A Survey on Contextual Embeddings

Qi Liu[‡], **Matt J. Kusner**[†*], **Phil Blunsom**[‡◇],
[‡]University of Oxford [◇]DeepMind
[†]University College London [*]The Alan Turing Institute
[‡]{firstname.lastname}@cs.ox.ac.uk
[†]m.kusner@ucl.ac.uk

Abstract

Contextual embeddings, such as ELMo and BERT, move beyond global word representations like Word2Vec and achieve groundbreaking performance on a wide range of natural language processing tasks. Contextual embeddings assign each word a representation based on its context, thereby capturing uses of words across varied contexts and encoding knowledge that transfers across languages. In this survey, we review existing contextual embedding models, cross-lingual polyglot pre-training, the application of contextual embeddings in downstream tasks, model compression, and model analyses.

1 Introduction

Distributional word representations (Turian et al., 2010; Mikolov et al., 2013; Pennington et al., 2014) trained in an unsupervised manner on large-scale corpora are widely used in modern natural language processing systems. However, these approaches only obtain a single global representation for each word, ignoring their context. Different from traditional word representations, contextual embeddings move beyond word-level semantics in that each token is associated with a representation that is a function of the entire input sequence. These context-dependent representations can capture many syntactic and semantic properties of words under diverse linguistic contexts. Previous work (Peters et al., 2018; Devlin et al., 2018; Yang et al., 2019; Raffel et al., 2019) has shown that contextual embeddings pre-trained on large-scale unlabelled corpora achieve state-of-the-art performance on a wide range of natural language processing tasks, such as text classification, question answering and text summarization. Further analyses (Liu et al., 2019a; Hewitt and Liang, 2019; Hewitt and Manning, 2019; Tenney et al., 2019a) demonstrate that contextual embeddings are capable of learning useful and transferable representations across languages.

The rest of the survey is organized as follows. In Section 2, we define the concept of contextual embeddings. In Section 3, we introduce existing methods for obtaining contextual embeddings. In Section 4, we present the pre-training methods of contextual embeddings on multi-lingual corpora. In Section 5, we describe methods for applying pre-trained contextual embeddings in downstream tasks. In Section 6, we detail model compression methods. In Section 7, we survey analyses that have aimed to identify the linguistic knowledge learned by contextual embeddings. We conclude the survey by highlighting some challenges for future research in Section 8.

2 Token Embeddings

Consider a text corpus that is represented as a sequence $\mathcal{S}$ of tokens, $(t_1, t_2, ..., t_N)$. Distributed representations of words (Harris, 1954; Bengio et al., 2003) associate each token t_i with a dense feature vector $\mathbf{h}_{t_i}$. Traditional *word embedding* techniques aim to learn a global word embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is the number of dimensions. Specifically, each row $\mathbf{e}_i$ of $\mathbf{E}$ corresponds to the global embedding of word type i in the vocabulary V. Well-known models for learning word embeddings include Word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014). On the other hand, methods that learn *contextual embeddings* associate each token t_i with a representation that is a function of the entire input sequence $\mathcal{S}$, i.e. $\mathbf{h}_{t_i} = f(\mathbf{e}_{t_1}, \mathbf{e}_{t_2}, ..., \mathbf{e}_{t_N})$, where each input token t_j is usually mapped to its non-contextualized representation $\mathbf{e}_{t_j}$ first, before applying an aggregation function f. These context-

dependent representations are better suited to capture sequence-level semantics (e.g. polysemy) than non-contextual word embeddings. There are many model architectures for f, which we review here. We begin by describing pre-training methods for learning contextual embeddings that can be used in downstream tasks.

3 Pre-training Methods for Contextual Embeddings

In large part, pre-training contextual embeddings can be divided into either unsupervised methods (e.g. language modelling and its variants) or supervised methods (e.g. machine translation and natural language inference).

3.1 Unsupervised Pre-training via Language Modeling

The prototypical way to learn distributed token embeddings is via language modelling. A language model is a probability distribution over a sequence of tokens. Given a sequence of N tokens, $(t_1, t_2, ..., t_N)$, a language model factorizes the probability of the sequence as:

$$p(t_1, t_2, ..., t_N) = \prod_{i=1}^{N} p(t_i | t_1, t_2, ..., t_{i-1}). \quad (1)$$

Language modelling uses maximum likelihood estimation (MLE), often penalized with regularization terms, to estimate model parameters. A left-to-right language model takes the left context, $t_1, t_2, ..., t_{i-1}$, of t_i into account for estimating the conditional probability. Language models are usually trained using large-scale unlabelled corpora. The conditional probabilities are most commonly learned using neural networks (Bengio et al., 2003), and the learned representations have been proven to be transferable to downstream natural language understanding tasks (Dai and Le, 2015; Ramachandran et al., 2016).

Precursor Models. Dai and Le (2015) is the first work we are aware of that uses language modelling together with a sequence autoencoder to improve sequence learning with recurrent networks. Thus, it can be thought of as a precursor to modern contextual embedding methods. Pre-trained on the datasets IMDB, Rotten Tomatoes, 20 Newsgroups, and DBpedia, the model is then fine-tuned on sentiment analysis and text classification tasks, achieving strong performance compared to randomly-initialized models.

Ramachandran et al. (2016) extends Dai and Le (2015) by proposing a pre-training method to improve the accuracy of sequence to sequence (seq2seq) models. The encoder and decoder of the seq2seq model is initialized with the pre-trained weights of two language models. These language models are separately trained on either the News Crawl English or German corpora for machine translation, while both are initialized with the language model trained with the English Gigaword corpus for abstractive summarization. These pre-trained models are fine-tuned on the WMT English $\rightarrow$ German task and the CNN/Daily Mail corpus, respectively, achieving better results over baselines without pre-training.

The work in the following sections improves over Dai and Le (2015) and Ramachandran et al. (2016) with new architectures (e.g. Transformer), larger datasets, and new pre-training objectives. A summary of the models and the pre-training objectives is shown in Table 1 and 2.

ELMo. The ELMo model (Peters et al., 2018) generalizes traditional word embeddings by extracting context-dependent representations from a bidirectional language model. A forward L-layer LSTM and a backward L-layer LSTM are applied to encode the left and right contexts, respectively. At each layer j, the contextualized representations are the concatenation of the left-to-right and right-to-left representations, obtaining N hidden representations, $(\mathbf{h}_{1,j}, \mathbf{h}_{2,j}, ..., \mathbf{h}_{N,j})$, for a sequence of length N.

To use ELMo in downstream tasks, the $(L+1)$-layer representations (including the global word embedding) for each token k are aggregated as:

$$\text{ELMo}_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}, \quad (2)$$

where $\mathbf{s}^{task}$ are layer-wise weights normalized by the softmax used to linearly combine the $(L+1)$-layer representations of the token k and γ^{task} is a task-specific constant.

Given a pre-trained ELMo, it is straightforward to incorporate it into a task-specific architecture for improving the performance. As most supervised models use global word representations $\mathbf{x}_k$ in their lowest layers, these representations can be concatenated with their corresponding context-dependent representations ELMo_k^{task}, obtaining

Method	Architecture	Encoder	Decoder	Objective	Dataset
ELMo	LSTM	✗	✓	LM	1B Word Benchmark
GPT	Transformer	✗	✓	LM	BookCorpus
GPT2	Transformer	✗	✓	LM	Web pages starting from Reddit
BERT	Transformer	✓	✗	MLM & NSP	BookCorpus & Wiki
RoBERTa	Transformer	✓	✗	MLM	BookCorpus, Wiki, CC-News, OpenWebText, Stories
ALBERT	Transformer	✓	✗	MLM & SOP	Same as RoBERTa and XLNet
UniLM	Transformer	✓	✗	LM, MLM, seq2seq LM	Same as BERT
ELECTRA	Transformer	✓	✗	Discriminator (o/r)	Same as XLNet
XLNet	Transformer	✗	✓	PLM	BookCorpus, Wiki, Giga5, ClueWeb, Common Crawl
XLM	Transformer	✓	✓	CLM, MLM, TLM	Wiki, parellel corpora (e.g. MultiUN)
MASS	Transformer	✓	✓	Span Mask	WMT News Crawl
T5	Transformer	✓	✓	Text Infilling	Colossal Clean Crawled Corpus
BART	Transformer	✓	✓	Text Infilling & Sent Shuffling	Same as RoBERTa

Table 1: A comparison of popular pre-trained models.

Objective	Inputs	Targets
LM	[START]	I am happy to join with you today
MLM	I am [MASK] to join with you [MASK]	happy today
NSP	Sent1 [SEP] Next Sent or Sent1 [SEP] Random Sent	Next Sent/Random Sent
SOP	Sent1 [SEP] Sent2 or Sent2 [SEP] Sent1	in order/reversed
Discriminator (o/r)	I am thrilled to study with you today	o o r o r o o o
PLM	happy join with	today am I to you
seq2seq LM	I am happy to	join with you today
Span Mask	I am [MASK] [MASK] [MASK] with you today	happy to join
Text Infilling	I am [MASK] with you today	happy to join
Sent Shuffling	today you am I join with happy to	I am happy to join with you today
TLM	How [MASK] you [SEP] [MASK] vas-tu	are Comment

Table 2: Pre-training objectives and their input-output formats.

$[\mathbf{x}_k; \text{ELMo}_k^{task}]$, before feeding them to higher layers.

The effectiveness of ELMo is evaluated on six NLP problems, including question answering, textual entailment and sentiment analysis.

GPT, GPT2, and Grover. GPT (Radford et al., 2018) adopts a two-stage learning paradigm: (a) unsupervised pre-training using a language modelling objective and (b) supervised fine-tuning. The goal is to learn universal representations transferable to a wide range of downstream tasks. To this end, GPT uses the BookCorpus dataset (Zhu et al., 2015), which contains more than 7,000 books from various genres, for training the language model. The Transformer architecture (Vaswani et al., 2017) is used to implement the language model, which has been shown to better capture global dependencies from the inputs compared to its alternatives, e.g. recurrent networks, and perform strongly on a range of sequence learning tasks, such as machine translation (Vaswani et al., 2017) and document generation (Liu et al., 2018). To use GPT on inputs with multiple sequences during fine-tuning, GPT applies task-specific input adaptations motivated by traversal-style approaches (Rocktäschel et al., 2015). These approaches pre-process each text input as a single contiguous sequence of tokens through special tokens including [START] (the start of a sequence), [DELIM] (delimiting two sequences from the text input) and [EXTRACT] (the end of a sequence). GPT outperforms task-specific architectures in 9 out of 12 tasks studied with a pre-trained Transformer.

GPT2 (Radford et al., 2019) mainly follows the architecture of GPT and trains a language model on a dataset as large and diverse as possible to learn from varied domains and contexts. To do so, Radford et al. (2019) create a new dataset of millions of web pages named WebText, by scraping outbound links from Reddit. The authors argue that a language model trained on large-scale unlabelled corpora begins to learn some common supervised NLP tasks, such as question answering, machine translation and summarization, without any explicit supervision signal. To validate this, GPT2 is tested on ten datasets (e.g. Children's Book Test (Hill et al., 2015), LAMBADA (Paperno et al., 2016) and CoQA (Reddy et al.,

2019)) in a zero-shot setting. GPT2 performs strongly on some tasks. For instance, when conditioned on a document and questions, GPT2 reaches an F1-score of 55 on the CoQA dataset without using any labelled training data. This matches or outperforms the performance of 3 out of 4 baseline systems. As GPT2 divides texts into bytes and uses BPE (Sennrich et al., 2016) to build up its vocabulary (instead of using characters or words, as in previous work), it is unclear if the improved performance comes from the model or the new input representation.

Grover (Zellers et al., 2019) creates a news dataset, RealNews, from Common Crawl and pretrains a language model for generating realistic-looking fake news that is conditioned on metadata including domains, dates, authors and headlines. They further study discriminators that can be used to detect fake news. The best defense against Grover turns out to be Grover itself, which sheds light on the importance of releasing trained models for detecting fake news.

BERT. ELMo (Peters et al., 2018) concatenates representations from the forward and backward LSTMs without considering the interactions between the left and right contexts. GPT (Radford et al., 2018) and GPT2 (Radford et al., 2019) use a left-to-right decoder, where every token can only attend to its left context. These architectures are sub-optimal for sentence-level tasks, e.g. named entity recognition and sentiment analysis, as it is crucial to incorporate contexts from both directions.

BERT proposes a masked language modelling (MLM) objective, where some of the tokens of a input sequence are randomly masked, and the objective is to predict these masked positions taking the corrupted sequence as input. BERT applies a Transformer encoder to attend to bi-directional contexts during pre-training. In addition, BERT uses a next-sentence-prediction (NSP) objective. Given two input sentences, NSP predicts whether the second sentence is the actual next sentence of the first sentence. The NSP objective aims to improve the tasks, such as question answering and natural language inference, which require reasoning over sentence pairs.

Similar to GPT, BERT uses special tokens to obtain a single contiguous sequence for each input sequence. Specifically, the first token is always a special classification token [CLS], and sentence pairs are separated using a special token [SEP]. BERT adopts a pre-training followed by fine-tuning scheme. The final hidden state of [CLS] is used for sentence-level tasks and the final hidden state of each token is used for token-level tasks. BERT obtains new state-of-the-art results on eleven natural language processing tasks, e.g. improving the GLUE (Wang et al., 2018) score to 80.5%.

Similar to GPT2, it is unclear exactly why BERT improves over prior work as it uses different objectives, datasets (Wikipedia and BookCorpus) and architectures compared to previous methods. For partial insight on this, we refer the readers to (Raffel et al., 2019) for a controlled comparison between unidirectional and bidirectional models, traditional language modelling and masked language modelling using the same datasets.

BERT variants. Recent work further studies and improves the objective and architecture of BERT.

Instead of randomly masking tokens, ERNIE (Zhang et al., 2019) incorporates knowledge masking strategies, including entity-level masking and phrase-level masking. SpanBERT (Joshi et al., 2019) generalizes this idea to mask random spans, without referring to external knowledge. StructBERT (Wang et al., 2019b) proposes a word structural objective that randomly permutes the order of 3-grams for reconstruction and a sentence structural objective that predicts the order of two consecutive segments.

RoBERTa (Liu et al., 2019c) makes a few changes to the released BERT model and achieves substantial improvements. The changes include: (1) Training the model longer with larger batches and more data; (2) Removing the NSP objective; (3) Training on longer sequences; (4) Dynamically changing the masked positions during pre-training.

ALBERT (Lan et al., 2019) proposes two parameter-reduction techniques (factorized embedding parameterization and cross-layer parameter sharing) to lower memory consumption and speed up training. Furthermore, ALBERT argues that the NSP objective lacks difficulty, as the negative examples are created by pairing segments from different documents, this mixes topic prediction and coherence prediction into a single task. ALBERT instead uses a sentence-order prediction (SOP) objective. SOP obtains positive examples

by taking out two consecutive segments and negative examples by reversing the order of two consecutive segments from the same document.

XLNet. The XLNet model (Yang et al., 2019) identifies two weaknesses of BERT:

1. BERT assumes conditional independence of corrupted tokens. For instance, to model the probability $p(t_2 = \text{cat}, t_6 = \text{mat}|t_1 = \text{The}, t_2 = \text{[MASK]}, t_3 = \text{sat}, t_4 = \text{on}, t_5 = \text{the}, t_6 = \text{[MASK]})$, BERT factorizes it as $p(t_2 = \text{cat}|...)p(t_6 = \text{mat}|...)$, where t_2 and t_6 are assumed to be conditionally independent.

2. The symbols such as [MASK] are introduced by BERT during pre-training, yet they never occur in real data, resulting in a discrepancy between pre-training and fine-tuning.

XLNet proposes a new auto-regressive method based on permutation language modelling (PLM) (Uria et al., 2016) without introducing any new symbols. The MLE objective for it is calculated as:

$$\max_{\theta} \mathbb{E}_{\mathbf{z} \in Z_N} \left[\sum_{j=1}^{N} \log p_\theta(t_{z_j} | t_{z_1}, t_{z_2}, ..., t_{z_{j-1}}) \right]. \tag{3}$$

For each sequence, XLNet samples a permutation order $\mathbf{z} = [z_1, z_2, ..., z_N]$ from the set of all permutations Z_N, where $|Z_N| = N!$. The probability of the sequence is factorized according to $\mathbf{z}$, where the z_j-th token t_{z_j} is conditioned on all the previous tokens $t_{z_1}, t_{z_2}, ..., t_{z_j}$ according to the permutation order $\mathbf{z}$.

XLNet further adopts two-stream self-attention and Transformer-XL (Dai et al., 2019) to take into account the target positions z_j and learn long-range dependencies, respectively.

As the cardinality of Z_N is factorial, naive optimization would be challenging. Thus, XLNet conditions on part of the input and generates the rest of the input to reduce the scale of the search space:

$$\max_{\theta} \mathbb{E}_{\mathbf{z} \in Z_N} \left[\sum_{j=c+1}^{N} \log p_\theta(t_{z_j} | t_{z_1}, t_{z_2}, ..., t_{z_{j-1}}) \right], \tag{4}$$

where c is the cutting point of the sequence. However, it is tricky to compare XLNet directly with BERT due to the multiple changes in loss and architecture.[1]

UniLM. UniLM (Dong et al., 2019) adopts three objectives: (a) language modelling, (b) masked language modelling, and (c) sequence-to-sequence language modelling (seq2seq LM), for pre-training a Transformer network. To implement three objectives in a single network, UniLM utilizes specific self-attention masks to control what context the prediction conditions on. For example, MLM can attend to its bidirectional contexts, while seq2seq LM can attend to bidirectional contexts for source sequences and left contexts only for target sequences.

ELECTRA. Compared to BERT, ELECTRA (Clark et al., 2019) proposes a more effective pre-training method. Instead of corrupting some positions of inputs with [MASK], ELECTRA replaces some tokens of the inputs with their plausible alternatives sampled from a small generator network. ELECTRA trains a discriminator to predict whether each token in the corrupted input was replaced by the generator or not. The pre-trained discriminator can then be used in downstream tasks for fine-tuning, improving upon the pre-trained representation learned by the generator.

MASS. Although BERT achieves state-of-the-art performance for many natural language understanding tasks, BERT cannot be easily used for natural language generation. MASS (Song et al., 2019) uses masked sequences to pre-train sequence-to-sequence models. More specifically, MASS adopts an encoder-decoder framework and extends the MLM objective. The encoder takes as input a sequence where consecutive tokens are masked and the decoder predicts these masked consecutive tokens autoregressively. MASS achieves significant improvements over baselines without pre-training or with other pre-training methods on a variety of zero/low-resource language generation tasks, including neural machine translation, text summarization and conversational response generation.

T5. Raffel et al. (2019) propose T5 (Text-to-Text Transfer Transformer), unifying natural language understanding and generation by converting the data into a text-to-text format and applying a encoder-decoder framework.

[1] We note that RoBERTa, which makes much smaller changes to BERT is able to outperform XLNet. Future study needs to be done to understand the precise advantages of XLNet's modifications to BERT.

T5 introduces a new pre-training dataset, Colossal Clean Crawled Corpus by cleaning the web pages from Common Crawl. T5 also systematically compares previous methods in terms of pre-training objectives, architectures, pre-training datasets, and transfer approaches. T5 adopts a text infilling objective (where spans of text are replaced with a single mask token), longer training, multi-task pre-training on GLUE or SuperGLUE, fine-tuning on each individual GLUE and SuperGLUE tasks, and beam search.

For fine-tuning, to convert the input data into a text-to-text framework, T5 utilizes the token vocabulary of the decoder as the prediction labels. For example, the tokens "entailment", "contradiction", and "neutral" are used as the labels for natural language inference tasks. For the regression task (e.g. STS-B (Cer et al., 2017)), T5 simply rounds up the scores to the nearest multiple of 0.2 and converts the results to literal string representations (e.g. 2.57 is converted to the string "2.6"). T5 also adds a task-specific prefix to each input sequence to specify its task. For instance, T5 adds the prefix "translate English to German" to each input sequence like "That is good." for English-to-German translation datasets.

BART. The BART model (Lewis et al., 2019) introduces additional noising functions beyond MLM for pre-training sequence-to-sequence models. First, the input sequence is corrupted using an arbitrary noising function. Then, the corrupted input is reconstructed by a Transformer network trained using teacher forcing (Williams and Zipser, 1989). BART evaluates a wide variety of noising functions, including token masking, token deletion, text infilling, document rotation, and sentence shuffling (randomly shuffling the word order of a sentence). The best performance is achieved by using both sentence shuffling and text infilling. BART matches the performance of RoBERTa on GLUE and SQuAD and achieves state-of-the-art performance on a variety of text generation tasks.

3.2 Supervised Objectives

Pre-training on the ImageNet dataset (which has supervision about the objects in images) before fine-tuning on downstream tasks has become the *de facto* standard in the computer vision community. Motivated by the success of supervised pre-training in computer vision, some work (Conneau et al., 2017; McCann et al., 2017; Subramanian et al., 2018) utilizes data-rich tasks in NLP to learn transferable representations.

CoVe (McCann et al., 2017) shows that the representations learned from machine translation are transferable to downstream tasks. CoVe uses a deep LSTM encoder from a sequence-to-sequence model trained for machine translation to obtain contextual embeddings. Empirical results show that augmenting non-contextualized word representations (Mikolov et al., 2013; Pennington et al., 2014) with CoVe embeddings improves performance over a wide variety of common NLP tasks, such as sentiment analysis, question classification, entailment, and question answering. InferSent (Conneau et al., 2017) obtains contextualized representations from a pre-trained natural language inference model on SNLI. Subramanian et al. (2018) use multi-task learning to pre-train a sequence-to-sequence model for obtaining general representations, where the tasks include skip-thought (Kiros et al., 2015), machine translation, constituency parsing, and natural language inference.

4 Cross-lingual Polyglot Pre-training for Contextual Embeddings

Cross-lingual polyglot pre-training aims to learn joint multi-lingual representations, enabling knowledge transfer from data-rich languages like English to data-scarce languages like Romanian. Based on whether joint training and a shared vocabulary are used, we divide previous work into three categories.

Joint training & shared vocabulary. Artetxe and Schwenk (2019) use a BiLSTM encoder-decoder framework with a shared BPE vocabulary for 93 languages. The framework is pre-trained using parallel corpora, including as Europarl and Tanzil. The contextual embeddings from the encoder are used to train classifiers using English corpora for downstream tasks. As the embedding space and the encoder are shared, the resultant classifiers can be transferred to any of the 93 languages without further modification. Experiments show that these classifiers achieve competitive performance on cross-lingual natural language inference, cross-lingual document classification, and parallel corpus mining.

Rosita (Mulcaire et al., 2019) pre-trains a language model using text from different languages,

showing the benefits of polyglot learning on low-resource languages.

Recently, the authors of BERT developed a multi-lingual BERT[2] which is pre-trained using the Wikipedia dump with more than 100 languages.

XLM (Lample and Conneau, 2019) uses three pre-training methods for learning cross-lingual language models: (1) Causal language modelling, where the model is trained to predict $p(t_i|t_1, t_2, ..., t_{i-1})$, (2) Masked language modelling, and (3) Translation language modelling (TLM). Parallel corpora are used, and tokens in both source and target sequences are masked for learning cross-lingual association. XLM performs strongly on cross-lingual classification, unsupervised machine translation, and supervised machine translation. XLM-R (Conneau et al., 2019) scales up XLM by training a Transformer-based masked language model on one hundred languages, using more than two terabytes of filtered CommonCrawl data. XLM-R shows that large-scale multi-lingual pre-training leads to significant performance gains for a wide range of cross-lingual transfer tasks.

Joint training & separate vocabularies. Wu et al. (2019) study the emergence of cross-lingual structures in pre-trained multi-lingual language models. It is found that cross-lingual transfer is possible even when there is no shared vocabulary across the monolingual corpora, and there are universal latent symmetries in the embedding spaces of different languages.

Separate training & separate vocabularies. Artetxe et al. (2019) use a four-step method for obtaining multi-lingual embeddings. Suppose we have the monolingual sequences of two languages L_1 and L_2: (1) Pre-training BERT with the vocabulary of L_1 using L_1's monolingual data. (2) Replacing the vocabulary of L_1 with the vocabulary of L_2 and training new vocabulary embeddings, while freezing the other parameters, using L_2's monolingual data. (3) Fine-tuning the BERT model for a downstream task using labeled data in L_1, while freezing L_1's vocabulary embeddings. (4) Replacing the fine-tuned BERT with L_2's vocabulary embeddings for zero-shot transfer tasks.

[2]https://github.com/google-research/bert/blob/master/multilingual.md

5 Downstream Learning

Once learned, contextual embeddings have demonstrated impressive performance when used downstream on various learning problems. Here we describe the ways in which contextual embeddings are used downstream, the ways in which one can avoid forgetting information in the embeddings during downstream learning, and how they can be specialized to multiple learning tasks.

5.1 Ways to Use Contextual Embeddings Downstream

There are three main ways to use pre-trained contextual embeddings in downstream tasks: (1) Feature-based methods, (2) Fine-tuning methods, and (3) Adapter methods.

Feature-based. One example of a feature-based is the method used by ELMo (Peters et al., 2018). Specifically, as shown in equation 2, ELMo freezes the weights of the pre-trained contextual embedding model and forms a linear combination of its internal representations. The linearly-combined representations are then used as features for task-specific architectures. The benefit of feature-based models is that they can use state-of-the-art handcrafted architectures for specific tasks.

Fine-tuning. Fine-tuning works as follows: starting with the weights of the pre-trained contextual embedding model, fine-tuning makes small adjustments to them in order to specialize them to a specific downstream task. One stream of work applies minimal changes to pre-trained models to take full advantage of their parameters. The most straightforward way is adding linear layers on top of the pre-trained models (Devlin et al., 2018; Lan et al., 2019). Another method (Radford et al., 2019; Raffel et al., 2019) uses universal data formats without introducing new parameters for downstream tasks.

To apply pre-trained models to structurally different tasks, where task-specific architectures are used, as much of the model is initialized with pre-trained weights as possible. For instance, XLM (Lample and Conneau, 2019) applies two pre-trained monolingual language models to initialize the encoder and the decoder for machine translation, respectively, leaving only cross-attention weights randomly initialized.

Adapters. Adapters (Rebuffi et al., 2017; Stickland and Murray, 2019) are small modules

added between layers of pre-trained models to be trained in a multi-task learning setting. The parameters of the pre-trained model are fixed while tuning these adapter modules. Compared to previous work that fine-tunes a separate pre-trained model for each task, a model with shared adapters for all tasks often requires fewer parameters.

5.2 Countering Catastrophic Forgetting

Learning on downstream tasks is prone to overwrite the information from pre-trained models, which is widely known as the catastrophic forgetting (McCloskey and Cohen, 1989; d'Autume et al., 2019). Previous work combats this by (1) Freezing layers, (2) Using adaptive learning rates, and (3) Regularization.

Freezing layers. Motivated by layer-wise training of neural networks (Hinton et al., 2006), training certain layers while freezing others can potentially reduce forgetting during fine-tuning. Different layer-wise tuning schedules have been studied. Long et al. (2015) freeze all layers except the top layer. Felbo et al. (2017) use "chain-thaw", which sequentially unfreezes and fine-tunes a layer at a time. Howard and Ruder (2018) gradually unfreeze all layers one by one from top to bottom. Chronopoulou et al. (2019) apply a three-stage fine-tuning schedule: (a) randomly-initialized parameters are updated for n epochs, (b) the pre-trained parameters (except word embeddings) are then fine-tuned, (c) at last, all parameters are fine-tuned.

Adaptive learning rates. Another method to mitigate catastrophic forgetting is by using adaptive learning rates. As it is believed that the lower layers of pre-trained models tend to capture general language knowledge (Tenney et al., 2019a), Howard and Ruder (2018) use lower learning rates for lower layers when fine-tuning.

Regularization. Regularization limits the fine-tuned parameters to be close to the pre-trained parameters. Wiese et al. (2017) minimize the Euclidean distance between the fine-tuned parameters and pre-trained parameters. Kirkpatrick et al. (2017) use the Fisher information matrix to protect the weights that are identified as essential for pre-trained models.

5.3 Multi-task Fine-tuning

Multi-task learning on downstream tasks (Liu et al., 2019b; Wang et al., 2019a; Jozefowicz et al., 2016) obtains general representations across tasks and achieves strong performance on each individual task.

MT-DNN (Liu et al., 2019b) fine-tunes BERT on all the GLUE tasks, improving the GLUE benchmark to 82.7%. MT-DNN also demonstrates that the representations from multi-task learning obtain better performance on domain adaptation compared to BERT.

Wang et al. (2019a) investigate further, non-GLUE tasks, such as skip-thought and Reddit response generation, for multi-task learning.

T5 (Raffel et al., 2019) studies various settings of multi-task learning and finds that using multi-task learning before fine-tuning on each task performs the best.

6 Model Compression

As many pre-trained language models have a prohibitive memory footprint and latency, it is a challenging task to deploy them in resource-constrained environments. To address this, model compression (Cheng et al., 2017), which has gained popularity in recent years for shrinking large neural networks, has been investigated for compressing contextual embedding models. Work on compressing language models utilizes (1) Low-rank approximation, (2) Knowledge distillation, and (3) Weight quantization, to make them usable in embedded systems and edge devices.

Low rank approximation. Methods that learn low rank approximations seek to compress the full-rank model weight matrices into low-rank matrices, thereby reducing the effective number of model parameters. As the embedding matrices usually account for a large portion of model parameters (e.g. 21% for $\text{BERT}_{\text{Base}}$), ALBERT (Lan et al., 2019) approximates the embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d}$ as the product of two smaller matrices, $\mathbf{E}_1 \in \mathbb{R}^{V \times d'}$ and $\mathbf{E}_2 \in \mathbb{R}^{d' \times d}$, where $d' \ll d$.

Knowledge distillation. A method called 'knowledge distillation' was proposed by Hinton et al. (2015), where the 'knowledge' encoded in a teacher network is transferred to a student network. Hinton et al. (2015) use the soft target probabilities, output by the teacher network, to

train the student network using the cross-entropy loss. The student network is smaller than the teacher network, resulting in a more lightweight model that nears the accuracy of the heavyweight teacher network. Tang et al. (2019) distill the knowledge from BERT into a single-layer BiLSTM, obtaining performance comparable to ELMo with roughly 100 times fewer parameters. DistilBERT (Sanh et al., 2019) uses MLM, distillation loss (Hinton et al., 2015), and cosine similarity between the embedding matrices of the teacher and student networks to train a smaller BERT model. BERT-PKD (Sun et al., 2019) uses a student BERT model with fewer layers compared to $\text{BERT}_{\text{Base}}$ or $\text{BERT}_{\text{Large}}$ and proposes two ways (learning from the last k layers and learning from every k layers) to map the layers of the student to the layers of $\text{BERT}_{\text{Base}}$ or $\text{BERT}_{\text{Large}}$. The hidden states of the student are kept close to the hidden states of the teacher from corresponding layers using a Euclidean distance regularizer. TinyBERT (Jiao et al., 2019) introduces a two-stage learning framework, where distillation is performed at both the pre-training and the fine-tuning stages.

Weight quantization. Quantization methods focus on mapping weight parameters to low-precision integers and floating-point numbers. Q-BERT (Shen et al., 2019) proposes a group-wise quantization scheme, where the parameters are divided into groups based on attention heads, and uses a Hessian-based, mixed-precision method to compress the model.

7 Analyzing Contextual Embeddings

While contextual embedding methods have impressive performance on a variety of natural language tasks, it is often unclear exactly why they work so well. To study this, work so far has used (1) Probe classifiers, and (2) Visualization.

Probe classifiers. A large body of work studies contextual embeddings using *probes*. These are constrained classifiers designed to explore whether syntactic and semantic information is encoded in these representations or not.

Liu et al. (2019a) design a series of token labelling, segmentation, and pairwise relation tasks for studying the effectiveness of contextual embeddings. Contextual embeddings achieve competitive results compared to the state-of-the-art models on most tasks, yet fail on some fine-grained linguistic tasks (e.g. conjunct identification).

Hewitt and Manning (2019) propose a structural probe for finding syntax in contextual embeddings. The model attempts to learn a linear transformation under which the L2 distances between tokens encode the distances between these tokens in syntactic parsing trees like dependency trees.

Tenney et al. (2019a) find that BERT rediscovers the traditional NLP pipeline in an interpretable and localizable way. Specifically, it is capable at POS tagging, parsing, NER, semantic roles, and coreference, and these are learned in order.

Jawahar et al. (2019) use ten sentence-level probing tasks (e.g. SentLen, TreeDepth) and find that BERT captures phrase-level information in earlier layers and long-distance dependency information in deeper layers.

Visualization. Another body of work uses visualization to analyze attention and fine-tuning procedures, among others.

Hao et al. (2019) visualize loss landscapes and optimization trajectories when fine-tuning BERT. The visualizations show that BERT reaches a good initial point during pre-training for downstream tasks, which can lead to better optima compared to randomly-initialized models.

Kovaleva et al. (2019) visualize the attention heads of BERT, discovering a limited set of attention patterns across different heads. This leads to the fact that the heads of BERT are highly redundant. After manually disabling certain attention heads, better performance is obtained compared to the fine-tuned BERT models that use the full set of attention heads.

Coenen et al. (2019) visualize and analyze the geometry of BERT embeddings, finding that BERT distinguishes word senses at a very fine-grained level. These word senses are also found to be encoded in a relatively low-dimensional subspace.

8 Current Challenges

There are many key challenges that, if solved, would improve future contextual embeddings.

Better pre-training objectives. BERT designed MLM to take advantage of bi-directional information during pre-training. It remains unclear whether there are pre-training objectives that are simultaneously more efficient and effective. Some

recent work focuses on designing new training methods (Clark et al., 2019), noise combination techniques, (Lewis et al., 2019) and multi-task learning approaches (Wang et al., 2019a).

Understanding the knowledge encoded in pre-trained models. As described above, a range of methods (Tenney et al., 2019a,b; Hewitt and Manning, 2019; Liu et al., 2019a) have been proposed to explore the effectiveness of pre-trained models via probes. Yet, controlled experiments are still lacking to understand whether the representations actually encode linguistic knowledge or the probes happen to learn to perform well on these linguistic tasks because the data they use is so high-dimensional (Hewitt and Liang, 2019). Hewitt and Liang (2019) devise control tasks, where a good probe is one that performs well on linguistic tasks, and badly on control tasks. They find that most existing probes fail to satisfy this condition. Indeed, most probes use shallow classifiers, which may not be able to extract the relevant information from contextual representations. New probes or better methods for understanding contextual representations are needed.

Model robustness. Concerns about the vulnerability of models to attack are growing when deploying NLP models into production. Wallace et al. (2019) show that universal adversarial triggers that cause significant performance deterioration of pre-trained models can be found. Additionally, concerns of abusing pre-trained models (e.g. generating fake news) have arisen[3]. Better methods for increasing model robustness are highly needed.

Controlled generation of sequences. Pre-trained language models (Radford et al., 2018, 2019) are able to generate realistic-looking text sequences. Yet it is hard to adapt these models to generate domain-specific sequences (Keskar et al., 2019) or to agree with common human knowledge (Zellers et al., 2019). As a result, we advocate research on more fine-grained control over sequence generation.

Acknowledgements

We thank Douwe Kiela, Jiatao Gu, Yi Tay, Xiaodong Liu, Ziyang Wang and Jake Zhao for their comments and discussions on this manuscript.

[3]https://www.theverge.com/2019/11/7/20953040/openai-text-generation-ai-gpt-2-full-model-release-1-5b-parameters

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2017. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*.

Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. 2019. An embarrassingly simple approach for transfer learning from pretrained language models. *arXiv preprint arXiv:1902.10547*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*.

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, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan

Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *arXiv preprint arXiv:1906.01076*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of bert. *arXiv preprint arXiv:1908.05620*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Ganesh Jawahar, Benoît Sagot, Djamé Seddah, Samuel Unicomb, Gerardo Iñiguez, Márton Karsai, Yannick Léo, Márton Karsai, Carlos Sarraute, Éric Fleury, et al. 2019. What does bert learn about the structure of language? In *57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. 2019a. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

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

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. 2015. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Phoebe Mulcaire, Jungo Kasai, and Noah Smith. 2019. Polyglot contextual representations improve crosslingual transfer. *arXiv preprint arXiv:1902.09697*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

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

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

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).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Prajit Ramachandran, Peter J Liu, and Quoc V Le. 2016. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

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*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2019. Q-bert: Hessian based ultra low precision quantization of bert. *arXiv preprint arXiv:1909.05840*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. *arXiv preprint arXiv:1902.02671*.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. 2016. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220.

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.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for nlp. *arXiv preprint arXiv:1908.07125*.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Liwei Peng, and Luo Si. 2019b. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. 2017. Neural domain adaptation for biomedical question answering. *arXiv preprint arXiv:1706.03610*.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Emerging cross-lingual structure in pretrained language models. *arXiv preprint arXiv:1911.01464*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *arXiv preprint arXiv:1905.12616*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.