol{emb}_{stack}$); and Stack+Term, where we concatenate Term and Stack embedding as the representation for sentences ($\boldsymbol{emb}_{term+stack}$).

Each sentence probing task is formulated as binary classification task. We use the dataset published by the authors. [2] We follow the same setup as discussed in the original paper: for each task, we use a MLP classifier with a single hidden layer of size 256 with ReLU activation. Train/test split is maintained so that sentences and context of probing words have no overlap between train and test set. The concatenation of sentence embedding and probing word embeddings is passed to the classifier as input for the binary classification task. And the classifier is trained until convergence on training set. Each evaluation task consists of 4000 training samples and 1000 test samples. For each sample, sentence embeddings are generated using pre-trained RNNG, and embeddings for probing words are generated as one-hot vectors over vocabulary.

For similarity correlation and paraphrase classification tasks, we use the same datasets and settings presented in Chapter 4. Code of the experiments in this chapter is available at `https://github.com/yulang/rnng-composition`.

1. `https://github.com/clab/rnng`

2. `https://github.com/aetting/compeval-generation-system`

6.4 Results

6.4.1 Sentence probing

Embedding	Dim	Content1	Content2	Negation	Order	SemRole
Stack	256	59.88	51.2	51.1	76.1	51.1
Term	256	90.31	62.7	92.6	87.3	59.9
Stack + Term	512	84.64	60.0	91.7	87.9	52.0

Table 6.1: Performance (in percentage) of probing classifiers trained with RNNG embeddings on different sentence composition tasks. Dim denotes the dimension of the sentence embeddings passed to the classifier.

Table 6.1 shows the classification accuracy of RNNG representations on all evaluation tasks. The first finding to note is **Content1**. The task tests whether lexical content information is extractable from sentence embedding. $\boldsymbol{emb}_{stack}$ shows comparatively weak accuracy, while $\boldsymbol{emb}_{term}$ achieves strong performance. It suggests that embeddings from direct LSTM composition over lexical tokens maintain more lexical information compared to hierarchical composition according to parsing trees. The fact that $\boldsymbol{emb}_{stack}$ is composed hierarchically also leads to less extractability of lexical information. Additionally, concatenation embedding $\boldsymbol{emb}_{term+stack}$ achieves strong yet slightly weaker than $\boldsymbol{emb}_{term}$ performance, implying that most lexical information comes from $\boldsymbol{emb}_{term}$ and the introduction of $\boldsymbol{emb}_{stack}$ confuses the probing classifier, thus degrades classification accuracy. The result accords with our assumption that $\boldsymbol{emb}_{stack}$ has less focus on lexical information whereas $\boldsymbol{emb}_{term}$ contains rich lexical information as a result of vanilla recurrent composition of input tokens.

When it comes to probing noun-verb phrase (**Content2**), performance of all three embeddings drops significantly. $\boldsymbol{emb}_{term}$ and $\boldsymbol{emb}_{term+stack}$ still shows non-trivial accuracy, but $\boldsymbol{emb}_{stack}$ drops down to near random guess. Despite explicit composition based on syntactic constituents, $\boldsymbol{emb}_{stack}$ still show random performance. It supports the argument that lexical information is largely missing in $\boldsymbol{emb}_{stack}$, even though the embedding contains to some extent information about presence of syntactic constituents. We find similar pattern in

Negation as in **Content2**, where term embedding outperforms stack embedding. Note that the **Negation** task cannot be solved trivially by identifying the presence of negation word. The strong performance on this task with $\boldsymbol{emb}_{term}$ hints that vanilla LSTM composition is able to maintain negation information, however hierarchical composition falls short on maintaining relevant information. On **SemRole** task, we see particularly weak performance among all three representations. Sentence representations investigated in the original paper show lowest accuracy on this task, and it requires substantial ability to capture semantic role systematically (Ettinger et al., 2018). Above chance performance of $\boldsymbol{emb}_{term}$ reflects that it captures certain level of semantic information, but still largely misses abstract compositionality of the input sequence. And surprisingly, $\boldsymbol{emb}_{stack}$ performs at chance, indicating that it does not capture semantic information, despite the help of explicit composition structure.

6.4.2 Similarity correlation and paraphrase classification

Embedding	BiRD	BiRD Controlled	PPDB (%)	PPDB Controlled (%)
Stack	0.123	-0.061	81.73	**59.76**
Term	**0.283**	**-0.0286**	81.58	58.6
Stack+Term	0.228	-0.055	**82.51**	57.09

Table 6.2: Performance of RNNG embeddings on phrase similarity correlation and paraphrase classification tasks (under both normal and controlled settings).

Table 6.2 presents the performance of RNNG embeddings on similarity correlation and paraphrase classification. On similarity correlation task, all three representations show relatively weak performance even under normal setting. However, when it comes to PPDB classification task, the accuracy is competitively strong compared to Transformer models. The performance discrepancy between correlation and classification tests has two potential causes: 1) extracting information encoded in RNNG embeddings requires more complicated operation than cosine similarity, and MLP classifier is more capable of utilizing the composed information; 2) similarity correlation is a finer-grained test, while binary classification on

paraphrases requires less sophisticated information.

For normal correlation task, $\boldsymbol{emb}_{term}$ achieves the strongest performance. The observation aligns with the findings that uncontrolled task requires lexical information, and $\boldsymbol{emb}_{term}$ encodes more lexical information compared to $\boldsymbol{emb}_{stack}$. When lexical cues are controlled, correlations of all three embeddings drop to random, with $\boldsymbol{emb}_{term}$ showing slightly better correlation. The weak correlation is also consistent with the result from sentence probing, where $\boldsymbol{emb}_{stack}$ shows very weak performance on tasks requiring knowledge beyond lexical content (e.g. **Negation** and **SemRole**).

When it comes to paraphrase classification task, $\boldsymbol{emb}_{term+stack}$ shows the strongest performance. The observation indicates that with powerful extraction operation, information encoded in $\boldsymbol{emb}_{term}$ and $\boldsymbol{emb}_{stack}$ complement each other, yielding a stronger performance with concatenated embeddings. Another result of note is that with MLP operation, $\boldsymbol{emb}_{stack}$ outperforms $\boldsymbol{emb}_{term}$. It reflects that there are rich information regarding phrase content encoded in $\boldsymbol{emb}_{stack}$, but it is not easily extractable as $\boldsymbol{emb}_{term}$. Another observation to note is that $\boldsymbol{emb}_{stack}$ achieves highest accuracy under PPDB controlled setting. However, RNNG still under-performs transformers under controlled settings, suggesting RNNG does not do composition more than transformers. Additionally, we observe consistent significant performance gap between uncontrolled and controlled tasks, which suggests reliance of lexical content despite the presence of explicit compositional architecture.

6.5 Discussion

Table 6.3 summarizes the peak performance of all models investigated in this thesis so far. Between RNNG and transformers, we find more significant performance gap in correlation tasks than classification tasks. It is consistent with our arguments that information encoded in the RNNG embeddings requires more powerful extraction operations than transformer representations. Additionally, we find that for RNNG, the performance drop between

Model	BiRD	BiRD Controlled	PPDB	PPDB Controlled
RNNG	0.283	-0.0286	82.51	59.76
BERT	0.565	0.201	95.07	64.42
BERT (PT)	0.565	0.204	95.85	63.98
BERT (ST)	**0.640**	**0.359**	**96.88**	**64.80**
RoBERTa	0.562	0.222	94.58	60.07
RoBERTa (PT)	0.502	0.147	94.75	61.88
RoBERTa (ST)	0.556	0.139	95.55	63.91
DistilBERT	0.606	0.228	95.25	63.51
DistilBERT (PT)	0.600	0.258	95.85	63.30
DistilBERT (ST)	0.593	0.245	96.58	63.57
XLM-RoBERTa	0.471	0.044	92.83	58.31
XLM-RoBERTa (PT)	0.471	0.166	94.32	61.20
XLM-RoBERTa (ST)	0.540	0.228	95.88	64.12
XLNet	0.568	0.07	94.24	63.13
XLNet (PT)	0.571	0.062	95.71	63.61
XLNet (ST)	0.625	0.196	96.38	63.61

Table 6.3: Peak performance of all models investigated in this thesis on phrase similarity correlation and paraphrase classification tasks (under both normal and controlled settings, phrase-only). (PT) corresponds to PAWS-tuned model, and (ST) stands for SST-tuned model. For transformer models, peak performances are maximum across all representation types and all layers. Performance of BiRD and BiRD controlled are correlation value. Performance of PPDB and PPDB controlled are classification accuracy in %.

controlled and uncontrolled settings is less severe, suggesting less reliance on word overlap cues. However, for all 4 tasks, RNNG under-performs transformers. We argue that despite having explicit compositional structure, RNNG does not show better compositionality than transformers. Along with the results in Table 6.1, we argue that RNNG is capable of composing input sequences while maintaining information that contains word content and word order. However, the result representation largely misses systematic encoding about lexical and semantic information.

As discussed earlier, Stanford Sentiment Treebank tuning shows localized benefit to BERT. Among all transformers, BERT tuned on SST shows consistently the strongest peak performance for all tasks. We also see that a majority of PAWS tuned models report weaker performance on controlled tasks, due to the fact that PAWS tuning pushes models to focus

more on lower-level information, further degrading compositionality.

6.6 Conclusion

In this chapter, I investigate the compositionality in each component of RNNG. I evaluate the model with two sets of tasks: sentence probing tasks proposed in Ettinger et al. (2018) and similarity correlation/classification tasks discussed in Chapter 4. With control of lexical cues, both task sets shed light on model's compositionality beyond lexical sensitivity.

In sentence probing tasks, embeddings from vanilla composition report stronger performance on lexical-oriented tasks. And surprisingly hierarchical composition structure fail to yield representations that systematically capture semantic information. As for similarity correlation and paraphrase classification, though RNNG shows non-trivial alignment with human judgment, performance degradation when lexical cues are removed is still significant. With the explicit compositional structure, the model does not do composition more than transformers. However, the model is less sensitive to lexical content removal, suggesting less reliance on word overlap information.

CHAPTER 7

CONCLUSION

In this dissertation, I have proposed evaluation tasks and artifact-removal strategy that allow teasing apart nuances of phrase-level composition from effective lexical encoding. I have investigated a variety of state-of-the-art transformer models. By applying the proposed tasks and qualitative analysis to these models, I have shown that despite achieving strong performance on full datasets, the performance is significantly inflated by the superficial cues like lexical overlap and word content. With cues removed, performance of all models drops significantly, suggesting heavy reliance on lexical content to infer phrase similarity. I have further explored the potentials of improving compositionality of pre-trained language models by fine-tuning on tasks with promise to address composition weaknesses and reliance on word overlap. However, I have shown in the follow-up analysis that spurious cues in the adversarial dataset undermine the learning of sophisticated information, trivializing the fine-tune task. Even though the other dataset of sentiment composition demonstrates localized improvement, the improvement does not generalize to models other than BERT. Additionally, I have analyzed a model with explicit compositional architecture. By applying evaluation tasks proposed in this thesis and a set of sentence probing tasks, I have shown that the model does not have stronger composition capability than transformers, despite the presence of hierarchical compositional structure. However, the model is less sensitive to lexical cues removal, suggesting less reliance on word overlap information.

7.1 Overview

In Chapter 1, I began the discussion by laying out the motivation of interpretability project. I presented the overall organization of the dissertation, and highlighted the efforts of analyzing and improving compositionality in language models.

In Chapter 2, I systematically reviewed previous work related to this dissertation. Specifically, I discussed progress on text representation learning and language modeling, from context-independent word embeddings to recent transformer-based contextualized embeddings. And I gave an overview of methods to evaluate quality of text representations and LM models, including semantic-similarity-based datasets, multilingual datasets for cross-lingual language understanding and general-purpose language understanding benchmarks. On interpretability of neural models, I first discussed work on identifying notable failures of NLP models, resulting from biases and uncontrolled cues in datasets. Then I reviewed previous work on interpreting model predictions. One line of work correlate importance of input features with model input. Another line of work systematically introduce perturbation to input as explanations. I also presented overview of work on analysis of neural models. Particularly, I focused on growing body of work on analyzing transformer models, among which two trends prevail—classification-based probing and intrinsic analysis. Additionally, I reviewed previous attempts to highlight model weaknesses via adversarial attack, and efforts to understand impacts of fine-tuning process. Lastly, I investigated work on analyzing composition in language models, as well as incorporating composition in models' architecture to improve performance.

In Chapter 3, I presented a variety of representation types analyzed in this dissertation. For the reason that transformer does not have a clear aggregated representation corresponds to input phrase, we investigate different representation types in a layer-wise manner. Furthermore, for each phrase being investigated, I proposed to have phrase-only setting (where only phrase tokens and special tokens are passed as input) and context-available setting (where phrases are embedded in the context sentences extracted from Wikipedia).

In Chapter 4, I systematically investigated the nature of phrase representations in state-of-the-art transformers. I proposed a set of model-agnostic tasks targeting at assessing phrasal representation and composition. The methods begin with meaning similarity evaluation:

correlation with human judgment on phrase similarity, and ability to identify paraphrases. The first two tasks I proposed is similarity correlation and paraphrase classification. Motivated by the findings that there are correlation between word overlap of phrase pairs and meaning similarity, I proposed a control strategy for both tasks to hold word overlap among phrase pairs to be constant. With overlap controlled, models are unable to infer phrase similarity based on lexical content. In addition to these two tasks, I presented feature importance analysis, which interprets trained paraphrase classifier with explainable linear models. As a complement to foregoing analyses, I discussed two qualitative analyses—landmark and inference experiments. These two tests are popular tasks on analyzing composition in early works.

I then presented in-depth analyses on numerous state-of-the-art pre-trained by applying these tasks and investigating layer-wise performance of different representation types (discussed in Chapter 3). All models show non-trivial performance on full correlation and classification tasks, suggesting representations produced by these pre-trained models contain relevant information regarding phrases. However, when lexical cues are removed, performance of all models drops significantly. We concluded that these contextualized embeddings reflect heavy reliance on word content, but little nuances of compositional information. Feature importance analysis demonstrates similar conclusion, where representations directly related to phrase tokens contain more phrase similarity information. The findings from landmark and inference experiment mirror the performance on full datasets, suggesting that these traditional experiments are essentially tasks on lexical encoding rather than composition task as previously believed.

In Chapter 5, I delved deeper into improving compositionality in state-of-the-art transformers. Motivated by conclusions in previous chapter that pre-trained models show weak compositionality and reflect heavy lexical encoding, I selected fine-tuning tasks with promise to guide models to focus more on compositional and higher-level information. Specifically, I

explored tuning on PAWS—an adversarial paraphrase classification dataset with high lexical overlap pairs, and Standford Sentiment Treebank—a sentiment dataset contains syntactic phrases of various length, together with human-annotated sentiment labels. I then applied similarity correlation and paraphrase classification to fine-tuned models under both full and controlled settings. We found that fine-tuning on both datasets improves overall performance on full datasets. However, models still show weak (often slightly above chance) performance on controlled tasks. Detailed discussion shows that fine-tuning improves models on maintaining lexical information in deeper layers, but not focusing on higher level compositional nuances. I presented follow-up analyses on failures of PAWS, and localized impact of SST. I showed that spurious cues can trivialize the fine-tuning task, and provided in-depth discussion on potential directions to improve composition.

In Chapter 6, I expanded the scope of this dissertation to RNNG—a model with compositional architecture. With workloads targeting at sentence composition, I showed that RNNG is able to achieve strong performance on lexical-focused tasks, but falls short when it comes to tasks required higher level composition tasks. This chapter complements other chapters that focus on transformer models.

In summary, in this dissertation, I discussed my efforts centered on *composition* in neural language models. I take steps to tackle this notion: with a focus on transformer models, Chapter 3 and Chapter 4 explore the methods to clarify two important but distinct notions: lexical encoding—how faithfully models encode information about input tokens, and nuances of composition of phrase meaning—independent of word contents. I concluded that pre-trained models have nontrivial correlation with human judgment on full datasets, but largely miss composition of phrase meaning. Chapter 5 targets at a follow-up question—whether we can improve compositionality through fine-tuning. I showed that these tasks only show localized impact on pre-trained models, due to existence of spurious cues in the dataset, and lack of rich compositional information. I further explore the potentials of incorporating

composition in model architecture in Chapter 6. Through composition task for sentence vectors, I concluded that the model achieves strong performance on lexical-focus tasks, but is incapable of composing information beyond lexical content.

7.2 Future directions

In this section, I will propose future directions of this dissertation, and discuss potential concerns and topics for future work.

7.2.1 Beyond Phrasal Representation and Composition

The majority work described in this dissertation focuses on phrase level composition. Specifically, we focus on two-word phrases. As a starting point, these are the smallest phrasal unit and the most conducive to lexical controls. Two-word phrases allow us to control word overlap among phrase pairs to be exact 100% or 50% as we did in Chapter 4. When it comes to composition beyond two-word phrases, controlling lexical overlaps is more complicated. As embeddings generated by contextualized encoders are context-dependent, the idea of controlling lexical content can be generalized to sentence level, with careful control of contexts. To generate syntactically and semantically plausible sentences while having full control on superficial cues, the method proposed in Ettinger et al. (2018) is one potential solution, where sentences are generated based on pre-defined oracles of sentence, and words are selected from a set of candidates to plug in the oracle to generate final sentences. Another concern is that focusing on two-word phrases allows us to leverage larger amounts of annotated phrase similarity data. Admittedly, there are large amounts of datasets on sentence similarity. As discussed in Chapter 2 and Chapter 5, biases and annotation artifacts in these datasets might introduce noise to model's performance. In order to tease apart true compositional effect from performance inflation, further investigation and data processing might be required to remove potential cues and biases.

Even though sentence-level composition introduces additional complications, it opens up various possibilities:

1. More representation types and accumulated contextualized representations can be explored. A variety of pooling methods and attention-based representations are not included in this dissertation, for the reason that we look for embeddings of clear correspondence to original phrase tokens. Moving to sentence level, these methods can be applied and analyzed.

2. Impact of larger context can be investigated. In this dissertation, we investigate the impact of sentence context on phrasal composition. With sentence level composition, passage context and even the position of the target sentence in its document can also play a role on the accumulated representation.

3. Evaluation tasks can be applied to wider scope of model types. In this dissertation, we focus our work on transformer models and RNNG. Evaluation on representation and composition on sentence level can be extended to other model types, shedding light on difference in compositionality and lexical encoding across different architectures.

Besides generalizing to larger linguistic units, future work can also be done to extend evaluation tasks. Evaluation tasks proposed in Chapter 4 rely on meaning similarity of phrases. Evaluation criteria can be extended to generalizability, which is often considered as another signal for compositionality. In addition to cosine similarity and single hidden layer classifier used in this dissertation, future work can look into other approaches to extract information from contextualized embeddings. However, we argue that the extraction operations should be simple, since extractability of information is also a dimension of compositionality. Moreover, when using a powerful extraction operation (e.g. deep neural network), question arises on whether good performance results from strong capability of the extraction model or compositional information encoded in the embedding.

Furthermore, there are other potential annotation artifacts and biases in existing tasks. In this dissertation, we concentrate on removing lexical cues and word overlap bias. Future work can look into controlling other spurious cues that trivialize the task, such as subject-object order, length of input sequence, subject-verb combinations etc.

7.2.2 Improving Compositionality in Language Models

Results presented in Chapter 5 indicate localized improvement from fine-tuning. The follow-up analyses suggest that training on labeled phrases of various sizes may be effective for learning composition. Specifically, future directions with promise to show better compositionality include:

1. Further investigation on how model properties interact with fine-tuning to produce improvements in particular models and layers. Dynamics of self-attention mechanisms during fine-tuning can benefit the understanding of fine-tuning process, and thus improving composition in language models. Moreover, gradient-based analysis with saliency maps can provide insights in finer granularity.

2. Better fine-tuning tasks and datasets. As discussed in earlier chapters, we should move toward phrase-level training with meaning-rich annotations, which we predict will be a promising direction for improving models' phrase meaning composition. As discussed in failure of PAWS-QQP (Section 5.6), a key factor to consider is biases and artifacts in these tasks. In addition to choosing tasks with rich composition information, further processing to control potential biases and artifacts is necessary to push models towards better compositionality.

3. Delving deeper into the interplay between compositionality and performance of downstream tasks. In this dissertation, attempts are made to isolate composition nuances from performance on vanilla downstream tasks. In future work, it will be useful to

explore in this line—how compositionality benefits or harms performance of different NLP tasks. As an essential component of language understanding, we speculate that models with better compositionality will naturally demonstrate generalizability, which benefits a wide variety of tasks. We leave this line of analyses for future work.

7.2.3 Extractability of representations

This dissertation mainly utilizes cosine similarity operations and shallow MLP classifiers to extract information from representations. The central assumption being made is that composed information encoded in the representations should be extractable with simply operations. However, as shown in Chapter 6, weak performance on certain tasks does not necessarily imply missing information in the representations, but rather extraction operation is not powerful enough.

In particular, possibility remains that model captures higher-level information about input sequence, but such information is not easily extractable as lower-level lexical information. Future work can be applied to fine-grained investigation on extractability of these information. Specifically, analysis can focus on the dynamics between the extractability of representations produced by a certain model and its downstream performance on tasks require different level of information. In order to isolate the signals of extractability, necessary efforts have to made to tease apart the impact of the capability of extraction operation on the downstream task from the extractability of relevant information.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1823–1832.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving bert a calculator: Finding operations and arguments with reading comprehension. 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 5949–5954.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2015. Deep compositional question answering with neural module networks. corr abs/1511.02799 (2015). *arXiv preprint arXiv:1511.02799*.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big bird: A large, fine-grained, bigram relatedness dataset for examining semantic composition. 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 505–516.

Joris Baan, Jana Leible, Mitja Nikolaus, David Rau, Dennis Ulmer, Tim Baumgärtner, Dieuwke Hupkes, and Elia Bruni. 2019. On the realization of compositionality in neural networks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 127–137.

Geoff Bacon and Terry Regier. 2019. Does bert agree? evaluating knowledge of structure dependence through agreement relations. *arXiv preprint arXiv:1908.09892*.

Christine Basta, Marta R Costa-jussà, and Noe Casas. 2019. Evaluating the underlying gender bias in contextualized word embeddings. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 33–39.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Jerome R Bellegarda. 2000. Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76–84.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Ben Bogin, Sanjay Subramanian, Matt Gardner, and Jonathan Berant. 2020. Latent compositional representations improve systematic generalization in grounded question answering. *arXiv preprint arXiv:2007.00266*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Ting-Yun Chang and Yun-Nung Chen. 2019. What does this word mean? explaining contextualized embeddings with natural language definition. 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 6066–6072.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th international conference on computational linguistics*, pages 133–144.

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.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

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

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018b. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

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)*, pages 4908–4926.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

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.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 199–209.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

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.

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Denis Filimonov, Ravi Teja Gadde, and Ariya Rastrow. 2020. Neural composition: Learning to generate from multiple models. *arXiv preprint arXiv:2007.16013*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1307–1323.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. 2004. Integrating topics and syntax. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, pages 537–544.

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*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933.

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

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Bo-June Paul Hsu and James Glass. 2006. Style & topic language model adaptation using hmm-lda. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 373–381.

Dieuwke Hupkes, Anand Singh, Kris Korrel, German Kruszewski, and Elia Bruni. 2018. Learning compositionally through attentive guidance. *arXiv preprint arXiv:1805.09657*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 1681–1691.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Jaap Jumelet, Willem Zuidema, and Dieuwke Hupkes. 2019. Analysing neural language models: Contextual decomposition reveals default reasoning in number and gender assignment. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 1–11.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Najoung Kim and Tal Linzen. 2020. Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105.

Najoung Kim, Roma Patelφ, Adam Poliak, Alex Wang, Patrick Xia, R Thomas McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, et al. 2019a. Probing what different nlp tasks teach machines about function word comprehension. *NAACL HLT 2019*, page 235.

Yoon Kim, Alexander M Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019b. Unsupervised recurrent neural network grammars. 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 1105–1117.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Walter Kintsch. 2001. Predication. *Cognitive science*, 25(2):173–202.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 3294–3302.

Josef Klafka and Allyson Ettinger. 2020. Spying on your neighbors: Fine-grained probing of contextual embeddings for information about surrounding words. *arXiv preprint arXiv:2005.01810*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning-Volume 32*, pages II–1188.

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

Kenton Lee, Shimi Salant, Tom Kwiatkowski, Ankur Parikh, Dipanjan Das, and Jonathan Berant. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Ying Lin, Liyuan Liu, Heng Ji, Dong Yu, and Jiawei Han. 2019. Reliability-aware dynamic feature composition for name tagging. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 165–174.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

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.

Adam Liška, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? searching for a compositional rnn in a haystack. *arXiv preprint arXiv:1802.06467*.

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*.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014b. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223.

Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. 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 622–628.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6297–6308.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

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.

Alessio Miaschi, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2020. Linguistic profiling of a neural language model. *arXiv preprint arXiv:2010.01869*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

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.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, pages 236–244.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

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. *arXiv preprint arXiv:2010.02616*.

David Mrva and Philip C Woodland. 2006. Unsupervised language model adaptation for mandarin broadcast conversation transcription. In *Ninth International Conference on Spoken Language Processing*.

W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *International Conference on Learning Representations*.

Navnita Nandakumar, Timothy Baldwin, and Bahar Salehi. 2019. How well do embedding models capture non-compositionality? a view from multiword expressions. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 27–34.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642.

Prasanna Parthasarathi, Sharan Narang, and Arvind Neelakantan. 2020. On task-level dialogue composition of generative transformer model. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 41–47.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430.

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.

Laura Perez-Mayos, Roberto Carlini, Miguel Ballesteros, and Leo Wanner. 2021. On the evolution of syntactic information encoded by bert's contextualized representations.

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

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

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *NAACL HLT 2018*, page 180.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.*

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, 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.

Sahana Ramnath, Preksha Nema, Deep Sahni, and Mitesh M Khapra. 2020. Towards interpreting bert for reading comprehension based qa. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3236–3242.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018a. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018b. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Xin Rong. 2014. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Shimi Salant and Jonathan Berant. 2018. Contextualized word representations for reading comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 554–559.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Naomi Saphra and Adam Lopez. 2020. Lstms compose (and learn) bottom-up.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. 2017. Neural language modeling by jointly learning syntax and lexicon. *arXiv preprint arXiv:1711.02013*.

Weijia Shi, Muhao Chen, Pei Zhou, and Kai-Wei Chang. 2019. Retrofitting contextualized word embeddings with paraphrases. 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 1198–1203.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Ieva Staliūnaitė and Ignacio Iacobacci. 2020. Compositional and lexical semantics in roberta, bert and distilbert: A case study on coqa. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7046–7056.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827.

Sanjay Subramanian, Sameer Singh, and Matt Gardner. 2019. Analyzing compositionality in visual question answering. *ViGIL@ NeurIPS*, 7.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328.

Yik-Cheung Tam and Tanja Schultz. 2005. Dynamic language model adaptation using variational bayes inference. In *Ninth European Conference on Speech Communication and Technology*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019*.

Shubham Toshniwal, Haoyue Shi, Bowen Shi, Lingyu Gao, Karen Livescu, and Kevin Gimpel. 2020. A cross-task analysis of text span representations. *arXiv preprint arXiv:2006.03866*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. 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.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019b. Allennlp interpret: A framework for explaining predictions of nlp models. *EMNLP-IJCNLP 2019*, page 7.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32.

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.

Chengyu Wang, Minghui Qiu, Jun Huang, and Xiaofeng He. 2020. Meta fine-tuning neural language models for multi-domain text mining. *arXiv preprint arXiv:2003.13003*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019a. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Chih-Kuan Yeh, Joon Sik Kim, Ian EH Yen, and Pradeep Ravikumar. 2018. Representer point selection for explaining deep neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9311–9321.

Da Yin, Tao Meng, and Kai-Wei Chang. 2020. Sentibert: A transferable transformer-based architecture for compositional sentiment semantics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3695–3706.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. Paws: Paraphrase adversaries from word scrambling. 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 1298–1308.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019c. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. 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 629–634.