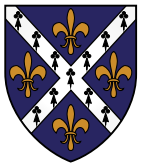


Better Conditioning on Context for Natural Language Processing

Qi Liu

St Hugh's College



University of Oxford

A thesis presented for the degree of

Doctor of Philosophy

Trinity 2022

I dedicate this thesis to my parents, Jingtian and Shuhua.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors, Prof. Phil Blunsom and Prof. Matt Kusner. I have benefited immeasurably from their guidance throughout my DPhil study. Prof. Phil Blunsom always impresses me with his repository of knowledge about natural language processing and machine learning. Prof. Phil Blunsom has the superpower to pinpoint the key ideas behind everything quickly. I especially thank Prof. Phil Blunsom for teaching me every aspect of research from formulating research ideas to presentation skills. I learned a lot from Prof. Matt Kusner's meticulous attitude towards research. His door has been always open to me whenever I have something to discuss. Prof. Matt Kusner has introduced me to the wonderful world of causal inference and data compression. He is also a kind, strict and tireless editor during paper submission.

Second, I want to thank the DeepMind language team for hosting my internships from 2020 to 2021. I especially thank Lei Yu, Laura Rimell and Dani Yogatama for supervising my internship projects. Many thanks to Chris Dyer, Sebastian Ruder, Cyprien de Masson d'Autume and Lingpeng Kong for their helpful discussions.

Next, I would like to thank my friends and collaborators, Yishu Miao, Dingmin Wang, Hanchen Wang, Hannan Xu, Ziyang Wang, Zhenguang Liu, Jean Kaddour and Linfeng Song, who gave me endless support during the pandemic. I thank my examiners Prof. Thomas Lukasiewicz and Dr. Angeliki Lazaridou for proofreading my thesis. I also thank my DPhil transfer and confirmation examiners Prof. Yarin Gal and Prof. Andrew Markham.

Last, I want to thank my parents (Jingtian and Shuhua), my sister (Xue) and my girlfriend (Lynn) for their tremendous support and constant love.

Abstract

Learning distributed representations of natural language has become a common practice for Natural Language Processing (NLP). Non-contextual embeddings map each token in the vocabulary to a low-dimensional real-valued vector. Although these representations perform competitively on word-level tasks, e.g. measuring word similarities, they are context-independent and fail to distinguish the semantics of words in different contexts. This gives rise to contextual embeddings, where each token is associated with a representation that is a function of the entire input sequence. Contextual embeddings are shown to better capture complex characteristics of word use and model polysemous words.

In this thesis, we follow up the line of research on contextual embeddings and explore better methods of conditioning on context for natural language processing. Specifically, we study incorporating contextual information for Natural Language Generation (NLG). We explore three different sources of context including local context of each input sequence (Chapter 3), global context from training data (Chapter 4) and external context from the Web (Chapter 5). In Chapter 3, we utilize the local context of each input sequence to better model word polysemy. We introduce a technique called translation-counterfactual word replacement, which leverages both local context and alignment for augmenting machine translation data. In Chapter 4, we retrieve relevant global context about a particular entity from training data. A memory-augmented approach is presented to condition an autoregressive language model on global context for more coherent and logical generation. In Chapter 5, we pre-train a noisy channel model with external context (e.g. Reddit posts from the Web) for task-oriented dialogue and show that the noisy channel model decodes better responses compared to direct decoding.

Finally, Chapter 6 summarizes our findings and directions for future work.

Table of Contents

1 Introduction **2**

1.1 Formalism of Contextual Embeddings 3

1.2 Contributions 4

1.3 Thesis Outline 7

2 Contextual Embeddings **9**

2.1 Transformer 9

2.2 Objectives for Learning Contextual Embeddings 11

2.2.1 Autoregressive Language Models 12

2.2.2 Seq2seq Language Models 15

2.2.3 Masked Language Models 17

2.2.4 Text Infilling 19

2.3 Linguistic Analysis 21

2.4 Summary 23

3 Utilizing Local Context **24**

3.1 Introduction 24

3.2 Related Work 27

3.2.1 Data Augmentation for NMT 27

3.3 Background 28

3.4 Method 30

3.4.1 Translation-Counterfactual Word Replacement 31

3.4.2 Data Augmentation 35

3.4.3 Training Language Models 35

3.5 Experiments 37

3.5.1 Language Model Details 37

3.5.2 NMT Model Details . . . 39
3.5.3 Sensitivity Study . . . 39
3.5.4 Ablation Study . . . 41
3.5.5 Translation Result . . . 42
3.5.6 Backtranslation Result . . . 44
3.5.7 Machine Translation Robustness . . . 44
3.5.8 Case Study . . . 45
3.6 Summary . . . 45

4 Retrieving Global Context **47**
4.1 Introduction . . . 47
4.2 Related Work . . . 49
4.3 Model . . . 52
4.3.1 Transformer-XL . . . 52
4.3.2 Relational Memory . . . 53
4.3.2.1 Open Information Extraction . . . 54
4.3.2.2 Relation Retrieval . . . 56
4.3.2.3 Integration with Transformer-XL . . . 57
4.4 Experiments . . . 59
4.4.1 Datasets and OpenIE . . . 59
4.4.2 Implementation Details . . . 60
4.4.3 Main Results . . . 61
4.5 Analysis . . . 62
4.5.1 Ablations and Design Choice Studies . . . 62
4.5.2 Does Relational Memory Improve Coherence? . . . 66
4.5.3 Qualitative Analysis . . . 69
4.6 Summary . . . 73

5 Pretraining the Noisy Channel Model with External Context **74**

5.1 Introduction . . . 74

5.2 Related Work . . . 76

5.3 A Seq-to-Seq Dialogue Model . . . 78

5.4 Noisy Channel Model for Dialogue . . . 80

5.4.1 Decoding . . . 82

5.5 Pretraining with External Context . . . 84

5.5.1 Implementation Details . . . 86

5.6 Experiments . . . 88

5.6.1 Datasets . . . 88

5.6.2 Fine-Tuning . . . 90

5.6.3 Decoding . . . 90

5.6.4 Evaluation Metrics . . . 91

5.7 Results . . . 92

5.8 Analysis . . . 95

5.8.1 Ablation on Model Combination . . . 96

5.8.2 Effect of Pretraining Scale . . . 96

5.8.3 Sample Efficiency of Fine-Tuning . . . 97

5.8.4 Decoding Runtime . . . 98

5.8.5 Decoding Properties . . . 98

5.9 Summary . . . 100

6 Conclusions **101**

List of Figures

1.1 The architecture for learning contextual embeddings. 4

2.1 A typical self-attention block. 10
2.2 Autoregressive language models with transformer. 12
2.3 Seq2seq language models with transformer. 16
2.4 Masked language models with transformer. 17
2.5 Text infilling with transformer. 20

3.1 We interpret a translation language model $p(Y_j|\mathcal{X}, \mathcal{Y}_{-j})$ ($\mathcal{Y}_{-j}$ means that phrase Y_j has been removed from sequence $\mathcal{Y}$) as a causal model. The randomness of the causal model comes from unobserved variables $\mathbf{G}$. For data augmentation, we sample counterfactual parallel sequences based on the causal effects singled-out by an unsupervised alignment model (i.e., the black arrows from $\mathcal{X}$ to $\mathcal{Y}$ above). 25
3.2 The three steps of Translation-Counterfactual Word Replacement. See text for details. 31
3.3 BLEU scores on the development set of the WMT'18 English $\rightarrow$ Turkish task with different pre-training steps. 39
3.4 Learning curves of training language models for the WMT'18 English $\rightarrow$ Turkish task with either the random initialization or the XLM initialization. 40
3.5 BLEU scores on the development set of the WMT'18 English $\rightarrow$ Turkish task with different sampling probabilities. 41

4.1 We identify salient entities in the previous text segment and extract relations to build our relational memory. We encode each relation with an LSTM encoder, aggregate the resulting representations into a vector, and use a gate mechanism that allows our language model to adaptively take advantage of relational information for predicting the next token. 53
4.2 Perplexity on WikiText-103 with different number of relation triples. 64
4.3 Increasing extended memory length. 65
4.4 Heatmap of gate values. 70
4.5 Histogram of gate values g_t. 71

5.1 The data flow of one turn in a task-oriented dialogue for train booking from MultiWOZ. 78
5.2 Results showing the effect of pretraining scale. 96
5.3 Pretraining improves sample efficiency during fine-tuning. 97

List of Tables

3.1 The statistics of the monolingual datasets. 37
3.2 The statistics of the parallel corpora. 37
3.3 The BLEU scores on the development set of the WMT'18 English → Turkish task with source context, target context, alignment, and Gumbel ablation. 41
3.4 The BLEU scores on the testing sets of En-Tr, En-De and En-Vi. The baseline method denotes training without any data augmentation. 42
3.5 The BLEU scores on the testing sets with backtranslation and TCWR. 43
3.6 The BLEU scores on the WMT'19 English → French robustness task. 44
3.7 A case study on TCWR, where augmented positions are marked as [original / **substituted**]. 45

4.1 Statistics of datasets used in our experiments. For each subset, we show the number of (sub)words for WikiText-103 and WMT19 or the number of characters for enwik8. 59
4.2 We use perplexity (↓) on WikiText-103 and WMT19 and bits per character (↓) on enwik8 for evaluation. 62
4.3 RELATIONLM with OpenIE or Freebase triples. 63
4.4 Ablating relation and/or tail entity from a relation triple. 64
4.5 Perplexity with and without dynamic OpenIE. 66
4.6 Perplexity with different entity scoring methods. 66
4.7 The unit is second/step. We use batch size 128 and 1 per step for training and evaluation, respectively. 66

4.8 Statistics of entity and non-entity tokens. 67
4.9 Knowledge perplexity (↓) and non-entity perplexity (↓). 67
4.10 Knowledge F_1 (↑). 68
4.11 We show the number of contexts in which a continuation from a particular model is chosen by human evaluators for each evaluation criterion. Recall that the total number of contexts used for human evaluation is 1,000. Since we have five annotators, we use majority voting to decide the favored model for each continuation. We use the Kappa statistic to measure inter-annotator agreement. The statistic is 0.64, which shows substantial agreement among the annotators. 70
4.12 Three examples of text segment and retrieved relations (based on previous text segments). 72

5.1 Statistics of task-oriented dialogue datasets. We define a multi-task dialogue as a dialogue involving multiple tasks, e.g. hotel and restaurant booking, while its counterpart handles a single task, e.g. hotel booking. Taskmaster and CamRest676 do not contain any multi-task dialogues. 88
5.2 MultiWOZ test results (end-to-end modeling with generated beliefs) with seq2seq approaches. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding. † This work also reports a combined score of 105.1 with an alternative context and evaluation setting, contributions orthogonal to our work and the other benchmarks reported here. 92

5.3 CamRest676 test results (end-to-end modeling with generated beliefs) with seq2seq approaches. Noisy channel reranking performs comparable with noisy channel online decoding, and the results are not shown. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding. 93
5.4 SMCalFlow results. Reranking performs worse than online decoding, and the results are not shown. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding. . . 94
5.5 Ablation results for model combination on development sets (combined score). Results for reranking are similar and are not shown. 'All', 'Direct' 'Source', and 'Channel' denote no ablation, direct model, source model and channel model, respectively. Rows with '+' are combinations of two sub-modules, while the rows with '-' are combinations of three sub-modules. 95
5.6 Average decoding time (in seconds) for each turn with different decoding methods. 98
5.7 The average length of responses with different decoding methods (on test set). The value closest to the ground truth is bold. 99
5.8 The Zipf scores of responses with different decoding methods (on test set). The value closest to the ground truth is bold. 99
5.9 The likelihood (%) of falling into repetition loops for different decoding methods (on test set). 99
5.10 Case study on the responses decoded by direct decoding, noisy channel reranking and noisy channel online decoding. The large noisy channel model is used. 100

List of Related Publications

Qi Liu, Matt J. Kusner, Phil Blunsom. *"Counterfactual Data Augmentation for Neural Machine Translation"*. Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 2021.

Qi Liu, Lei Yu, Laura Rimell, Phil Blunsom. *"Pretraining the Noisy Channel Model for Task-Oriented Dialogue"*. Transactions of the Association for Computational Linguistics (TACL), 2021.

Qi Liu, Dani Yogatama, Phil Blunsom. *"Relational Memory Augmented Language Models"*. Transactions of the Association for Computational Linguistics (TACL), 2022.

1 | Introduction

Learning distributed representations of natural language has become a common practice for NLP. Neural networks are widely adopted for learning distributed representations. Previous work (Mikolov et al., 2013b; Zhila et al., 2013) has shown that neural networks perform significantly better compared to their counterparts, such as latent semantic analysis (Landauer and Dumais, 2008) and latent Dirichlet allocation (Blei et al., 2003), in preserving linguistic regularities. In this thesis, we focus on using neural networks to learn dense, distributed and fixed-length representations of tokens and sequences and use these representations to improve NLP tasks such as machine translation, language modelling and task-oriented dialogue.

Non-contextual embedding is one popular way to learn from large-scale textual corpora. Based on the distributional hypothesis (Harris, 1954), which states that tokens that are used in similar contexts tend to have similar meanings, they map each token in the vocabulary to a low-dimensional real-valued vector according to their contexts. Two well-known examples include word2vec (Mikolov et al., 2013a) and Glove (Pennington et al., 2014). The continuous bag-of-words model and the continuous skip-gram model are two popular ways to train word2vec. The continuous bag-of-words model predicts the target token given its context, while the skip-gram model predicts each token in the context given the target token. Glove trains a weighted least squares model that makes efficient use of global word-word co-occurrence counts. The representations learned with these models perform competitively on word-level tasks, e.g. word analogy (Mikolov et al., 2013a) and word similarity (Luong et al., 2013). They can also reproduce semantic and syntactic patterns with vector arithmetic operations. For example, vector(“King”) - vector(“Man”) + vector(“Woman”) results in a vector that is closest to the representation of the word “Queen” with word2vec. However, non-

contextual embeddings map each token in the vocabulary to a vector independent of its context and fail to distinguish the semantics of words in different contexts. This gives rise to contextual embeddings, where each token is associated with a representation that is a function of the entire input sequence. Contextual embeddings (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2018) are shown to better capture complex characteristics of word use and model polysemous words. In this thesis, we follow up the line of research on contextual embeddings and explore better methods for conditioning on context for natural language processing. We explore three different sources of context including local context of each input sequence, global context from training data and external context from the Web.

In the remainder of this chapter, we first formalise contextual embeddings (§1.1) and then describe the contributions of the thesis (§1.2). We conclude this chapter by providing an outline of the thesis (§1.3).

1.1 Formalism of Contextual Embeddings

Consider a text corpus that consists of textual sequences $\mathcal{S}$, where each sequence is represented as a series of tokens, $(x_1, x_2, ..., x_N)$. Distributed representations of words associate each token x_i with a dense feature vector $\mathbf{h}_{x_i}$. Non-contextual embeddings 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. On the other hand, as shown in Figure 1.1, contextual embeddings associate each token x_i with a representation that is a function of the entire input sequence $\mathcal{S}$, i.e. $\mathbf{h}_{x_i} = f(\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, ..., \mathbf{e}_{x_N})$, where each input token x_i is usually mapped to its non-contextualized representation $\mathbf{e}_{x_i}$, before applying an aggregation function f. The aggregation function is usually implemented with contextual models, such

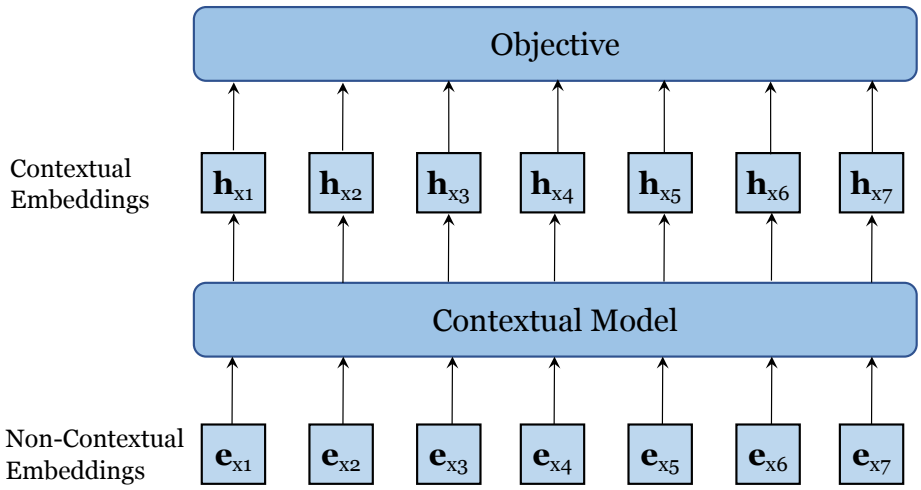


Figure 1.1: The architecture for learning contextual embeddings.

as LSTM (Hochreiter and Schmidhuber, 1997) and transformer (Vaswani et al., 2017). Contextual embeddings are learned with different supervised and unsupervised objectives. We detail widely-adopted objectives in Chapter 2.

1.2 Contributions

In this thesis, we explore better methods of conditioning on context for natural language processing. We explore three different sources of context including local context of each input sequence, global context from training data and external context from the Web. In the rest of this section, we briefly introduce these three aspects.

Local context. Given a sequence of tokens $\mathcal{S} = (x_1, x_2, ..., x_N)$, we refer to the local context of each token x_i as all the tokens $(x_1, x_2, ..., x_N)$ in $\mathcal{S}$. Local context provides both surface forms and ordering of tokens, which are particularly helpful for disambiguating polysemous tokens. For example, we can easily disambiguate the word "bank" based on its local contexts in the sequences "We should deposit

this money in a bank." and "We're camping near the river bank.". In particular, we investigate the problem of better utilizing local context for augmenting a Neural Machine Translation (NMT) dataset. Given a set of parallel sequences, $\mathcal{S} = \{(\mathcal{X}^i, \mathcal{Y}^i)\}_{i=1}^N$, e.g. Chinese sequences and their English translations, the goal of NMT is to translate the source sequence $\mathcal{X}$ to the target sequence $\mathcal{Y}$, i.e. modelling $p(\mathcal{Y}|\mathcal{X})$. As most NMT models rely on large-scale parallel corpora to obtain good performance, data augmentation is a promising direction to achieve this goal, which works by enlarging the number of data points for training without expensive manual annotations. We propose a new data augmentation algorithm for NMT that utilizes both local context and alignment between source and target sequences and demonstrate that our algorithm can substantially improve the performance of translation. It works by interpreting language models and phrasal alignment causally. Specifically, it creates augmented parallel translation corpora by generating (path-specific) counterfactual aligned phrases. We generate these by sampling new source phrases from a masked language model, then sampling an aligned counterfactual target phrase by noting that a translation language model can be interpreted as a Gumbel-Max Structural Causal Model (Oberst and Sontag, 2019). Our method can well maintain the symmetry between source and target sequences. Experiments on IWSLT'15 English $\rightarrow$ Vietnamese, WMT'17 English $\rightarrow$ German, WMT'18 English $\rightarrow$ Turkish, and WMT'19 robust English $\rightarrow$ French show that the method can improve the performance of translation, backtranslation and translation robustness.

Qi Liu, Matt J. Kusner, Phil Blunsom. *"Counterfactual Data Augmentation for Neural Machine Translation"*. Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 2021.

Global context. Given a training corpus that consists of textual sequences, each entity e may appear in different sequences. We refer to the sequences that contain the entity e as the global context of e as they provide "global" information about e in the training corpus. We move beyond local context and explicitly condition on global context of each entity for improving language modelling. More specifically, we use Open Information Extraction (OpenIE) to extract relation triples (head entity, relation, tail entity) from $\{\mathcal{S}_1, ..., \mathcal{S}_M\}$. These relation triples form a corpus-specific knowledge graph. Given an entity e, we can efficiently retrieve all the relation triples relevant to e. These relation triples concisely represent the global context of e. A relational memory is constructed based on these retrieved relation triples. We condition an autoregressive language model on the relational memory and design a gating function that allows the language model to adaptively combine information from retrieved relations and observed textual context to predict the next token. Experiments on WikiText-103, WMT19, and enwik8 English datasets demonstrate that our approach produces a better language model in terms of perplexity and bits per character. We also show that relational memory improves coherence, is complementary to token-based memory, and enables causal interventions. Our model provides a simple yet effective way to combine an autoregressive language model and a corpus-specific knowledge graph for more coherent and logical generation.

> Qi Liu, Lei Yu, Laura Rimell, Phil Blunsom. *"Pretraining the Noisy Channel Model for Task-Oriented Dialogue"*. Transactions of the Association for Computational Linguistics (TACL), 2021.

External context. Beyond training corpus, we can incorporate resources from the Web to learn more universal representations. We refer to the resources beyond training corpus as external context in this thesis. In particular, we study

pretraining with external context for task-oriented dialogue. We use a two-stage pretraining strategy. In the first stage, we use a Reddit dump for pretraining. In the second stage, we use two task-oriented dialogue datasets, Taskmaster (Byrne et al., 2019) and Schema-Guided Dialogue (Rastogi et al., 2020), to specialize the Reddit pretrained models. Since the Reddit data consists of open-domain-style dialogues, pretraining on these two datasets can adapt the models for task-oriented dialogue. Besides, we observe that direct decoding for task-oriented dialogue suffers from the explaining-away effect, manifested in models that prefer short and generic responses. We argue for the use of Bayes' theorem to factorize the dialogue task into two models, the distribution of the context given the response, and the prior for the response itself. This approach, an instantiation of the noisy channel model, both mitigates the explaining-away effect and allows the principled incorporation of large pretrained models for the response prior. We present extensive experiments showing that a noisy channel model decodes better responses compared to direct decoding and that the two stage pretraining strategy improves over randomly initialized models.

Qi Liu, Dani Yogatama, Phil Blunsom. *"Relational Memory Augmented Language Models"*. Transactions of the Association for Computational Linguistics (TACL), 2022.

1.3 Thesis Outline

In this section, we outline the contents of each chapter.

Chapter 2: Contextual Embeddings — We provide an overview of the widely-adopted objectives to learn contextual embeddings. We introduce autoregressive language models, masked language models, text infilling,

and seq2seq language models. We also show prototypical models for each objective.

Chapter 3: Utilizing Local Context — We study utilizing local context for augmenting a machine translation dataset. We propose a data augmentation method, which introduces a causal inductive bias that takes both context and alignment into account.

Chapter 4: Retrieving Global Context — We present RELATIONLM, a language model augmented with relational memory that stores retrieved relation triples of entities. We show how to use OpenIE to obtain a corpus-specific knowledge graph and how to combine it with a state-of-the-art language model.

Chapter 5: Pre-Training with External Context — We study using the noisy channel model for task-oriented dialogue. We apply large-scale pretraining with external context to tackle data scarcity in downstream tasks.

Chapter 6: Conclusions — We summarise the thesis with future work.

2 | Contextual Embeddings

Contextual embeddings associate each token with a representation that is a function of the entire input sequence. In this chapter, we provide an overview of ways to learn contextual embeddings and the linguistic knowledge encoded in contextual embeddings. Although contextual embeddings can be learned with different ways, we focus on self-supervised learned contextual embeddings in this chapter. Self-supervised learning usually consists of two steps. In the first step, it constructs pseudo labels from unlabelled data. In the second step, the model takes the unlabelled data to predict pseudo labels. As a result, self-supervised learning can make use of large-scale unlabelled data for training.

We organize this chapter as follows. We first review transformer, a popular architecture for learning contextual embeddings (§2.1). Then, we introduce four self-supervised objectives, including autoregressive language models (§2.2.1), masked language models (§2.2.3), text infilling (§2.2.4), and seq2seq language models (§2.2.2). Next, we study the work that analyzes the hierarchical features emerging from self-supervised learning (§2.3). Finally, we summarize this chapter (§2.4).

2.1 Transformer

Recurrent neural networks (RNNs), such as long short-term memory (Hochreiter and Schmidhuber, 1997) and gated recurrent units (Chung et al., 2014), process the input sequentially. At each step the hidden state of the previous step $t-1$ and the input at position t are aggregated to obtain the hidden state of the step t. Despite the success of applying RNNs for NLP, they have been shown to suffer from several limitations. First, the sequential nature of RNNs precludes parallelization within training examples, which negatively affects their efficiency for

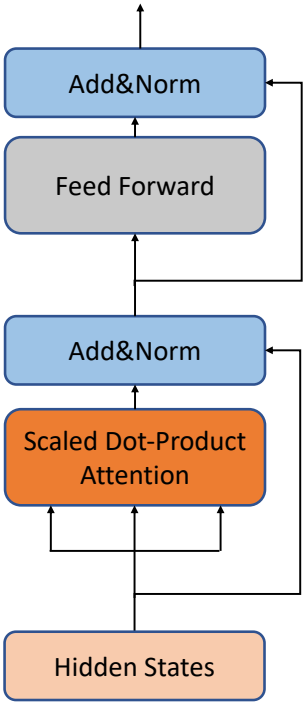


Figure 2.1: A typical self-attention block.

processing long sequences. Second, the sequential information flow also leads to worse performance in capturing long-range dependencies (Koehn and Knowles, 2017). Transformer (Vaswani et al., 2017) has become a promising alternative to RNNs. Transformer eschews recurrence and instead relies on an attention mechanism named self-attention, to draw global dependencies between input and output.

Given a sequence $\{x_1, ..., x_N\}$, the initial representation of each token is a summation of its word embedding and position embedding to capture the order of the sequence. Transformer stacks several self-attention blocks to draw global dependencies between tokens. In each self-attention block, as shown in Figure 2.1, transformer first applies scaled dot-product attention. Then, it uses a feed forward network to transform the output of scaled dot-product attention. Two residual connections are added in each block. Next we detail scaled dot-product attention. Given the hidden states $\{\mathbf{h}_1, ..., \mathbf{h}_N\}$ of a sequence, we use $\mathbf{H}$ to denote

the stacked states, where each row is a hidden state. Next transformer calculates queries, keys, and values with three learnable matrices $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$.

$$\mathbf{Q} = \mathbf{H}\mathbf{W}_q,\ \mathbf{K} = \mathbf{H}\mathbf{W}_k,\ \mathbf{V} = \mathbf{H}\mathbf{W}_v. \tag{2.1}$$

Finally, scaled dot-product attention uses dot product scaled by $\sqrt{d}$ to measure the compatibility between queries and keys, where we use d to denote the number of dimensions. The compatibility scores are normalized with softmax and the output is a weighted sum of values.

$$\mathbf{H}' = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{\mathrm{d_{qk}}}})\mathbf{V}. \tag{2.2}$$

In Eq. 2.2, each token can attend to all tokens in a sequence. This is particularly helpful for using transformer as an encoder where a sequence is given as input. However, when using transformer as a decoder for generating sequences, each token is not allowed to attend to future tokens. To this end a simple modification to Eq. 2.2 is adding a mask $\mathbf{M}$, e.g. a causal attention mask, to avoid using future information:

$$\mathbf{H}' = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{\mathrm{d_{qk}}}} + \mathbf{M})\mathbf{V}, \tag{2.3}$$

where $\mathbf{M}_{ij} = 0$ if i can attend to j and $\mathbf{M}_{ij} = -\infty$ otherwise.

2.2 Objectives for Learning Contextual Embeddings

In this section, we introduce some objectives for learning contextual embeddings.

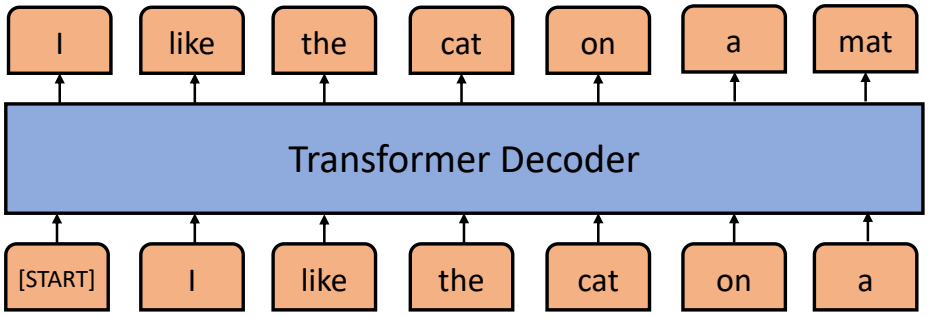


Figure 2.2: Autoregressive language models with transformer.

2.2.1 Autoregressive Language Models

Autoregressive language models are a common way to learn contextual embeddings. The goal of autoregressive language models is to learn the joint distribution of sequences of tokens. Given a sequence of N tokens, $(x_1, x_2, ..., x_N)$, an autoregressive language model factorizes the probability of the sequence as:

$$p(x_1, x_2, ..., x_N) = \prod_{i=1}^{N} p(x_i | x_1, x_2, ..., x_{i-1}). \tag{2.4}$$

Applying the chain rule, a left-to-right autoregressive language model takes the left context $x_1, x_2, ..., x_{i-1}$ to estimate the conditional probability of x_i. Autoregressive language modelling often uses the maximum likelihood estimation for optimization. An example of autoregressive language models with transformer is shown in Figure 2.2. As textual data are easy to obtain, autoregressive language models are often trained with large-scale unlabelled text corpora.

Some notable models using the autoregressive language modelling are demonstrated below.

Precursor models. Dai and Le (2015) is the first work we are aware of that uses autoregressive language modelling together with a sequence autoencoder to im-

prove 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. (2017) 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 a seq2seq model are initialized with the pre-trained weights of two autoregressive language models. These autoregressive language models are separately trained on either the News Crawl English or German corpora for machine translation, while both are initialized with the autoregressive 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, achieving better results over baselines without pre-training.

ELMo. ELMo (Peters et al., 2018) generalizes traditional word embeddings by extracting context-dependent representations from a bidirectional autoregressive 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 left-to-right and right-to-left representations are concatenated, 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 $(L+1)$ representations (including the global word embedding and L hidden representations) of each token k are aggregated as:

$$\text{ELMo}_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}, \tag{2.5}$$

where $\mathbf{s}^{task}$ are layer-wise weights to linearly combine the $(L+1)$ 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. As most supervised models use global word representations in their initial layer, these representations can be concatenated with their corresponding context-dependent representations ELMO_k^{task} before feeding them to higher layers.

GPT, GPT2, and GPT3. GPT (Radford et al., 2018) follows the pre-training then fine-tuning paradigm: (a) unsupervised pre-training with an autoregressive 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, to train an autoregressive language model. Transformer (Vaswani et al., 2017) is used to implement the autoregressive language model. 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., 2016). It pre-processes each text input as a single contiguous sequence of tokens with special tokens including [START] (the start of a sequence), [DELIM] (delimiting two sequences) and [EXTRACT] (the end of the input).

GPT2 (Radford et al., 2019) mainly follows the architecture of GPT and trains an autoregressive 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 of 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., 2016), 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.

Different from the pre-training then fine-tuning paradigm, GPT3 (Brown et al., 2020) proposes a prompt-based method to interact with large language models with only a few annotated examples. This is motivated by the fact that humans can generally comprehend a new language task from only a few examples, while most NLP systems still struggle to do so. GPT3 finds that scaling up autoregressive language models greatly improves task-agnostic, few-shot performance. The largest model of GPT-3 is an autoregressive language model with 175 billion parameters and is tested in the few-shot setting. For all tasks, GPT-3 is applied without any fine-tuning. The tasks and few-shot demonstrations are specified purely via text input. GPT-3 achieves strong performance on a wide range of NLP datasets, such as translation, question-answering, and cloze tasks.

2.2.2 Seq2seq Language Models

Seq2seq language models map an input sequence $\mathcal{X}$ to an output sequence $\mathcal{Y}$ by modelling $p(\mathcal{Y}|\mathcal{X})$. An example is shown in Figure 2.3. The model conditions on the partial sequence "I like the cat on" to predict "a mat", i.e. it calcualtes $p(x_6 = \text{a}, x_7 = \text{mat}|x_1 = \text{I}, x_2 = \text{like}, x_3 = \text{the}, x_4 = \text{cat}, x_5 = \text{on})$. This objective is well-suited for pre-training with datasets consisting of sequence pairs, e.g. machine

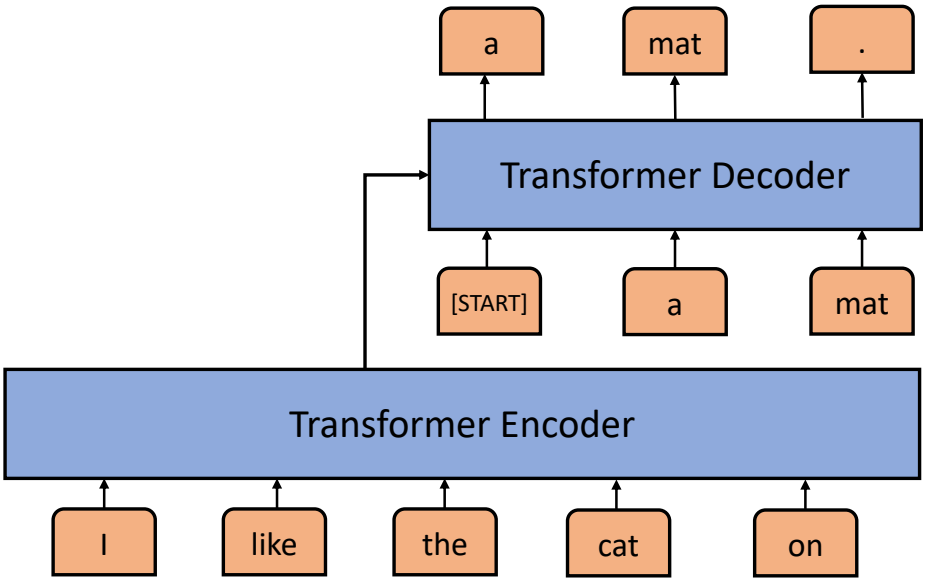


Figure 2.3: Seq2seq language models with transformer.

translation datasets (source and target sequences) and dialogue datasets (dialogue contexts and responses).

Some notable models using the seq2seq language modelling are demonstrated below.

UniLM. UniLM (Dong et al., 2019) adopts three objectives: (a) autoregressive language modelling, (b) masked language modelling, and (c) seq2seq language modelling, for pre-training a transformer. To implement three objectives in a single network, UniLM utilizes specific self-attention masks to control data flow. For example, masked language modelling can attend to bidirectional context, while seq2seq language modelling can attend to bidirectional context for a source sequence and left context only for a target sequence.

Plato. Plato (Bao et al., 2020) uses dialogue-style data from social networks to pre-train a dialogue model. It constructs pre-training data into (context, response)

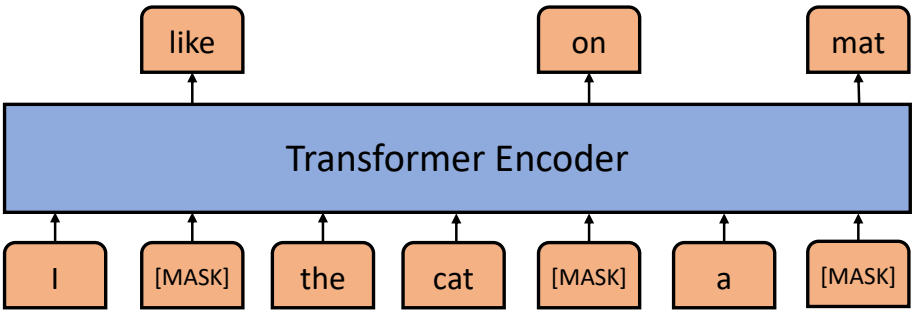


Figure 2.4: Masked language models with transformer.

pairs and uses the seq2seq language modelling for pre-training. Moreover, it designs a response selection task for pre-training, where the model needs to select the real response from a list of candidates.

2.2.3 Masked Language Models

Given a sequence $\{x_1, ..., x_N\}$, masked language modelling (Devlin et al., 2019) usually consists of two steps:

1. The model randomly masks some input tokens. Each masked token is replaced by the special token [MASK].

2. The model reconstructs the masked tokens. An example is shown in Figure 2.4, the model predicts the masked tokens with a transformer encoder, i.e. it calculates $p(x_2 = \text{like}, x_5 = \text{on}, x_7 = \text{mat} | x_1 = \text{I}, x_2 = \text{[MASK]}, x_3 = \text{the}, x_4 = \text{cat}, x_5 = \text{[MASK]}, x_6 = \text{a}, x_7 = \text{[MASK]})$.

Compared to the autoregressive language models, where the next token is predicted with either its left or right context, the masked tokens are predicted with both left-to-right and right-to-left contexts for masked language models.

Some notable models using the masked language modelling are demonstrated below.

BERT. BERT (Devlin et al., 2019) proposes the masked language modelling objective. BERT applies a transformer encoder to attend to bi-directional contexts during pre-training. BERT also 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. Specifically, the first token is always a special classification token [CLS], and sentence pairs are separated by a special token [SEP]. BERT adopts the pre-training then fine-tuning scheme. The hidden state of [CLS] is used for sentence-level tasks and the hidden state of each token is used for token-level tasks. BERT improves the GLUE (Wang et al., 2019a) score to 80.5%. However, it is unclear why BERT improves over prior work as it uses different objectives, datasets (Wikipedia and BookCorpus) and architectures compared to previous methods. We refer readers to (Raffel et al., 2020) for a controlled comparison between unidirectional and bidirectional models and autoregressive language modelling and masked language modelling.

BERT variants. Some follow-up projects aim to improve the objective and architecture of BERT.

Instead of randomly masking tokens, ERNIE (Sun et al., 2019) incorporates knowledge masking strategies, including entity-level masking and phrase-level masking. SpanBERT (Joshi et al., 2020) generalizes ERNIE to mask random spans, without referring to external knowledge. StructBERT (Wang et al., 2020b) 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 consec-

utive segments.

RoBERTa (Liu et al., 2019b) 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 (Cer et al., 2017) 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.

2.2.4 Text Infilling

Masked language modelling randomly masks some input tokens. Each masked token is replaced by the special token [MASK]. As a result, the corruption step of masked language modelling does not change the input length. Besides, the masked tokens are not necessarily contiguous. Text infilling improves masked language modelling by masking contiguous tokens with variable length. The masked tokens are replaced by a single [MASK] token. The model needs to infer input length and reconstruct the masked tokens. An example is shown in Figure 2.5, continuous tokens "the cat on" are masked and the model needs to reconstruct the tokens based on the input "I like [MASK] a mat.". Compared to masked language

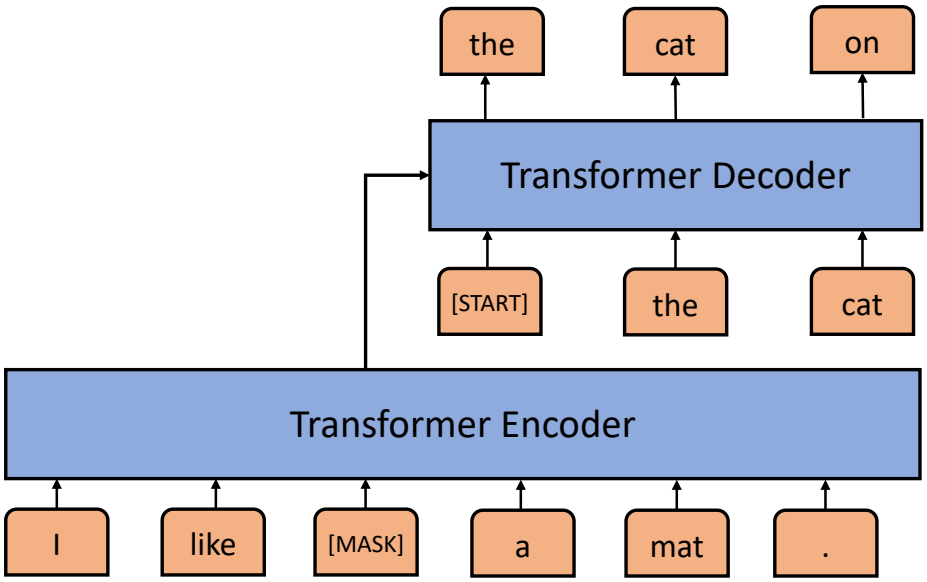


Figure 2.5: Text infilling with transformer.

modelling, text infilling is harder. The reasons are two-fold. First, the model needs to infer the length of masked tokens, as only a single [MASK] token is given. Second, predicting contiguous tokens is harder than predicting non-contiguous tokens. For example, predicting "New York" is much harder than predicting "New" when the next word "York" is given.

Some notable models using text infilling are demonstrated below.

T5. Raffel et al. (2020) propose T5 (Text-to-Text Transfer Transformer), unifying natural language understanding and generation and applying an encoder-decoder framework.

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 uses text infilling as the objective. It also uses multi-task fine-tuning on GLUE or SuperGLUE. For fine-tuning, to convert the input data into a text-to-text format, T5 utilizes some tokens in the vocabulary as labels. For example, the tokens "entailment", "contradiction", and "neutral" are used as the labels for natural language inference tasks. For regression tasks (e.g. STS-B (Cer et al., 2017)), T5 simply rounds up the scores to the nearest multiple of 0.2 and converts the numbers 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 source sequence for English-to-German translation datasets.

BART. BART (Lewis et al., 2020a) introduces additional noising functions beyond masked language modelling for pre-training sequence-to-sequence models. First, the input sequence is corrupted with an noising function. Then, the corrupted input is reconstructed by a transformer. BART evaluates a wide variety of noising functions, including token masking, token deletion, text infilling, document rotation, and sentence shuffling (randomly shuffling words of a sentence). The best performance is achieved by using both sentence shuffling and text infilling.

2.3 Linguistic Analysis

A good representation of language should capture implicit linguistic knowledge in textual data, such as lexical meanings, syntactic structures and semantic information. Previous work finds that such knowledge emerges from learning contextual embeddings. In this section, we introduce some work studying contextual embeddings using probe classifiers and 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 of embeddings encode the distances of tokens in syntactic parsing trees (e.g. dependency trees).

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

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

2.4 Summary

In this chapter, we have introduced transformer, a popular architecture for learning contextual embeddings. Then, we have demonstrated four training objectives, including autoregressive language models, masked language models, text infilling, and seq2seq language models. We have also reviewed the work that analyzes contextual embeddings with probes and visualization. In the next three chapters, we will study using local context (Chapter 3), global context (Chapter 4) and external context (Chapter 5) for different NLP tasks.

3 | Utilizing Local Context

3.1 Introduction

Given a sequence of tokens $\mathcal{S} = (x_1, x_2, ..., x_N)$, we refer to the local context of each token x_i as all the tokens $(x_1, x_2, ..., x_N)$ in $\mathcal{S}$. In Chapter 2, we have reviewed different objectives to utilize local context for learning contextual embeddings. In this chapter, we specifically study utilizing local context for augmenting a Neural Machine Translation (NMT) dataset. NMT datasets contain parallel sequences $(\mathcal{X}, \mathcal{Y})$, where the source sequence $\mathcal{X}$ and the target sequence $\mathcal{Y}$ are highly aligned. The goal of NMT is to translate the source sequence $\mathcal{X}$ to the target sequence $\mathcal{Y}$, i.e. modelling $p(\mathcal{Y}|\mathcal{X})$. As most NMT models rely on large-scale parallel corpora to obtain good performance, data augmentation is a promising direction to tackle data scarcity and improve robustness of NMT models. It works by enlarging the number of data points for training without manually collecting new data. It is widely used to improve diversity and robustness and to avoid overfitting on small datasets. Even though data augmentation (e.g. image flipping, cropping and blurring) has become a standard technique in computer vision (Krizhevsky et al., 2017; Li et al., 2022; Chen et al., 2020), it is non-trivial to apply in machine translation since even a slight modification to a sequence can result in drastic changes in its syntax and semantics and break the symmetry between source and target sequences. Indeed there is relatively little work in this direction due to these difficulties (Sennrich et al., 2016a; Fadaee et al., 2017; Wang et al., 2018c; Gao et al., 2019; Xia et al., 2019; Kobayashi, 2018). Further, previous work based on word replacement either ignores the contexts of replaced words or breaks the alignment between source and target sequences, both detrimental for generating high-quality data.

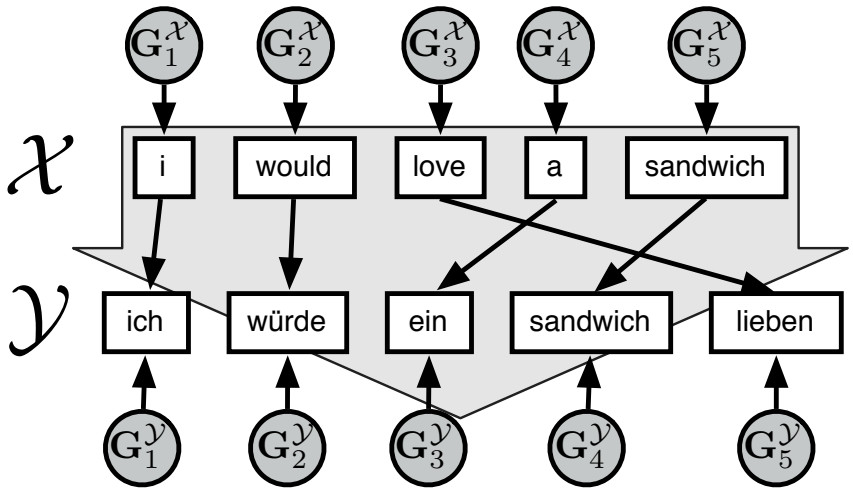


Figure 3.1: We interpret a translation language model $p(Y_j|\mathcal{X}, \mathcal{Y}_{-j})$ ($\mathcal{Y}_{-j}$ means that phrase Y_j has been removed from sequence $\mathcal{Y}$) as a causal model. The randomness of the causal model comes from unobserved variables $\mathbf{G}$. For data augmentation, we sample counterfactual parallel sequences based on the causal effects singled-out by an unsupervised alignment model (i.e., the black arrows from $\mathcal{X}$ to $\mathcal{Y}$ above).

Our data augmentation algorithm is based on the observation that a translation language model can be interpreted as a causal model, as described in Figure 3.1. Doing so allows us to ask *counterfactual* questions of the form: Given source and target sequences, if a phrase in the source sequence is changed, how would the target sequence change? We propose a data augmentation method for machine translation that generates counterfactual parallel translation data.

To ensure these counterfactuals are close to the original data we sample a new source phrase from a masked language model. We then consider the (path-specific) counterfactual target phrase that is aligned to that source phrase (given by an unsupervised phrasal alignment method). The idea is that this augmentation procedure exposes inductive biases in existing language models that enables new translation models to learn more efficiently and exhibit more robust generalisation. Specifically, our augmentation procedure performs the following three steps:

1. We utilize unsupervised phrasal alignment (e.g. Neubig et al. (2011) and Dyer et al. (2013)) to obtain correspondences between source and target phrases.

2. A source phrase is removed and then resampled according to a trained masked language model (Devlin et al., 2019; Raffel et al., 2020).

3. We perform (path-specific) counterfactual inference on the causal model given by a trained translation language model (Conneau and Lample, 2019) to resample *only* the aligned target phrase, given the changed source phrase.

Different from prior work, our approach takes advantage of both source/target *context* and *alignment* for data augmentation. Experiments on IWSLT'15 English $\rightarrow$ Vietnamese, WMT'17 English $\rightarrow$ German, and WMT'18 English $\rightarrow$ Turkish show that our method improves the translation performance on both high-resource and low-resource datasets. We additionally demonstrate that our method complements existing approaches such as backtranslation (Sennrich et al., 2016b). Finally, we demonstrate that our method improves translation robustness (we evaluate this on the WMT'19 English $\rightarrow$ French robustness dataset).

In the rest of the chapter, we first introduce related work (§3.2). Next, we introduce the background of neural machine translation, phrasal alignment and causal modelling (§3.3). Then, we explain our algorithm based on a procedure named translation-counterfactual word replacement (§3.4.1). Finally, we evaluate the effectiveness of our data augmentation algorithm (§3.5).

3.2 Related Work

3.2.1 Data Augmentation for NMT

We categorize previous work on data augmentation for NMT into two classes, *word replacement* and *backtranslation*.

Word replacement. WordDropout (Sennrich et al., 2016a) randomly zeros out word embeddings in order to introduce noises. BPEDropout (Provilkov et al., 2020) stochastically corrupts the segmentation procedure of BPE, leading to different subword segmentations with the same BPE vocabulary. RAML (Norouzi et al., 2016) applies a reward-augmented maximum likelihood objective, which essentially augments target sequences with sequences sampled based on metrics, such as edit distance and BLEU score (Wang et al., 2018c). SwitchOut (Wang et al., 2018c) extends RAML, augmenting both source and target sequences by randomly replacing words with noisy words sampled from a uniform distribution. These works do not take context and alignment into account. TDA (Fadaee et al., 2017) first uses two uni-directional language models to replace a word in the source sequence, before replacing the corresponding word based on a bilingual lexicon. TDA does not consider contexts in target sequences and relies on a high-quality bilingual lexicon. SCDA (Gao et al., 2019) uses a soft augmentation approach, where the one-hot representation of a word is replaced by a soft distribution of words given by a language model. DADA (Cheng et al., 2019) uses gradient information to generate adversarial sequences for more robust NMT. AdvAug (Cheng et al., 2020) extends DADA, where embeddings of virtual sequences are sampled from an adversarial distribution for augmentation. SCDA and AdvAug ignore the alignment information, thereby breaking the symmetry of source and target sequences. While DADA takes both context and alignment into ac-

count, it replaces multiple words in source and target sequences simultaneously, which risks generating unnatural sequences. In our work, we utilize both alignment and contextual information to sequentially replace aligned phrases for better performance.

Backtranslation. The idea of backtranslation dates back to statistical machine translation (Goutte et al., 2009; Bojar and Tamchyna, 2011). Senrich et al. (2016a) use backtranslation, where monolingual sequences in the target language are translated into the source language, and obtain substantial improvements on the WMT and IWSLT tasks. Currey et al. (2017) apply backtranslation to low-resource languages, finding that even low-quality translations due to limited parallel corpora are beneficial. He et al. (2016) propose a dual learning framework, where the primal task (source-to-target translation) and the dual task (target-to-source translation) teach each other through a reinforcement learning process until convergence. Edunov et al. (2018) scale backtranslation to millions of monolingual data and obtain state-of-the-art performance on WMT'14 English→German. Xia et al. (2019) use a two-step pivoting method for improving backtranslation on low-resource languages. We show that TCWR can be used together with backtranslation and obtain further improvements.

3.3 Background

In this section we describe background on neural machine translation (NMT), phrasal alignment, and causal modelling.

Neural machine translation. Given a set of parallel sequences, $\mathcal{S} = \{(\mathcal{X}^i, \mathcal{Y}^i)\}_{i=1}^N$, NMT maximizes the log-likelihood of $\mathcal{Y}$ given $\mathcal{X}$, assuming each $(\mathcal{X}^i, \mathcal{Y}^i)$ pair is

independently and identically distributed:

$$\max_{\theta} \sum_{(\mathcal{X}^i, \mathcal{Y}^i) \in \mathcal{S}} \log p_\theta(\mathcal{Y}^i | \mathcal{X}^i).$$

However, paired sequences are usually expensive to collect, as it requires an expert to translate sequences $\mathcal{X}^i$ into another language $\mathcal{Y}^i$. Data augmentation aims to generate new parallel sequences $(\hat{\mathcal{X}}^i, \hat{\mathcal{Y}}^i)$ without manually collecting new data.

Phrasal alignment. Phrasal alignment identifies the translation relationships among phrases in parallel sequences. Given a parallel sequence $(\mathcal{X}, \mathcal{Y})$, where $\mathcal{X} = (X_1 = x_1, X_2 = x_2, ..., X_{|\mathcal{X}|} = x_{|\mathcal{X}|})$ and $\mathcal{Y} = (Y_1 = y_1, Y_2 = y_2, ..., Y_{|\mathcal{Y}|} = y_{|\mathcal{Y}|})$ (X/Y and x/y denote a phrase and its value, respectively), phrasal alignment h learns a mapping that projects each position i of $\mathcal{X}$ to a position j of $\mathcal{Y}$, i.e. $j = h(i)$. In our work, we use pialign (Neubig et al., 2011) to obtain alignments.

Causal modelling. We formulate causality using the *structural causal model* (SCM) framework of Pearl (2003). Each SCM is a set of structural equations represented by a graph. The edges of this graph specify the inputs and outputs of the structural equations. Specifically, a variable V_i is caused by a set of observable parent variables $pa(V_i)$ and unobserved variables $\mathbf{U}_i$ if there exists a (deterministic or stochastic) structural equation f_i:

$$V_i = f_i(pa(V_i), \mathbf{U}_i).$$

If the structural equations f are identified, it is possible to compute a causal quantity called *counterfactuals*. Counterfactuals are questions that, given the current state of the world, ask what would have changed if some variable V had been

different. For example, "Would a person have been able to obtain a visa if they had been born in a different country?". Formally we denote the counterfactual value of a variable V_i, had another variable $W \in pa(V_i)$ been $\hat{w}$ (i.e., compared to its observed value w) as $V_{i,W \leftarrow \hat{w}}$. To compute counterfactuals we can follow a three-step procedure (for more details see Chapter 4 of Pearl (2016)): 1. **Abduction**: Given a prior distribution on unobserved variables $p(\mathbf{U}_i)$, compute the posterior given all observed variables $\mathbf{V} = \mathbf{v}$: $p(\mathbf{U}_i|\mathbf{V} = \mathbf{v})$; 2. **Action**: Modify the structural equation for V_i, so that W is fixed to the counterfactual value $\hat{w}$ (the modified equation is denoted as $f_{i,\hat{w}}$); 3. **Prediction**: Compute the distribution $p(V_{i,W \leftarrow \hat{w}}|\mathbf{V} = \mathbf{v})$ using $p(\mathbf{U}_i|\mathbf{V} = \mathbf{v})$, the observed variables $\mathbf{v}$, and the modified structural equation $f_{i,\hat{w}}$.

3.4 Method

Our goal is to take an input sequence pair $(\mathcal{X}, \mathcal{Y})$ and create augmented data from it. We aim to do so by removing phrases, resampling them in the source sequence, and computing the counterfactual effect of doing so in the target sequence. We argue that for any such augmentation method for NMT, it is crucial to leverage both contextual and alignment information, for the following reasons. (1) **Context**: As contextual information is widely used to disambiguate words (Peters et al., 2018) and generate realistic-looking sequences (Zellers et al., 2019), it is critical to utilize contextual information to obtain grammatically-correct and semantically-sound sequences. (2) **Alignment**: Phrasal alignment plays a critical role in statistical machine translation (Brown et al., 1993; Vogel et al., 1996). As phrasal alignment provides information about which phrase in the source sequence produces a phrase in the target sequence, a data augmentation algorithm which disregards alignment risks breaking the symmetry between source and tar-

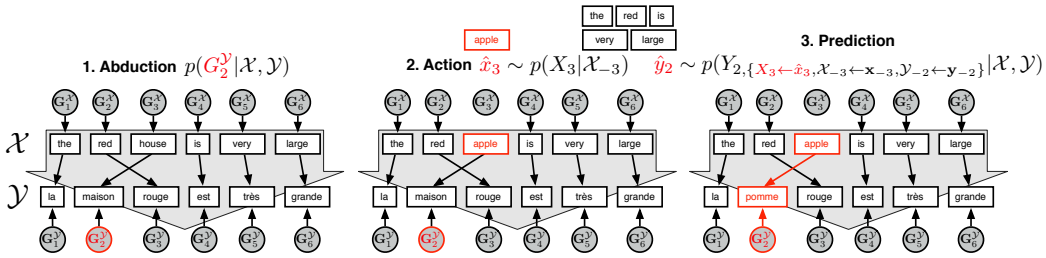


Figure 3.2: The three steps of Translation-Counterfactual Word Replacement. See text for details.

get sequences. To this end, in Section 3.4.1, we introduce a technique called *Translation-Counterfactual Word Replacement* (TCWR) for leveraging both context and alignment to replace phrases in source and target sequences. In Section 3.4.2, we propose a new data augmentation algorithm based on this replacement technique. In Section 3.4.3, we describe the architectures used to parameterize the models.

3.4.1 Translation-Counterfactual Word Replacement

Consider the sequence pair $(\mathcal{X}, \mathcal{Y})$ in Figure 3.2. A translation language model that learns $p(Y_j|\mathcal{X}, \mathcal{Y}_{-j})$ (where $\mathcal{Y}_{-j}$ indicates the sequence $\mathcal{Y}$ with Y_j removed) for all $j \in \{1, ..., |\mathcal{Y}|\}$ induces a causal graph on this pair. Specifically it is fully connected, in the following way: (a) all phrases in $\mathcal{X}$ cause all phrases in $\mathcal{Y}$, (b) all phrases in $\mathcal{Y}$ cause all other phrases in $\mathcal{Y}$ (these connections are signified by the wide gray arrow in Figure 3.2). Additionally, there are unobserved variables $\mathbf{G}_i^{\mathcal{X}}, \mathbf{G}_i^{\mathcal{Y}}$ that cause each individual phrase (more on this below). We choose this fully connected structure to take contexts of each phrase into account. Note that this graph is cyclic, yet the counterfactual distribution we care about is identifiable given the posterior of the unobserved variables (which we describe below) and the known equations of the causal model (i.e., the translation language model).

Consider that we have an alignment between $\mathcal{X}$ and $\mathcal{Y}$, which singles-out the causal effects shown with black arrows in Figure 3.2. Our idea is to derive a new sequence pair $(\hat{\mathcal{X}}, \hat{\mathcal{Y}})$ by computing a counterfactual. We propose to calculate the counterfactual corresponding to a single alignment, i.e. a path-specific counterfactual: "What would Y_j have looked like, had $X_i = \hat{x}_i$ instead of x_i, given that Y_j is aligned to X_i, and *all other phrases* $\mathcal{X}_{-i}, \mathcal{Y}_{-j}$ *had been held constant*?". This allows us to consider 1. **Context**: By holding all other phrases constant we control for the specific context around the changed phrases $\hat{x}_i, \hat{y}_j$[1]; 2. **Alignment**: The derived counterfactual is based on the direct effect of X_i on Y_j, where this singled-out link is identified from an alignment.

We now outline the three steps to calculate the counterfactual. The example in Figure 3.2 is used for illustration. The goal is to sample from the following counterfactual distribution: $p(Y_{2,\{X_3 \leftarrow \hat{x}_3, \mathcal{X}_{-3} \leftarrow \mathbf{x}_{-3}, \mathcal{Y}_{-2} \leftarrow \mathbf{y}_{-2}\}} | \mathcal{X}, \mathcal{Y})$ with the translation language model, which describes "What would Y_2 have looked like, had $X_3 = \hat{x}_3$ instead of x_3, given that Y_2 is aligned to X_3, and *all other phrases* $\mathcal{X}_{-3}, \mathcal{Y}_{-2}$ *had been held constant*?". For ease of illustration, we assume both X_3 and Y_2 contain one token after Byte Pair Encoding (BPE) segmentation (Sennrich et al., 2016c). In Section 3.4.3, we explain how we use a sequence-to-sequence model to generate phrases containing multiple tokens after segmentation.

1. Abduction. The goal of the abduction step is to estimate any unobserved variables that impact the counterfactual. As our translation language model specifies a categorical distribution $p(Y_2 | \mathcal{X}, \mathcal{Y}_{-2})$, this unobserved randomness, i.e. the prior of $\mathbf{G}_2^{\mathcal{Y}}$, takes the form of a Gumbel random vector. This is due to the fact that random sampling from a categorical distribution can be done via a procedure called the *Gumbel-Max Trick* (Maddison et al., 2014).

[1]Further, the posterior of unobserved random variables $\mathbf{G}_j^{\mathcal{Y}}$ given $\mathcal{X}, \mathcal{Y}$ will encode additional context w.r.t. the original sequence pair $\mathcal{X}, \mathcal{Y}$.

Definition 3.4.1 (Gumbel-Max Trick) *Two steps are required to sample from a categorical distribution $p(Y)$ with K categories: 1. Sample $g_1, \ldots, g_K \sim$ Gumbel$(0, 1)$. Each g_k can be computed as $g_k = -\log(-\log u_k)$ where $u_k \sim$ Uniform$(0, 1)$; 2. Compute $y = \arg\max_{k=1,\ldots,K} \log p(Y = k) + g_k$.*

As such, sampling from the translation language model $p(Y_2|\mathcal{X}, \mathcal{Y}_{-2})$ with vocabulary size $|V|$ can be written as follows,

$$y_2 = \arg\max_{k=1,\ldots,|V|} \log p(Y_2 = k|\mathcal{X}, \mathcal{Y}_{-2}) + g_k,$$
$$s.t.\ g_k \sim \text{Gumbel}(0, 1).$$

The abduction step samples from the posterior distribution over these Gumbel random variables, given the observed pair $(\mathcal{X}, \mathcal{Y})$, i.e., $p(\mathbf{G}_2^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$. Fortunately, sampling from the posterior is straightforward to do in two steps (Maddison et al., 2014; Maddison and Tarlow, 2017): 1. Let $y_2 = k^*$. Sample $\hat{g}_{k^*} \sim$ Gumbel$(0, 1)$; 2. For the remaining k, compute the probabilities from the model $p(Y_2 = k|\mathcal{X}, \mathcal{Y}_{-2})$, and sample from the distribution $\hat{g}_k \sim$ Gumbel$(\log p(Y_2 = k|\mathcal{X}, \mathcal{Y}_{-2}), 1)$ truncated within the range $(-\infty, \hat{g}_{k^*})$. The resulting samples $[\hat{g}_1, \ldots, \hat{g}_{|V|}]$ are from the posterior $p(\mathbf{G}_2^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$. We describe these steps in more detail in Algorithm 1.

2. Action. In this step, we replace a phrase x_3 in the source sequence with a substitute phrase $\hat{x}_3$. While any replacement leads to a valid counterfactual, we propose to sample $\hat{x}_3$ as

$$\hat{x}_3 \sim p(X_3|\mathcal{X}_{-3}),$$

where $p(X_3|\mathcal{X}_{-3})$ is given by a trained masked language model. By sampling from a distribution conditioned on the remaining phrases in $\mathcal{X}$, we sample a realistic

Algorithm 1: Gumbel Posterior Sampling

Input : The observed phrase $y_j = k^*$
Probabilities $p(Y_j = k|\mathcal{X}, \mathcal{Y}_{-j})$
for $k = 1, \ldots, |V|$
Output: Sampled Gumbel values $\hat{\mathbf{g}} \sim p(\mathbf{G}_j^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$
Sample $\hat{g}_{k^*} \sim \text{Gumbel}(0, 1)$
for $k \leftarrow 1$ **to** $|V|$ **do**
  if $k \neq k^*$ **then**
    `// Sample from truncated Gumbel`
    Sample $h_k \sim \text{Gumbel}(0, 1)$
    $u_k = h_k + \log p(Y_j = k|\mathcal{X}, \mathcal{Y}_{-j})$
    $\hat{g}_k = -\log(e^{-u_k} + e^{-\hat{g}_{k^*}})$

replacement word for X_3. In Figure 3.2, we sample $\hat{x}_3 =$ 'apple' in place of $x_3 =$ 'house'.

3. Prediction. Given the posterior samples $[\hat{g}_1, \ldots, \hat{g}_{|V|}] \sim p(\mathbf{G}_2^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$ and the substitute phrase $\hat{x}_3$, we can compute the counterfactual distribution of interest $p(Y_{2,\{X_3 \leftarrow \hat{x}_3, \mathcal{X}_{-3} \leftarrow \mathbf{x}_{-3}, \mathcal{Y}_{-2} \leftarrow \mathbf{y}_{-2}\}}|\mathcal{X}, \mathcal{Y})$, via the trained translation language model. We do so by computing:

$$\hat{y}_2 = \arg\max_{k=1,\ldots,|V|} \log p(Y_2 = k|\hat{x}_3, \mathcal{X}_{-3}, \mathcal{Y}_{-2}) + \hat{g}_k. \tag{3.1}$$

The sample $\hat{y}_2$ from the counterfactual distribution is based on the direct effect of X_3 on Y_2. We remark that the causal model we consider was first introduced by Oberst and Sontag (2019) and called the *Gumbel-Max Structural Causal Model*. Our insight here is that counterfactuals from this model can be used as an effective data augmentation method for machine translation.

Algorithm 2: Data Augmentation

Input : $(\mathcal{X}, \mathcal{Y})$: A sequence pair
c: A sampling probability
h: An alignment mapping
Output: A new pair $(\hat{\mathcal{X}}, \hat{\mathcal{Y}})$
$\hat{\mathcal{X}}, \hat{\mathcal{Y}} = \mathcal{X}, \mathcal{Y}$
for $i \leftarrow 1$ **to** $|\mathcal{X}|$ **do**
  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < c$ **then**
    $\hat{\mathcal{X}}, \hat{\mathcal{Y}} \leftarrow$`replace`$(\hat{\mathcal{X}}, \hat{\mathcal{Y}}, i, h)$

Function `replace`$(\mathcal{X}, \mathcal{Y}, i, h)$
  Get aligned index $j = h(i)$
  $\hat{\mathbf{g}} \sim p(\mathbf{G}_j^{\mathcal{Y}} | \mathcal{X}, \mathcal{Y})$
  $\hat{x}_i \sim p(X_i | \mathcal{X}_{-i})$
  $\hat{y}_j \sim p(Y_{j,\{X_i \leftarrow \hat{x}_i, \mathcal{X}_{-i} \leftarrow \mathbf{x}_{-i}, \mathcal{Y}_{-j} \leftarrow \mathbf{y}_{-j}\}} | \mathcal{X}, \mathcal{Y})$
  Set the i-th phrase of $\mathcal{X}$ to $\hat{x}_i$
  Set the j-th phrase of $\mathcal{Y}$ to $\hat{y}_j$
  return $\mathcal{X}$, $\mathcal{Y}$

3.4.2 Data Augmentation

Given the above procedure to replace phrases, we propose a new data augmentation method, shown in Algorithm 2. The algorithm takes an input pair of sequences $(\mathcal{X}, \mathcal{Y})$ and loops through every phrase $X_i \in \mathcal{X}$. At each iteration with probability c it replaces the phrase pair (x_i, y_j) with $(\hat{x}_i, \hat{y}_j)$.

3.4.3 Training Language Models

We introduce a special [MASK] token (Devlin et al., 2019) to represent a removed phrase for parameterizing both $p(X_i | \mathcal{X}_{-i})$ and $p(Y_j | \mathcal{X}, \mathcal{Y}_{-j})$ as:

$$p_{\theta_1}(X_i | X_1, ..., X_i = \text{[MASK]}, ..., X_{|X|}) \tag{3.2}$$

and

$$p_{\theta_2}(Y_j|\mathcal{X}, Y_1, ..., Y_j = \text{[MASK]}, ..., Y_{|Y|}). \tag{3.3}$$

Eq. 3.2 only requires monolingual datasets, which are abundant. On the other hand, Eq. 3.3 requires parallel corpora to train. We parameterize Eq. 3.3 using a variant of the translation language model (Conneau and Lample, 2019). The main difference is that only phrases in target sequences are masked, whereas Lample and Conneau (2019) mask both source and target tokens, with the goal of learning bilingual relations. Another difference is that a phrase with consecutive tokens is masked, while masked tokens in Lample and Conneau (2019) are not necessarily consecutive.

To better tackle unknown and rare tokens, we adopt BPE to segment phrases into tokens. As the number of tokens is undetermined during generation, we use a sequence-to-sequence transformer model (Vaswani et al., 2017) to encode inputs and decode tokens one by one until a special end-of-sequence symbol is encountered.

More specifically, given a sequence of N tokens $(t_1, ..., t_N)$, the sequence contains a special [MASK] token signifying a masked phrase. Each token t_i is first projected into its embedding $\mathbf{e}_{t_i}$, which is a sum of its token embedding, position embedding, and language embedding, inspired by XLM (Conneau and Lample, 2019). Then, a transformer encoder is applied to encode the tokens into their hidden representations $\mathbf{H} \in \mathbb{R}^{N \times o}$ (where o denotes the hidden size), i.e. $\mathbf{H} = \text{Encoder}(\mathbf{e}_{t_1}, ..., \mathbf{e}_{t_N})$. The hidden representation of [MASK], $\mathbf{h}_{\text{[MASK]}} \in \mathbb{R}^o$, is fed into a transformer decoder to predict the tokens of the masked phrase.

We learn our models $p_{\theta_1}(X_i|\mathcal{X}_{-i})$, $p_{\theta_2}(Y_j|\mathcal{X}, \mathcal{Y}_{-j})$ by maximizing the following

Dataset	# Sequences	# Words	# Chars
News Commentary	0.46M	10.05M	63.96M
News Crawl 2010	6.8M	0.14B	0.83B

Table 3.1: The statistics of the monolingual datasets.

Dataset	# Train	# Dev	# Test
WMT'18 En-Tr	206K	3007	3000
WMT'17 En-De	5.85M	2,999	3,004
IWSLT'15 En-Vi	133K	1,553	1,268
WMT'19 Robust En-Fr	36,058	852	1,401
Europarl-v7 En-Fr	2M	-	-

Table 3.2: The statistics of the parallel corpora.

objectives:

$$\mathbb{E}_{\mathcal{X}\sim\mathcal{D}}\left[\mathbb{E}_{i\sim\text{Uniform}(1,\ldots,|\mathcal{X}|)}[\log p_{\theta_1}(X_i|\mathcal{X}_{-i})]\right]$$

and

$$\mathbb{E}_{(\mathcal{X},\mathcal{Y})\sim\mathcal{S}}\left[\mathbb{E}_{j\sim\text{Uniform}(1,\ldots,|\mathcal{Y}|)}[\log p_{\theta_2}(Y_j|\mathcal{X},\mathcal{Y}_{-j})]\right].$$

Here $\mathcal{D}$ is a monolingual dataset and $\mathcal{S}$ is a parallel corpus.

3.5 Experiments

We now describe the improvements with the data augmentation based on TCWR.

3.5.1 Language Model Details

We use the monolingual training data, including News Commentary and News Crawl 2010, provided by WMT'18, for Eq. 3.2, while the training set of each language pair is used for Eq. 3.3. The statistics of the monolingual and parallel corpora are summarized in Table 3.1 and 3.2, respectively.

To reduce memory overhead, we train a shared language model for Eq. 3.2 and 3.3, i.e. θ_1 and θ_2 are tied. A language model is trained for each language pair to avoid performing multilingual NMT for a fair comparison with baselines, as jointly training a single model for several language pairs has been shown to be effective for both low-resource and high-resource languages (Aharoni et al., 2019). Therefore, we pre-train four models for En-Tr, En-De, En-Vi and En-Fr, respectively.

The encoder and decoder are composed of six layers. The encoder is initialized with XLM (Conneau and Lample, 2019) pre-trained with the masked language model, while the decoder is randomly initialized. The input-output embeddings are tied for reducing the size of the model (Press and Wolf, 2017). To achieve faster convergence, we apply PreNorm (Nguyen and Salazar, 2019) for getting rid of the warm-up stage of transformer. The learning rate is set to 1e-5 and is linearly decayed with more training steps. The hidden size o is set to 1024. Same as BERT, the maximum sequence size is set to 512. We use LAMB (You et al., 2020) as the optimizer. GELU (Hendrycks and Gimpel, 2016) is used as the activation function. 16 sequences are used at each pre-training step. We train the masked language model for 50% of the time and the left time is used for training the translation language model.

After pre-training, we use the pre-trained models to perform data augmentation on training data. Then, the augmented data are combined with training data for training NMT models.

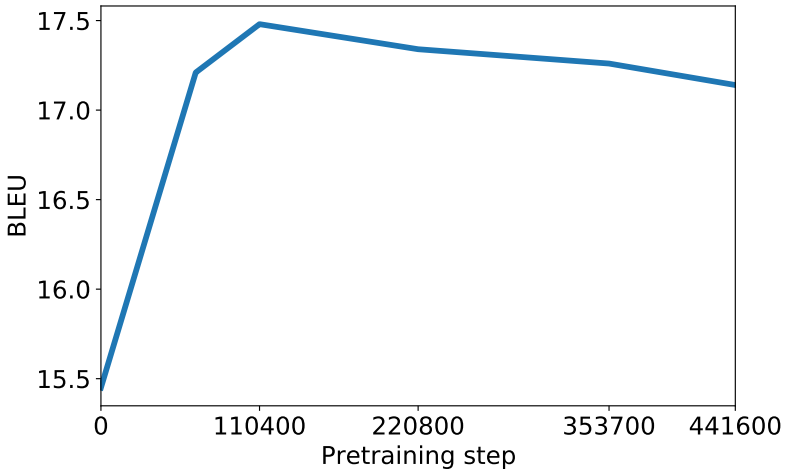


Figure 3.3: BLEU scores on the development set of the WMT'18 English → Turkish task with different pre-training steps.

3.5.2 NMT Model Details

We use fairseq[2] to implement the NMT models. The vocabularty size is 37K. Six encoder and decoder layers are applied. The hidden size is set to 1024. 16 self-attention heads are employed. We use Adam as the optimizer. The learning rate is initially set to 1e-7 and is gradually increased to 5e-4 with 4K warm-up steps, before applying linear decay. Dropout is set to 0.3. Label smoothing with the smoothing factor 0.1 is used. For decoding, we use beam search, and the beam size is set to 12. SacreBLEU (Post, 2018) is used as the metric.

3.5.3 Sensitivity Study

Pre-training steps. We study the effect of pre-training steps on machine translation quality. We use the language models at different pre-training steps and evaluate these models on the development set of the WMT'18 English → Turkish task. The results are shown in Figure 3.3. The BLEU score improves with more

[2] https://github.com/pytorch/fairseq

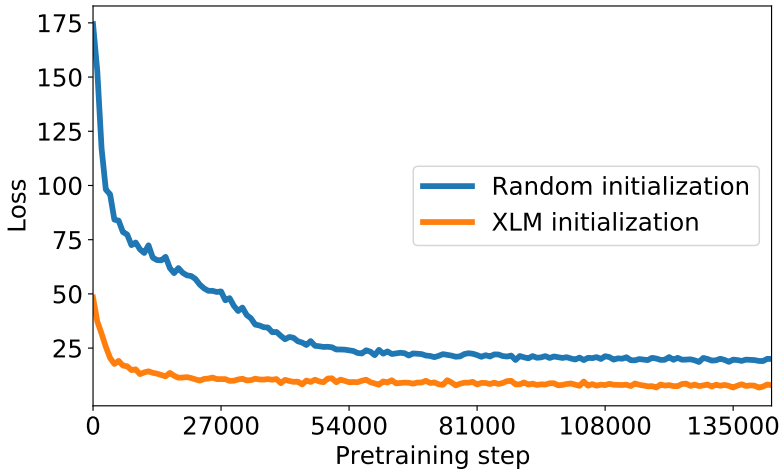


Figure 3.4: Learning curves of training language models for the WMT'18 English $\rightarrow$ Turkish task with either the random initialization or the XLM initialization.

pre-training steps and peaks at around 110K steps. We do not observe better performance with more pre-training steps, as the models become more overfitted on the training sets.

Effect of the XLM initialization. We plot the learning curves of the language model for En-Tr with/without XLM initialization. As shown in Figure 3.4, the model with the XLM initialization converges faster and better compared to the model with the random initialization. As XLM is trained using the masked language model objective on large-scale monolingual data, we draw the conclusion that large-scale pre-training can improve downstream language model pre-training tasks. We further evaluate the models on the development set of the WMT'18 English $\rightarrow$ Turkish task. The model with the XLM initialization also performs better (17.49 BLEU) compared to its counterpart (16.68 BLEU). Thus, the model with the XLM initialization can also generate better data for improving NMT.

Sampling probability. As shown in Figure 3.5, we vary the sampling probability in Algorithm 2 and evaluate on the development set of the WMT'18 English

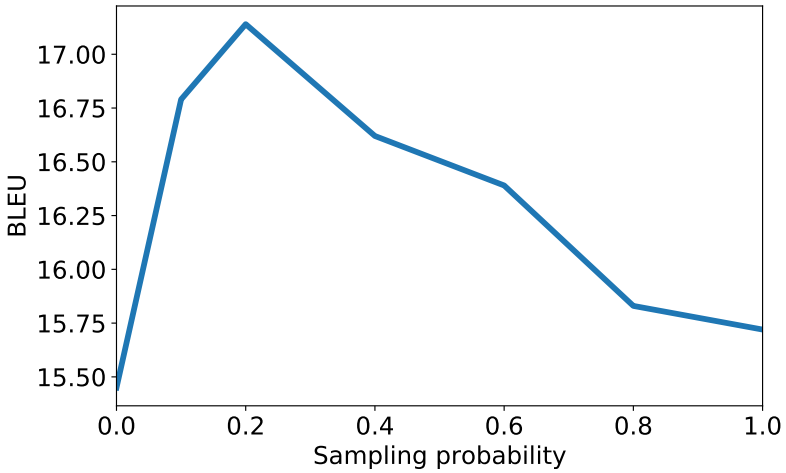


Figure 3.5: BLEU scores on the development set of the WMT'18 English → Turkish task with different sampling probabilities.

→ Turkish task. We observe that the BLEU score is maximized with sampling probability 0.2. The BLEU scores decrease with larger sampling probabilities.

3.5.4 Ablation Study

Method	**En → Tr**
TCWR	17.49
-Source	16.47
-Target	16.29
-Alignment	16.59
-Gumbel	16.85

Table 3.3: The BLEU scores on the development set of the WMT'18 English → Turkish task with source context, target context, alignment, and Gumbel ablation.

We ablate the source context, target context and alignment to validate the effectiveness of these components. We randomly choose a phrase with uniform distribution to replace the original phrase for ablating source and target contexts. For ablating alignment, we randomly choose a position in the target sequence instead

Method	En → Tr	En → De	En → Vi
Baseline	15.35	27.54	31.66
+WordDropout	15.4	27.81	31.81
+SwitchOut	15.52	27.92	31.83
+SCDA	15.72	28.05	31.72
+TDA	15.69	28.16	31.79
+BPEDropout	15.95	28.29	33.59
+DADA	16.14	29.03	32.15
+TCWR	**17.38**	**29.37**	**33.76**

Table 3.4: The BLEU scores on the testing sets of En-Tr, En-De and En-Vi. The baseline method denotes training without any data augmentation.

of following the alignment given by pialign. We also study removing $\hat{g}_k$ in Eq. 3.1. Since $\hat{g}_k$ comes from the abduction step, which encodes the information from the original pair $(\mathcal{X}, \mathcal{Y})$, Eq. 3.1 encourages the model to sample a new pair that is similar to the original pair. Therefore, the model without $\hat{g}_k$ collapses to a probabilistic approach that directly samples phrases from the translation language model, disregarding the information from the original pair.

The results are shown in Table 3.3. We observe that ablating the source context, target context and alignment are negative for translation quality, demonstrating the necessity of considering all these components for data augmentation. The result of ablating $\hat{g}_k$ shows the effectiveness of incorporating the original information from $(\mathcal{X}, \mathcal{Y})$.

3.5.5 Translation Result

We evaluate the algorithms on WMT'17 English → German (En-De), WMT'18 English → Turkish (En-Tr) and IWSLT'15 English → Vietnamese (En-Vi). As shown in Table 3.2, En-Tr and En-Vi are two low-resource language pairs, while En-De is a high-resource language pair.

Method	En → Tr	En → De	En → Vi
Baseline	15.35	27.54	31.66
+TCWR	17.38	29.37	33.76
+BT	19.24	29.19	33.38
+BT +TCWR	**20.19**	**30.26**	**35.72**

Table 3.5: The BLEU scores on the testing sets with backtranslation and TCWR.

For En-Tr, we use *newstest17* for validation and *newstest18* for testing. For En-De, we use *newstest16* for validation and *newstest17* for testing. For En-Vi, we use the TED *tst2012* for validation and the TED *tst2013* for testing.

We compare TCWR with six baselines, WordDropout, BPEDropout, SwitchOut, SCDA, TDA and DADA. For WordDropout and BPEDropout, we perform a range search on its dropout probability from 0 to 1 and select the best one on development sets. Similarly, we choose the temperature with the highest score on development sets for SwitchOut. For SCDA, we search the replacing probability and set it to 0.15. We follow the official implementation[3] of TDA. We reuse the hyperparameters from Cheng et al. (2019) for DADA.

The results on three language pairs are shown in Table 3.4. Compared to the baseline with no data augmentation, TCWR yields improvements of 2.03, 1.63 and 1.79 BLEU for En-Tr, En-De and En-Vi, respectively. TCWR also outperforms the other augmentation methods, which further confirms the effectiveness of considering source context, target context, and alignment for NMT data augmentation. Besides, these results demonstrate that TCWR brings consistent improvements to both low-resource and high-resource language pairs.

3.5.6 Backtranslation Result

As backtranslation is a widely-used data augmentation method by utilizing monolingual data to generate new parallel pairs, we show how TCWR can be used with backtranslation. To perform backtranslation, we use the monolingual sequences from News Crawl 2017, News Crawl 2010 and VNTC[4] for En-Tr, En-De and En-Vi, respectively. Then we perform data augmentation on both training data and backtranslated data. As shown in Table 3.5, TCWR improves upon backtranslation, demonstrating that TCWR and backtranslation are not mutually exclusive, and TCWR can enhance the performance of backtranslation.

Method	**En → Fr**
Baseline	26.0
+WordDropout	26.52
+SwitchOut	26.61
+SCDA	26.85
+BPEDropout	27.08
+TDA	27.11
+DADA	28.14
+TCWR	**28.92**

Table 3.6: The BLEU scores on the WMT'19 English → French robustness task.

3.5.7 Machine Translation Robustness

Noisy or non-standard input text (e.g. text with spelling errors and code switching) can cause significant degradation in most NMT systems. We use the WMT'19 English → French robustness dataset for evaluating translation robustness. As the parallel pairs are scarce for this task, we combine its training data with the English → French pairs from Europarl-v7. The models are validated on the development

[3] https://github.com/marziehf/DataAugmentationNMT
[4] https://github.com/duyvuleo/VNTC

En:	Kosovo is taking a hard look at its privatisation process in light of recurring [complaints / **problems**].
Tr:	Kosova, tekrar eden [şikayetler / **sorunlar**] ışığında özelleştirme sürecini incelemeye alıyor.
En:	A decade later, we see that the [economy / **system**] is terribly unstructured.
Tr:	On yıl sonra, [ekonominin / **sistemin**] yapısının çok kötü bozulduğunu görüyoruz.
En:	Report : most [SEE / **independent**] countries advance in economic freedom.
Tr:	Rapor : [GDA / **bagımsız**] ülkelerinin çoğu ekonomik özgürlükte ilerliyor.

Table 3.7: A case study on TCWR, where augmented positions are marked as [original / **substituted**].

set of the MTNT dataset and tested on the released test set of the WMT'19 robustness task. The results are shown in Table 3.6. We observe that TCWR outperforms the baseline without any data augmentation or with the other methods. If we regard the task as adapting from the source dataset with clean text (Europarl-v7) to the target dataset with noisy text (WMT'19 robustness dataset), TCWR helps this adaptation via enlarging training examples with language models trained using noisy and non-standard text. We thereby conclude that TCWR can improve NMT robustness.

3.5.8 Case Study

As shown in Table 3.7, we perform a case study of TCWR. We observe that TCWR can reasonably substitute words in source sequences based on contexts and modify corresponding target words, which demonstrates the benefits of considering both context and alignment for augmentation.

3.6 Summary

In this chapter, we studied utilizing local context for augmenting a Neural Machine Translation (NMT) dataset. We proposed a data augmentation method, which introduces a causal inductive bias that takes both context and alignment into account. Our method uses language models to answer the counterfactual "What

would Y_j have looked like, had $X_i = \hat{x}_i$ instead of x_i, given that Y_j is aligned to X_i, and *all other phrases* $\mathcal{X}_{-i}, \mathcal{Y}_{-j}$ *had been held constant*?". The method was shown to improve the performance of translation, backtranslation and translation robustness on four NMT benchmarks. In the next chapter, we introduce retrieving global context for improving language modelling.

4 | Retrieving Global Context

4.1 Introduction

In the last chapter, we have utilized local context to augment machine translation datasets. In this chapter, we study using global context for natural language processing. Given a training corpus that consists of textual sequences $\{\mathcal{S}_1, ..., \mathcal{S}_M\}$, each entity e may appear in different sequences. We refer to the sequences that contain the entity e as the global context of e as it provides "global" information about e in the training corpus.

Relation triples (head entity, relation, tail entity) form the basis of knowledge bases, empowering a wide range of applications such as question answering (Yasunaga et al., 2021), machine reading (Yang and Mitchell, 2019), and reasoning (Minervini et al., 2020). In this work we extract relation triples from the global context of each entity to provide propositional information about entities for language modelling. An autoregressive language model defines the probability of a sequence of tokens. Our motivation is based on the observation that existing autoregressive language models (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020) do not have explicit representations for propositional information (e.g., who did what to whom) and rely on it being implicitly encoded in their parameters (Liu et al., 2019a; Petroni et al., 2019; Wang et al., 2020a). This encoding mechanism makes it difficult to interpret what the autoregressive language models know and often leads to generating illogical and contradictory contents. For example, Logan et al. (2019) observe that existing autoregressive language models rely heavily on word correlation and fall short of logical reasoning. This causes the model to hallucinate—e.g. that Barack Obama's wife is Hillary Clinton based on the high co-occurrence of the two entities. In another example, Lake and

Murphy (2020) notice that GPT-2 (Radford et al., 2019) states that unicorns have four horns, directly after speaking that unicorns have one horn.

In this work we argue that utilizing global context is a viable approach to remedy these problems. Based on global contexts of entities, we perform open information extraction (OpenIE) on these contexts to get relation triples. We construct a knowledge graph with these relation triples to represent the interactions between entities. Further, we design a memory-augmented architecture to explicitly condition an autoregressive language model on the knowledge graph constructed with global context. Given an observed sequence, the goal of language modelling is to generate a continuation of this observed sequence. We first run an entity tagger to identify entities in the observed sequence. We then use tf-idf (Ramos, 2003) to select salient entities. We retrieve relations (from the knowledge base constructed with global context) for the selected entities and design a gating function that allows the language model to adaptively combine information from retrieved relation triples and the observed sequence to predict the next token. As a result, our model is able to move beyond simple co-occurrence statistics and generate text that is more grounded to global context. From a cognitive science perspective, we can consider the neural language model to be an instance of System 1 which performs fast inference and the symbolic relational memory as a world model to support slow and logical reasoning of System 2 (Kahneman, 2011).[1] We hypothesise that our relational memory can improve performance and coherence of an autoregressive language model.

Our main contributions are as follows:

- We evaluate the model on three English language modelling datasets. We show that our model outperforms a strong transformer-XL baseline (Dai

[1]This view is also advocated in a parallel work by Nye et al. (2021) which presents a model for story generation and instruction following.

et al., 2019) on both word-level (WikiText-103 and WMT19) and character-level (enwik8) language modelling in terms of perplexity and bits per character respectively (§4.4.3).

- We conduct comprehensive ablation and design choice studies to understand contributions of different components of our models (§4.5.1).
- We measure coherence with human evaluation and two automatic metrics (knowledge perplexity and knowledge F_1) and demonstrate that relational memory improves coherence (§4.5.2).
- We study the relationship between our method and a typical memory augmented language model which stores word tokens in its memory (Yogatama et al., 2021). We show that relational memory is complementary to token-based memory and combining them improves performance further (§4.4.3).
- We perform qualitative analysis by examining gate values and retrieved relations. In line with our main motivation, we find that the relational memory is particularly useful for predicting entities. Further, we demonstrate that such explicit propositional representations allow causal interventions and increase interpretability of language models (§4.5.3).

In the rest of this chapter, we first review related work (§4.2). Next, we elaborate on the model including the transformer-XL baseline and our relational memory constructed with global context (§4.3). Finally, we evaluate the effectiveness of the relational memory (§4.4 and §4.5).

4.2 Related Work

Knowledge-enhanced architectures. Injecting symbolic knowledge to machine learning models is widely-adopted to improve the performance of natural lan-

guage understanding (Annervaz et al., 2018; Ostendorff et al., 2019), question answering (Zhang et al., 2018; Huang et al., 2019; Hixon et al., 2015), dialogue systems (Zhou et al., 2018; Moon et al., 2019; Guo et al., 2018; Liu et al., 2021b) and recommendation systems (Zhang et al., 2016; Wang et al., 2018a, 2019b). Different from these models, we focus on using symbolic knowledge for language modelling. Existing language models are prone to generating illogical and contradictory contents. We believe that connecting language modelling and knowledge graphs is a promising direction to overcome the problem. Next we review previous knowledge-enhanced language models.

Knowledge-enhanced language models. Our model is closely related to previous work on grounding autoregressive language models with knowledge graphs (Ahn et al., 2016; Logan et al., 2019; Hayashi et al., 2020; Wang et al., 2021a). However, these models rely on complex and adhoc preprocessing or rules to link text with knowledge bases, e.g. Freebase and Wikidata. As a result, previous work is more aligned with conditional language modelling, e.g. graph-to-text generation $p(\mathbf{x}|\mathcal{G})$ in Wang et al. (2021a), which contrasts with unconditional language modeling $p(\mathbf{x})$ considered in this work. As the graph $\mathcal{G}$ is constructed with the *unseen* text $\mathbf{x}$, predicting $\mathbf{x}$ given $\mathcal{G}$ is easier due to this information leakage for Wang et al. (2021a). Also in Hayashi et al. (2020), topic entities are required for language modelling, which may not be available in most datasets, e.g. the news domain. We do not compare with these previous models due to the different settings. In contrast, we adopt OpenIE relations and use a tf-idf search to retrieve relation triples for connecting language models and knowledge graphs. In the experiments, we demonstrate the effectiveness of our approach on three datasets, WikiText-103, WMT19 and enwik8.

There are language models incorporating entity information, such as entity coref-

erence annotations (Ji et al., 2017; Clark et al., 2018), surface forms of entities (Kiddon et al., 2016; Yang et al., 2017; Cao et al., 2021), entity types (Parvez et al., 2018; Wang et al., 2018b) and entity descriptions (Bahdanau et al., 2017). Different from these models, we augment language models with a relational memory consisting of relation triples. We demonstrate the effectiveness of using relation triples by ablating tail entities and relations in §4.5.1.

Knowledge-enhanced pretraining. Using knowledge information for pretraining language models (Peters et al., 2019; Sun et al., 2019; Liu et al., 2020; Guu et al., 2020; Wang et al., 2021b; Agarwal et al., 2021; Verga et al., 2021) has recently grown in popularity and has achieved substantial improvements on knowledge driven tasks such as question answering and named entity recognition. Instead of using knowledge information for improving downstream knowledge-driven tasks, we focus on using knowledge information for improving the generation capability of the language model itself.

Retrieval-augmented models. Retrieval augmented models are widely used in open-domain question answering (Chen et al., 2017; Lewis et al., 2020b; de Masson d'Autume et al., 2019; Izacard and Grave, 2021), dialogue (Dinan et al., 2019; Fan et al., 2021; Thulke et al., 2021) and machine translation (Bapna and Firat, 2019; Khandelwal et al., 2020a). We focus on retrieval augmentation for language modelling (Merity et al., 2017; Grave et al., 2016; Khandelwal et al., 2020b; Yogatama et al., 2021). These algorithms are specifically tailored for language modelling, where related tokens are retrieved to help predict the next token. In this work, we move beyond token augmentation and show the benefits of retrieving relation triples. We also demonstrate that our model is complementary to a token augmentation model, SPALM (Yogatama et al., 2021) in the experiments.

4.3 Model

An autoregressive language model defines the probability of a sequence of tokens $p(\boldsymbol{x}) = p(x_1, \ldots, x_T)$. It is common to factorise this joint probability as a product of conditional probabilities with the chain rule (Jelinek, 1980; Bengio et al., 2003):

$$p(x_1, ..., x_T) = \prod_{t=1}^{T} p(x_t | x_0, \ldots, x_{t-1}), \tag{4.1}$$

where x_0 is a special start token.

Our language model is based on transformer-XL (§4.3.1) which is augmented with a relational memory (§4.3.2). We discuss them in detail below.

4.3.1 Transformer-XL

We use transformer-XL (Dai et al., 2019)—which is based on transformer (Vaswani et al., 2017)—to parametrise the conditional probabilities in Eq. 4.1. Transformer stacks multiple self-attention layers to obtain contextualised representations.

Language modelling datasets usually consist of articles of different lengths. It is impractical to apply transformer to encode long articles, as its computational complexity is quadratic in the sequence length. In practice, each article is usually truncated into fixed-length text segments $\{x_{t-N+1}, ..., x_t\}$ of length N to train and evaluate the model. However, this approximation prevents transformer from capturing long-term dependency beyond text segments. Transformer-XL reuses hidden states from previous text segments to extend the context window.

More specifically, denote the hidden state of x_t at layer ℓ as $\mathbf{h}_t^\ell$. Given a text segment $\{x_{t-N+1}, \ldots, x_t\}$ and its extended context $\{x_{t-N-M+1}, \ldots, x_{t-N}\}$ of length M, both the hidden states of the text segment $\{\mathbf{h}_{t-N+1}^\ell, \ldots, \mathbf{h}_t^\ell\}$ and the hidden

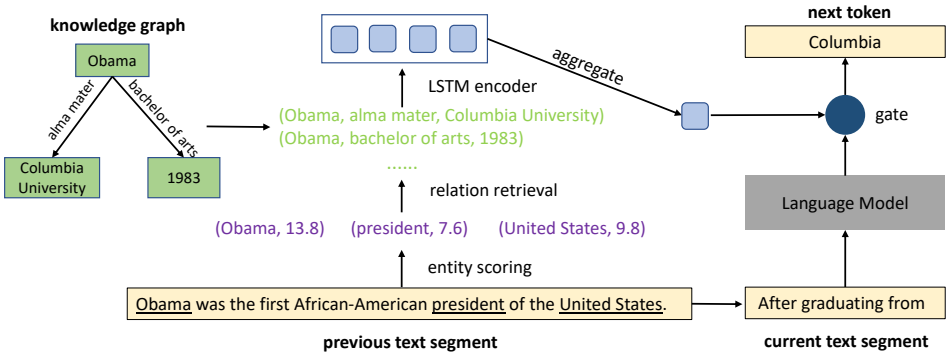


Figure 4.1: We identify salient entities in the previous text segment and extract relations to build our relational memory. We encode each relation with an LSTM encoder, aggregate the resulting representations into a vector, and use a gate mechanism that allows our language model to adaptively take advantage of relational information for predicting the next token.

states of the extended context $\{\mathbf{h}^{\ell}_{t-N-M+1}, \ldots, \mathbf{h}^{\ell}_{t-N}\}$ are used. When performing self-attention, each token in the text segment can attend to the preceding tokens in the text segment and all the tokens in the extended context, enabling longer-term dependency compared to a vanilla transformer. Importantly, transformer-XL does not backpropagate through the hidden states of the extended context during training (by adding stop gradient operators to all the hidden states in the extended context).

4.3.2 Relational Memory

In this section, we first introduce how we obtain relation triples from global context using OpenIE (§4.3.2.1). We then use tf-idf to score entities in the observed context and retrieve relation triples related to these entities (§4.3.2.2) to construct relational memory. Finally, we show an integrated architecture that allows transformer-XL to incorporate the relational memory for predicting the next token (§4.3.2.3). We show our architecture in Figure 4.1. The pseudocode of training or evaluating with the relational memory is demonstrated in Algorithm 3. In the

pseudocode, we use $\text{TRAIN}(\boldsymbol{x}_c, \mathcal{M})$ and $\text{EVAL}(\boldsymbol{x}_c, \mathcal{M}$ to refer to training with the cross entropy loss and evaluating (e.g. calculating perplexity) on the text segment $\boldsymbol{x}_c$ conditioned on the relational memory $\mathcal{M}$, respectively.

4.3.2.1 Open Information Extraction

To condition on propositional information, a key challenge is obtaining high-quality relation triples. There are several well-established knowledge bases, such as Freebase (Bollacker et al., 2007) and YAGO (Rebele et al., 2016). However, existing knowledge bases suffer from missing relations and often do not contain relation triples related to observed contexts in a target corpus, even though research on knowledge base completion has resulted in significant advances (Bordes et al., 2013; Trouillon et al., 2016; Zhang et al., 2019) .

In this work, we use OpenIE (Angeli et al., 2015; Etzioni et al., 2008) to obtain relation triples from global context. Since OpenIE directly extracts relation triples from each dataset $\mathcal{D}$, it provides a structured way to represent knowledge in $\mathcal{D}$.[2] Specifically, we perform OpenIE on the *training set* of $\mathcal{D}$. Given an entity e, we retrieve a set of relation triples $\mathcal{R}_e = \{r_1, ..., r_O\}$, where e is either the head entity or the tail entity in these relation triples. Conceptually, $\mathcal{R}_e$ consists of all the relation triples from the one-hop subgraph centred at the entity e in the knowledge graph constructed from $\mathcal{D}$. Therefore, $\mathcal{R}_e$ can provide "global" information about the entity.

Dynamic OpenIE. Dynamic OpenIE takes advantage of the autoregressive nature of language modelling, where text segments are sequentially processed. In addition to extracting relations from the training set of $\mathcal{D}$, we can also extract relations from *previously seen* text segments of our evaluation set. We refer to this

[2] We provide a comparison of using relations extracted from OpenIE and Freebase in §4.5.1.

Algorithm 3: Train/Eval w/ Relational Memory

```
Procedure train/eval split(S)
    for each article A in S do
        Initialise M to empty
        for each text segment x_c in A do
            if S is train set then
                TRAIN(x_c, M)
            else
                EVAL(x_c, M)
                Run dynamic OpenIE on x_c
            Perform relation retrieval with x_c
            Update M with retrieved triples
```

extraction mechanism as dynamic OpenIE. After a text segment $\{x_{t-N+1}, ..., x_t\}$ has been evaluated, e.g. after calculating perplexity on this text segment, we perform OpenIE on it to obtain new relation triples to be added to our knowledge graph. Note that we only perform OpenIE on previously seen text segments and do not use unseen text. We expect that the relation triples extracted from seen text segments are potentially useful for predicting the next tokens. This extraction mechanism will not violate the autoregressive nature of language modelling. Metrics such as perplexity and bits per character are calculated as usual. The idea of using seen text segments during evaluation to improve language modelling is related to dynamic evaluation (Krause et al., 2018, 2019). In dynamic evaluation, the model is adapted based on recent history during evaluation via gradient descent so that it can assign higher probabilities to re-occurring patterns. In contrast to dynamic evaluation, we do not update model parameters and only extract new relations from seen text segments to enrich our corpus-specific knowledge graph.

Mismatch between training and evaluation. As shown in Algorithm 3, since we do not use dynamic OpenIE during training due to its additional efficiency overhead (see speed comparison in §4.5.1), this results in a mismatch between

training and evaluation. We extract all the relation triples from the training set of each dataset $\mathcal{D}$ before training on D. As a result, during training we may retrieve relation triples extracted from *unseen* text of the training set when performing relation retrieval (§4.3.2.2). We do not suffer from this issue during evaluation, as we extract relations from previously seen text of our evaluation set. We believe this mismatch is minor given the superior performance of our model in the experiments.

4.3.2.2 Relation Retrieval

Given a knowledge graph (represented as a collection of triples), an ideal relational memory consists of a set of triples that are relevant to the observed context. There are many choices to measure the relatedness between the observed context and relation triples in our knowledge graph— e.g. based on keyword search or dense retrieval (Karpukhin et al., 2020; Guu et al., 2020; Yogatama et al., 2021).

In this work, we use keyword search due to its simplicity and leave methods based on dense retrieval to future work. Specifically, given the observed context, we perform entity recognition (Ratinov and Roth, 2009; Nadeau and Sekine, 2007) on this context and score the tagged entities with tf-idf (Ramos, 2003). The top-K scored entities (K is set to 5 in our experiments) are used to retrieve relations $\{\mathcal{R}_{e_1}, ..., \mathcal{R}_{e_K}\}$. These retrieved relations are used to construct the relational memory $\mathcal{M}$. Note that the entities are selected from the observed context, so that unseen text is not utilized. We limit the capacity of $\mathcal{M}$ to P. If the number of newly retrieved triples is larger than P, we randomly drop relations and only select P of them to be inserted into $\mathcal{M}$. Otherwise, the relational memory operates with a first-in-first-out principle. When $\mathcal{M}$ is full, older relations retrieved will be overwritten by newly retrieved relations. The relational memory is re-initialized to empty when an article ends.

As shown in Algorithm 3, since we update $\mathcal{M}$ only after processing an entire text segment, all the tokens in the same text segment will be conditioned on the same relational memory. This approach is more efficient compared to updating $\mathcal{M}$ each time a new entity is encountered and is more amenable for batch training.

4.3.2.3 Integration with Transformer-XL

We now show how we can integrate relational memory with transformer-XL. We refer to our model as RELATIONLM.

Relation triple encoding. We first discuss how we encode relation triples in the relational memory $\mathcal{M}$. We treat relation triples as text and serialise each relation triple into a sequence, e.g. (Barack Obama, president of, United States) is converted into a sequence "Barack Obama, president of, United States". This sequential representation can well capture the order of head entities and tail entities and is also adopted by KG-BERT (Yao et al., 2019) and Kepler (Wang et al., 2021b). Since each example in a batch corresponds to P retrieved relations, we obtain $B \cdot P$ relation sequences for each batch, where B and P denote batch size and relational memory length, respectively. In the order of hundreds of relation triples, this prevents us from using large models (e.g. a multi-layer transformer) to encode these sequences due to memory constraints. In our preliminary experiments, we compare LSTM (Hochreiter and Schmidhuber, 1997), GRU (Cho et al., 2014) and a one-layer transformer and find that LSTM performs marginally better. Therefore, for each relation triple r_p, we reuse the transformer-XL word embedding matrix $\mathbf{W}_e$ to map each token in the sequence to its embedding vector. We then run LSTM to encode the sequence and use the hidden representation of the last token as the relation representation $\mathbf{r}_p$.

There are other approaches to encode relation triples, e.g. embedding-based (Bor-

des et al., 2013; Trouillon et al., 2016) and graph-based (Schlichtkrull et al., 2018; Zhang and Chen, 2018) methods. We leave a comparison of these approaches to future work.

Integration. Given a text segment $\boldsymbol{x}_c = \{x_{t-N+1}, ..., x_t\}$, after L self-attention layers with transformer-XL, we obtain contextualized representations $\{\mathbf{h}^L_{t-N+1}, ..., \mathbf{h}^L_t\}$. At each timestep t, we use its hidden representation $\mathbf{h}^L_t$ as the query vector to attend over the P encoded contents of $\mathcal{M}$, i.e., $\{\mathbf{r}_1, ..., \mathbf{r}_P\}$. We use a standard scaled dot-product attention (Vaswani et al., 2017) to aggregate all triples into a single vector:

$$\mathbf{m}_t = \sum_{p=1}^{P} \frac{\exp(\mathbf{h}^L_t \cdot \mathbf{r}_p / \sqrt{d})}{\sum_{j=1}^{P} \exp(\mathbf{h}^L_t \cdot \mathbf{r}_j / \sqrt{d})} \mathbf{r}_p,$$

where d denotes the hidden size of our transformer-XL. Finally, we combine $\mathbf{m}_t$ and transformer-XL representation $\mathbf{h}^L_t$ via a gate:

$$\begin{aligned}
\mathbf{g}_t &= \sigma(\mathbf{W}_g[\mathbf{h}^L_t, \mathbf{m}_t]) \\
\mathbf{z}_t &= \mathbf{g}_t \odot \mathbf{h}^L_t + (1 - \mathbf{g}_t) \odot \mathbf{m}_t \\
p(x_{t+1} \mid \boldsymbol{x}_{\leq t}) &= \text{softmax}(\mathbf{W}_e \mathbf{z}_t),
\end{aligned}$$

where σ is the sigmoid function, $[,]$ denotes concatenation of two vectors, $\odot$ is element-wise multiplication, and $\mathbf{W}_e$ is the embedding matrix shared by both input and output embeddings (Inan et al., 2016). The only new parameters introduced by our method are an LSTM relation encoder and the gate matrix $\mathbf{W}_g$. This gating mechanism allows our model to adaptively take advantage of relational information for predicting the next token.

Dataset	# Train	# Valid	# Test	# Articles	# Vocab	# Entities	# Relations	# Relations/Entity
WikiText	103M	0.2M	0.2M	28,595	267,735	980K	8.9M	9.03
WMT19	151M	0.3M	0.3M	169,180	50,259	976K	7.8M	7.97
enwik8	94M	5M	5M	12,350	256	361K	2.4M	6.66

Table 4.1: Statistics of datasets used in our experiments. For each subset, we show the number of (sub)words for WikiText-103 and WMT19 or the number of characters for enwik8.

4.4 Experiments

Our experiments seek to evaluate the effect of augmenting language models with a relational memory constructed with global context. We introduce datasets used for evaluation (§4.4.1), discuss implementation details (§4.4.2), and present our main results (§4.4.3). We show ablation studies and further analysis of our model in (§4.5).

4.4.1 Datasets and OpenIE

We use three English language modelling datasets: WikiText-103 (Merity et al., 2017), WMT19 (Barrault et al., 2019), and enwik8 (Hutter, 2012). Descriptive statistics of these datasets are shown in Table 4.1. WikiText-103 and WMT19 are (sub)word-level datasets, while enwik8 is a character-level dataset.

WikiText-103 is a knowledge-driven dataset consisting of featured articles from English Wikipedia. WMT19 contains English news from the WMT19 workshop.[3] The news are segmented into months. We use the news from January to October for training, and news in November and December for development and test respectively. Compared to Wikipedia articles, news contains more dynamic and temporal information, exposing new challenges for utilising relational information. We reuse the vocabulary of GPT-2 (Radford et al., 2019) with 50,259 tokens

[3] http://www.statmt.org/wmt19/

to tokenise this dataset. enwik8 contains more than 100M bytes of Wikipedia text. Character-level language modelling has a much smaller vocabulary size than (sub)word-level language modelling.

We perform OpenIE on each dataset. For enwik8, OpenIE is performed after detokenizing its text into words. Statistics of extracted relations are also included in Table 4.1. Each entity from WikiText-103, WMT19 and enwik8 has 9.03, 7.97 and 6.66 relation triples on average.

4.4.2 Implementation Details

All models are implemented with JAX[4] (Bradbury et al., 2018) and Haiku[5] (Hennigan et al., 2020). We set the hidden size to 512 and the number of layers to 16 for all models. In (sub)word-level language modelling, we use adaptive softmax (Grave et al., 2017) for efficiency. We use GELU (Hendrycks and Gimpel, 2016) as our activation function and Adam (Kingma and Ba, 2015) as the optimizer. For training, we use batch size 128 and train the models on 64 16GB TPUs. We apply 4,000 warmup steps, before utilizing cosine annealing to decay the learning rate. Dropout (Srivastava et al., 2014) is applied during training with a rate of 0.25.

We set the lengths of text segment N, extended context M, and the relational memory P to (512, 512, 300), (384, 384, 800) and (768, 1536, 400) for WikiText-103, WMT19 and enwik8, respectively. These are determined by grid searches on development sets.

[4] https://github.com/google/jax
[5] https://github.com/deepmind/dm-haiku

4.4.3 Main Results

We compare with a strong transformer-XL baseline trained under the same setting as our model. Our main results are shown in Table 4.2. We obtain three observations comparing transformer-XL and RELATIONLM. First, RELATIONLM consistently outperforms transformer-XL on all three datasets, demonstrating the effectiveness of relational memory. Note that a decrease of 0.01 is considerable on enwik8 with the bits per character metric. Second, relational memory not only improves language modelling on knowledge-driven articles (WikiText-103), but also generalises to the challenging news domain (WMT19) where information is more dynamic and temporal. Last, the results indicate that relational memory improves both (sub)word-level and character-level language modelling.

Complementarity to SPALM. SPALM (Yogatama et al., 2021) is a state-of-the-art memory-augmented language model. Instead of retrieving relation triples, it retrieves a set of related tokens at each timestep. Specifically, it first stores (context, the next token) pairs from training data. It then uses a pre-trained transformer language model to measure the similarities between the stored contexts and the observed context during training/evaluation. The next tokens of similar contexts are retrieved and are integrated with the observed context via a gating mechanism for generation.

We investigate whether RELATIONLM is complementary to SPALM. Since SPALM also uses a gating mechanism for integrating the retrieved tokens, we first apply RELATIONLM to combine transformer-XL output $\mathbf{h}_t^L$ with relational information to obtain $\mathbf{z}_t$ (as shown in §4.3.2.3), before using SPALM to integrate $\mathbf{z}_t$ with retrieved tokens. The results are shown in Table 4.2. SPALM outperforms transformer-XL and even performs comparably or better compared to RELA-

	Model	# Params	Dev	Test
WikiText	Transformer-XL	122M	19.0	19.9
	RELATIONLM	124M	18.5	19.2
	SPALM	122M	18.1	19.0
	↪ + RELATIONLM	124M	**17.7**	**18.6**
WMT19	Transformer-XL	114M	21.7	21.5
	RELATIONLM	116M	21.0	20.7
	SPALM	114M	20.4	20.3
	↪ + RELATIONLM	116M	**19.8**	**19.6**
enwik8	Transformer-XL	93M	1.05	1.03
	RELATIONLM	95M	1.04	1.02
	SPALM	93M	1.04	1.02
	↪ + RELATIONLM	95M	**1.03**	**1.01**

Table 4.2: We use perplexity (↓) on WikiText-103 and WMT19 and bits per character (↓) on enwik8 for evaluation.

TIONLM on three datasets, demonstrating the effectiveness of retrieving related tokens. However, integrating RELATIONLM and SPALM can further improve the performance, indicating that these two models are not mutually exclusive. Therefore, retrieving relation triples brings complementary benefits to retrieving tokens.

4.5 Analysis

In this section, we study several design choices of relational memory, including its knowledge source, input component, capacity, dynamic OpenIE, entity scoring method used, and speed comparison. We then show quantitative and qualitative analysis results to better understand our model.

4.5.1 Ablations and Design Choice Studies

For this ablation studies, we use the development set of WikiText-103.

Source of relation triples. We compare relation triples extracted from Freebase or using OpenIE on global context. In the Freebase case, we use the Freebase API[6] to obtain relation triples for each entity. For WikiText-103, there are 10.74 relations per entity on average, which is comparable to OpenIE relations (9.03 relations/entity). The results are shown in Table 4.3. Although Freebase relations have been observed to improve the performance on smaller datasets (e.g. WikiText-2 (Logan et al., 2019)) and particular domains (e.g. movies and actors (Ahn et al., 2016)), we find that RELATIONLM with Freebase relations does not improve over transformer-XL on a much larger WikiText-103 dataset. We observe that a large portion of Freebase relations is from infoboxes of Wikipedia pages, which only cover information such as occupation, birth place, and religion. We believe these triples are too general to be useful for most contexts. The result of RELATIONLM with OpenIE shows the advantages of extracting relations from each dataset compared to using Freebase relations.

Model	**Dev**
Transformer-XL	19.0
RELATIONLM + Freebase	19.0
RELATIONLM + OpenIE	**18.5**

Table 4.3: RELATIONLM with OpenIE or Freebase triples.

Ablating relation triples. We ablate relation and/or tail entity from a relation triple (head entity, relation, tail entity) to study the contribution brought by each component. The results are shown in Table 4.4. We find that ablating both relation and tail entity performs comparably to transformer-XL. As head entities are extracted from the observed context, we believe the extended memory of transformer-XL can offset the effect brought by conditioning on head entities.

[6] https://developers.google.com/freebase

Ablating relation performs better than transformer-XL. This shows the advantage of introducing tail entities. Using complete relation triples performs the best, demonstrating the effectiveness of this triple representation of knowledge.

Model	**Dev**
Transformer-XL	19.0
Triple - Relation - Tail	19.0
Triple - Relation	18.7
Triple	**18.5**

Table 4.4: Ablating relation and/or tail entity from a relation triple.

Length of relational memory. We study how many relation triples need to be stored in the relational memory. As shown in Figure 4.2, we can see that the perplexity improves with more relation triples. However, the curve becomes flat with more than 300 relation triples.

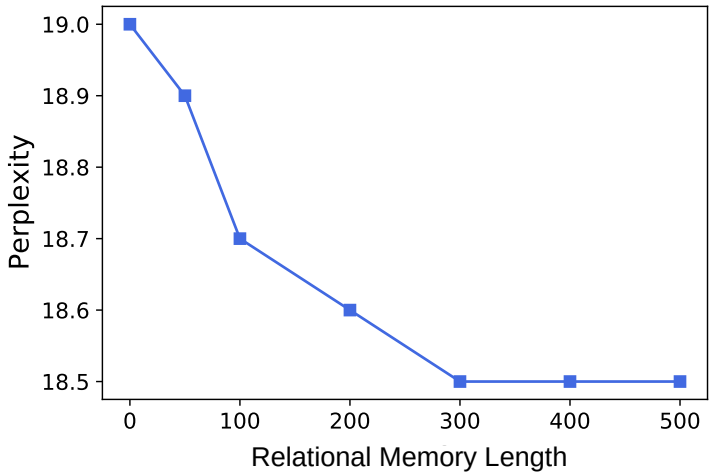


Figure 4.2: Perplexity on WikiText-103 with different number of relation triples.

Length of transformer-XL memory. As increasing the length of context window can capture longer dependency, we study whether increasing the length of extended (transformer-XL) memory removes the performance gap between RE-

LATIONLM and transformer-XL. As shown in Figure 4.3, the performance of both RELATIONLM and transformer-XL improves with larger extended memory. However, RELATIONLM still outperforms transformer-XL even with extended memory length 3072. We conclude that relational memory brings complementary benefits to simply expanding extended memory, since it provides global information about entities on each dataset.

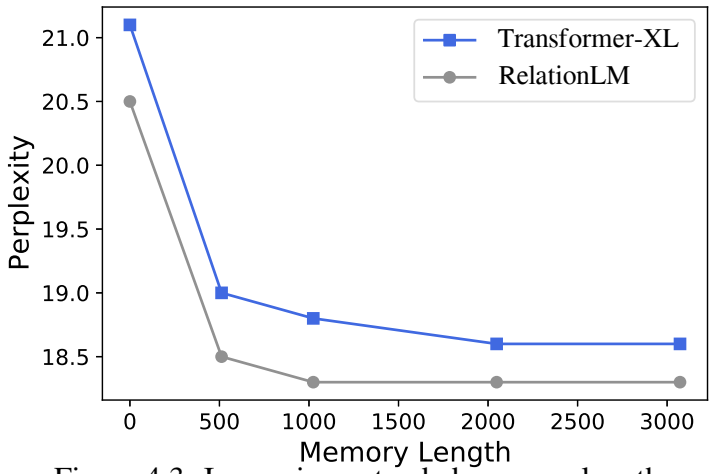


Figure 4.3: Increasing extended memory length.

Dynamic OpenIE. All our main results use dynamic OpenIE. We show results without dynamic OpenIE in Table 4.5. We include the results on three datasets for a comparison. We can see that RELATIONLM with dynamic OpenIE performs comparably to RELATIONLM without dynamic OpenIE on WikiText-103 and enwik8, while larger improvements are obtained on WMT19. This indicates that dynamic OpenIE is more helpful for the news domain, which is more dynamic and temporal compared to knowledge-driven articles.

Entity scoring We study different entity scoring mechanisms for relation retrieval. We consider random selection (where entities extracted from the observed context are randomly selected), frequency-based scoring, and tf-idf scoring. As

Model	Wiki	WMT	ew8
Transformer-XL	19.0	21.7	1.05
w/o Dynamic OpenIE	18.6	21.4	**1.04**
w/ Dynamic OpenIE	**18.5**	**21.0**	**1.04**

Table 4.5: Perplexity with and without dynamic OpenIE.

shown in Table 4.6, tf-idf performs the best.

Model	Dev
Random	19.1
Frequency	18.7
tf-idf	**18.5**

Table 4.6: Perplexity with different entity scoring methods.

Speed comparison. The wall clock time for both training and evaluation is shown in Table 4.7. RELATIONLM is 1.5 and 2.1 times slower during training and evaluation, respectively. Evaluation slows down some more due to dynamic OpenIE as shown in Algorithm 3.

Model	Train	Eval
Transformer-XL	**0.51**	**0.31**
RELATIONLM	0.76	0.65

Table 4.7: The unit is second/step. We use batch size 128 and 1 per step for training and evaluation, respectively.

4.5.2 Does Relational Memory Improve Coherence?

For evaluating coherence, we use two automatic metrics—knowledge perplexity and knowledge F_1—to investigate whether the models can faithfully use entities. We further perform a human evaluation to study whether language models can

Dataset	Subset	# Entity	# Non-Entity
WikiText	Dev	61.6K	155.9K
	Test	65.8K	179.7K
WMT	Dev	84.9K	262.2K
	Test	81.0K	256.6K
enwik8	Dev	1.7M	3.3M
	Test	1.7M	3.3M

Table 4.8: Statistics of entity and non-entity tokens.

	Metric	Model	Dev	Test
Knowledge PPX	WikiText	Transformer-XL	47.3	52.3
		RELATIONLM	**45.6**	**50.9**
	WMT	Transformer-XL	77.2	77.0
		RELATIONLM	**73.2**	**73.1**
	enwik8	Transformer-XL	2.25	2.21
		RELATIONLM	**2.22**	**2.19**
Non-entity PPX	WikiText	Transformer-XL	13.3	13.8
		RELATIONLM	**13.0**	**13.4**
	WMT	Transformer-XL	14.4	14.4
		RELATIONLM	**14.2**	**14.3**
	enwik8	Transformer-XL	**1.98**	**1.95**
		RELATIONLM	**1.98**	**1.95**

Table 4.9: Knowledge perplexity (↓) and non-entity perplexity (↓).

generate coherent and knowledgeable sequences. We believe the human evaluation is a reliable way of evaluating coherence. This claim is advocated in Barzilay et al. (2005). We note that question answering is also often used to evaluate coherence (Guu et al., 2020; Lin et al., 2021). We leave this to future work.

Knowledge perplexity. While vanilla perplexity considers all words in an evaluation set, knowledge perplexity only considers entities for calculating perplexity. We use it to evaluate whether the model can assign higher probabilities for the cor-

Metric	Model	Dev	Test
WikiText	Transformer-XL	9.9	9.4
	RELATIONLM	**11.4**	**11.2**
WMT	Transformer-XL	11.4	11.0
	RELATIONLM	**12.6**	**12.3**
enwik8	Transformer-XL	16.0	18.9
	RELATIONLM	**16.6**	**19.4**

Table 4.10: Knowledge F_1 (↑).

rect entities under different contexts. Table 4.8 shows the numbers of entity words and non-entity words in our corpora. We show the results in Table 4.9. We observe that the gap between RELATIONLM and transformer-XL is larger on knowledge perplexity. RELATIONLM only performs comparably or slightly better compared to transformer-XL on non-entity perplexity. This shows that relational memory is helpful for predicting entity words. Note that knowledge perplexity tends to be much higher than perplexity on non-entity words, indicating the difficulty of predicting entity words. This collection of results indicates that relational memory helps the model use entities coherently and consistently under different contexts.

Knowledge F_1. We use knowledge F_1 to explore whether our model generates tokens that are grounded to its contexts. Given a context as input, we sequentially generate 32 words (or 128 characters) for word-(character-)level language modelling by sampling from the distribution of the next word (character). To reduce variance, we generate 100 continuations for each context. We then perform entity recognition for both the generated sequences and their corresponding ground-truth sequences and calculate an F_1 score based on these two sets of entities. For example, given the context "...Ayola was nominated and shortlisted for the 'Female Performance in TV' award", we compare the generated text and the ground truth "in the 2006 Screen Nation Awards, for her role as Kyla Tyson in Holby City..." to

calculate F_1. The results are shown in Table 4.10. We notice that RELATIONLM performs better compared to transformer-XL. We conclude that models with relational memory can generate more coherent and logical text.

Human evaluation. We conduct a human evaluation to study whether language models can generate coherent and knowledgeable sequences. We take 1,000 contexts from the test set of WikiText-103. We show the contexts, ground-truth sequences, and continuations generated by RELATIONLM and transformer-XL to five annotators. We use greedy decoding for both models. We shuffle the order of the continuations generated by RELATIONLM and transformer-XL so that the annotators are unaware of the sources of sequences. We then pose the following questions to the annotators:

1. **Coherent.** Given the context and its ground-truth continuation for reference, which generated sequence is more logical and coherent?
2. **Knowledgeable.** Given the context and its ground-truth continuation, which generated sequence provides more insights and is more knowledgeable?

We show the results in Table 4.11. We find that RELATIONLM outperforms transformer-XL in the human evaluation. These results are consistent with the two automatic metrics, knowledge perplexity and knowledge F_1. This corroborates our claim that relational memory improves coherence in language modelling.

4.5.3 Qualitative Analysis

Gate values. As we use a gating function to integrate transformer-XL with relational information, we study gate values in this section. The histogram of gate values is shown in Figure 4.5. We notice that the histogram concentrates around 0.9. This is expected because non-entity words, which account for a large por-

Model	Coherent	Knowledgeable
Transformer-XL	388	416
RELATIONLM	**612**	**584**

Table 4.11: We show the number of contexts in which a continuation from a particular model is chosen by human evaluators for each evaluation criterion. Recall that the total number of contexts used for human evaluation is 1,000. Since we have five annotators, we use majority voting to decide the favored model for each continuation. We use the Kappa statistic to measure inter-annotator agreement. The statistic is 0.64, which shows substantial agreement among the annotators.

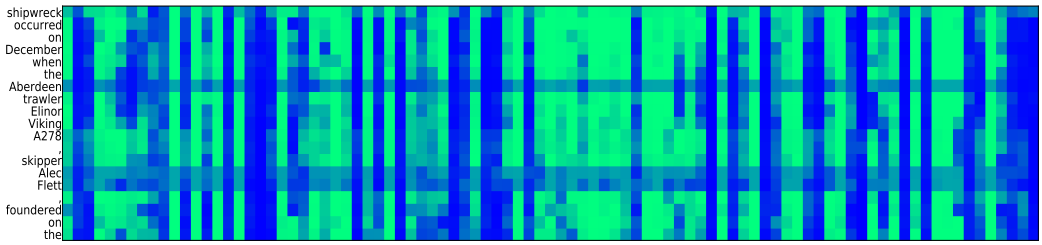


Figure 4.4: Heatmap of gate values.

tion of text (according to Table 4.8), benefit less from the relational memory and mainly rely on the observed context for prediction as shown in §4.5.2. We further calculate the average gate values for entity words and non-entity words. The average gate value for entity words is 0.87, while the average value is 0.92 for non-entity words. This confirms that entity words rely more on relational information for prediction compared to non-entity words. We also plot the heatmap of gate values and a cherry-picked example is shown in Figure 4.4. Note that we randomly select 100 dimensions from 512 dimensions for readability. We notice that the entities, Aberdeen and Alec Flett, use more relational information than other positions (as shown by the horizontal blue lines). These results demonstrate that RELATIONLM can adaptively incorporate relational information for prediction.

Example. We show three cherry-picked examples in Table 4.12. We take the first for illustration, which shows a text segment from the article, Joe Biden 2008

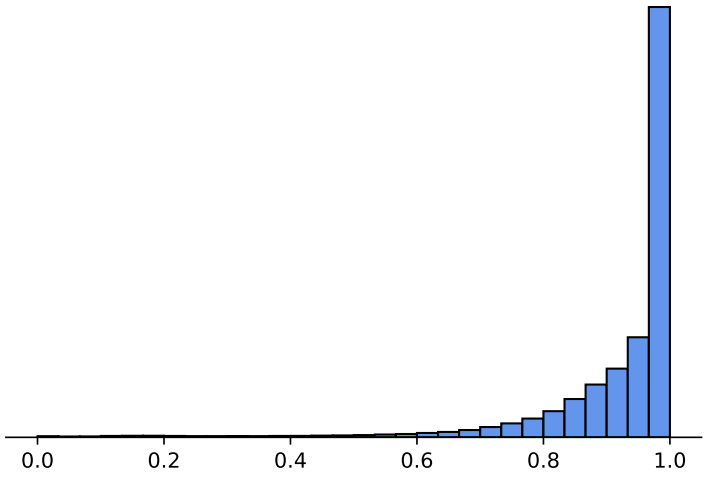


Figure 4.5: Histogram of gate values g_t.

presidential campaign[7] and some retrieved relations. We find that the first two relations, (Joe Biden, senior Senator, Delaware) and (Joe Biden presidential campaign, began, January 7 2007), are extracted from previous text segments, while (Joe Biden, was nominated, vice president) and (Biden, withdrew nomination, 1987) are extracted from the other articles, Joe Biden[8] and Joe Biden 1988 presidential campaign[9], respectively. We notice that the relation (Joe Biden, was nominated, vice president) is highly predictive of the sequence, "Biden was selected to be Democratic presidential nominee Barack Obama's vice presidential running mate". From the observed context, the model also identifies a closely related entity, Barack Obama, and retrieves the relation (Barack Obama, president of, United States). Therefore, we conclude that the relational memory can give a global picture of related entities and provide relevant information for language modelling.

[7] https://en.wikipedia.org/wiki/Joe_Biden_2008_presidential_campaign
[8] https://en.wikipedia.org/wiki/Joe_Biden
[9] https://en.wikipedia.org/wiki/Joe_Biden_1988_presidential_campaign

Seven months after conclusion of his campaign, Biden was selected to be Democratic presidential nominee Barack Obama's vice presidential running mate. The pair won in the general election, and were sworn in on January 20, 2009 ... (Joe Biden, senior Senator, Delaware) (Joe Biden presidential campaign, began, January 7 2007) (Joe Biden, was nominated, vice president) (Biden, withdrew nomination, 1987) (Barack Obama, president of, United States)
From 7 February 2006 to 9 December 2008, Ayola starred in BBC medical drama Holby City as nurse Kyla Tyson. She had previously appeared in Holby City 's sister show Casualty … (Holby City, is, BBC medical drama) (Rakie Ayola, played the role, Kyla Tyson)
Independiente became Arjona's fourth number one album on the Billboard Top Latin Albums where it debuted for the week ending 22 October 2011. For thirteen non-consecutive weeks it topped the Latin Pop Albums chart ... (Independiente, number one on, Top Latin Albums chart) (Independiente, became number one on, 22 October 2011)

Table 4.12: Three examples of text segment and retrieved relations (based on previous text segments).

Causal intervention. We use causal intervention to study whether changing the contents in the relational memory will affect language model prediction. Given the relation (Obama, born in, Hawaii) along with other relations about Barack Obama, we let the model complete the sequence, "Obama was born in". RELATIONLM outputs "Obama was born in and raised in Hawaii." with greedy decoding. However, after modifying the relation to (Obama, born in, Kenya), we obtain "Obama was born in Kenya and was the first African-American president.". We further

change to (Obama, born in, Paris) and the model outputs "Obama was born in Paris, France.". This indicates that RELATIONLM can take advantage of relation triples for making prediction. While we can also use prompts as intervention for vanilla language models, it remains challenging about selecting the appropriate prompts in different applications (Liu et al., 2021a).

4.6 Summary

In this chapter, we presented RELATIONLM, a language model that is augmented with relational memory constructed with global context. We showed how to obtain relevant knowledge graphs for a given corpus and how to combine them with a state-of-the-art language model such as transformer-XL. We demonstrated that our model improves performance and coherence on WikiText-103, WMT19 and enwik8. We also performed a comprehensive analysis to better understand how our model works. In the next chapter, we introduce pretraining the noisy channel model with external context for task-oriented dialogue.

5 | Pretraining the Noisy Channel Model with External Context

5.1 Introduction

In Chapters 3 and 4, we have utilized local context and global context, which come from each input sequence and the whole training corpus, respectively. Beyond these, we can take advantage of resources from the Web for training more universal representations. We refer to the resources beyond the training corpus as external context in this thesis. In this chapter, we study utilizing external context to pretrain the noisy channel model for task-oriented dialogue. Task-oriented dialogue agents provide a conversational interface to assist users in accomplishing specific goals, such as finding a restaurant or booking a hotel (Seneff and Polifroni, 2000; Raux et al., 2005; Budzianowski et al., 2018; Peng et al., 2020a). Increasing demand from industry for natural language assistants and scalable customer service solutions has recently been driving a renaissance in the development of task-oriented dialogue models. In addition, the specification of explicit dialogue agent goals, afforded by the task-oriented paradigm, makes such research easier to ground and evaluate than open-domain chatbots.

Current research on task-oriented dialogue is dominated by monolithic sequence-to-sequence models that directly parameterize the conditional distribution of the response given the prior dialogue context. However, this monolithic approach conflates the task-specific and language-general aspects of dialogue, and adversely favors short and generic responses (Bao et al., 2020) due to the explaining-away effect (Klein and Manning, 2002).

Here we pursue an alternative to the direct model. Employing Bayes' rule allows

us to factorize the probability of the response given the context $p(\mathcal{R}|\mathcal{C})$ into a language model $p(\mathcal{R})$ and a context model $p(\mathcal{C}|\mathcal{R})$.[1] Within NLP, this approach is traditionally known as the noisy channel model (Shannon, 1948), and has recently seen renewed interest with its successful application to neural machine translation (Yu et al., 2017, 2020; Yee et al., 2019).

We hypothesize that the noisy channel reformulation is advantageous for dialogue because the factorization enables each sub-module to specialize in a dialogue sub-task. In particular, the context conditional model can help to discount short and generic responses and mitigate the explaining-away effect, while the language model helps ensure that responses are natural. Moreover, a larger noisy channel model can be trained with the same hardware, by training the sub-modules separately, yielding additional improvements.

We use external context to pretrain the noisy channel model. We first utilise Reddit data for pretraining and find that pretraining with Reddit data can significantly improve the performance of the noisy channel model. Further, we evaluate transfer across task-oriented dialogue datasets by implementing a second pretraining stage using Taskmaster (Byrne et al., 2019) and Schema-Guided Dialogue (Rastogi et al., 2020) as training data, before fine-tuning on our final tasks.

We evaluate the algorithm on three datasets, MultiWOZ 2.0 (Budzianowski et al., 2018), CamRest676 (Wen et al., 2017a) and SMCalFlow (Andreas et al., 2020). We find that a noisy channel model with the same number of parameters as a direct model achieves better accuracy on three task-oriented dialogue datasets. Further analysis demonstrates that the noisy channel models can decode responses with similar lengths and Zipf scores compared to ground-truth responses and reduce the likelihood of falling into repetition loops (Holtzman et al., 2019).

[1] Here we abstract away from the prediction of belief states and dialogue acts, which also form part of our generative model.

In the rest of this chapter, we first review related work (§5.2). Next, we introduce a seq-to-seq dialogue model (§5.3). Then, we present our noisy channel model to mitigate the explaining-away effect manifested in the seq-to-seq dialogue model (§5.4). Next, we utilize external context to pretrain the noisy channel model (§5.5). Finally, we evaluate the performance of the noisy channel model (§5.6, §5.7, §5.8).

5.2 Related Work

Task-oriented dialogue models: Most task-oriented dialogue systems break down the task into three components: belief tracking (Henderson et al., 2013; Mrksic et al., 2017; Rastogi et al., 2017; Nouri and Hosseini-Asl, 2018; Wu et al., 2019; Zhang et al., 2020a; Zhou and Small, 2019; Heck et al., 2020), dialogue act prediction (Wen et al., 2017a; Tanaka et al., 2019) and response generation (Chen et al., 2019; Budzianowski et al., 2018; Lippe et al., 2020). Traditionally, a modular approach is adopted, where these components are optimized independently (i.e. a pipeline design) or learned via multi-task learning (i.e. some parameters are shared among the components) (Wen et al., 2017b; Neelakantan et al., 2019; Zhao et al., 2019; Mehri et al., 2019; Tseng et al., 2020; Lee et al., 2020). However, it is known that improvements in one component do not necessarily lead to overall performance improvements (Ham et al., 2020), and the modular approach suffers from error propagation in practice (Liu and Lane, 2018). These observations gave rise to the sequence-to-sequence approach (Lei et al., 2018; Pei et al., 2019; Budzianowski and Vulić, 2019; Wu et al., 2021a; Zhang et al., 2020b; Ham et al., 2020; Hosseini-Asl et al., 2020; Peng et al., 2020a; Yang et al., 2020), where dialogue beliefs and acts are represented as text spans, and a sequence-to-sequence model is applied to subsume the three components. Our work is situated

within this general approach. In contrast to previous work, however, which uses a direct model for decoding, we introduce the noisy channel model to improve task-oriented dialogue.

Pretraining models for dialogue: Recent work has applied pretraining (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019) to dialogue. For open-domain dialogue, DialoGPT (Zhang et al., 2020c) and CGRG (Wu et al., 2021b) extend GPT-2 (Radford et al., 2019) for response generation. PLATO (Bao et al., 2020) and PLATO-2 (Bao et al., 2021) pretrain a latent variable model with social media data for diversified response generation. Meena (Adiwardana et al., 2020) collects a large-scale social media corpus for pretraining and proposes a metric named sensibleness and specificity average for evaluation. Roller et al. (2021) study various strategies for building an open-domain chatbot with Reddit for pretraining. For task-oriented dialogue, ToD-BERT (Wu et al., 2020) fine-tunes BERT (Devlin et al., 2019) for four tasks, including intention detection, belief tracking, dialogue act prediction, and response selection. SC-GPT (Peng et al., 2020b) fine-tunes GPT-2 for few-shot response generation with given dialogue acts. Ham et al. (2020) fine-tune GPT-2 for belief tracking and context-to-response generation. SimpleTOD (Hosseini-Asl et al., 2020) proposes a method to serialize dialogue beliefs and acts into text spans and fine-tunes GPT-2 for end-to-end dialogue modeling. SOLOIST (Peng et al., 2020a) uses a series of task-oriented dialogue datasets to further pretrain GPT-2 before fine-tuning it on final tasks for evaluation. Unlike these BERT- or GPT-initialized task-oriented dialogue models, which are essentially pretrained with general text, such as Wikipedia and BookCorpus, we use a Reddit dump to pretrain the models to learn from open-domain dialogues.

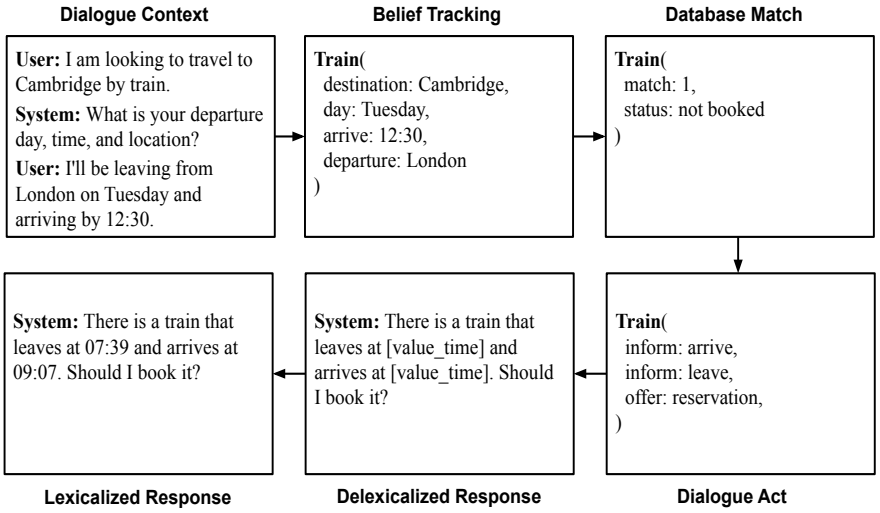


Figure 5.1: The data flow of one turn in a task-oriented dialogue for train booking from MultiWOZ.

5.3 A Seq-to-Seq Dialogue Model

In this section, we introduce a discriminative sequence-to-sequence model for task-oriented dialogue. The traditional sequence of steps needed to produce a system turn in a task-directed dialogue is shown in Figure 5.1, with an example from MultiWOZ 2.0 (Budzianowski et al., 2018). Given a dialogue context containing previous user and system utterances, the dialogue system first predicts a belief state, consisting of a set of slot-value pairs (e.g. `destination: Cambridge`), to capture user intent. To ground the system with external information, the belief state can be converted into a database query in order to retrieve relevant information, such as the number of matches and booking information. Next, the system predicts a set of dialogue acts, representing the abstract meaning of the proposed dialogue response (Austin, 1975). Finally, a delexicalized dialogue response is generated, where slot values are replaced by generic placeholders, such as `value_time` for a train departure time, in order to reduce lexical variation. The delexicalized response can be converted to a lexicalized response

in post-processing by filling in the slot values based on belief states and database information.

We use the MultiWOZ schema for illustration in Section 5.3 and 5.4, but our models easily generalize to different schema annotations (e.g. datasets without annotated dialogue acts (Andreas et al., 2020)).

Since it is well known that pipelined models tend to suffer from error propagation, many NLP tasks have been reformulated in recent years as end-to-end text-to-text transformations (Raffel et al., 2020; Brown et al., 2020). State-of-the-art task-oriented dialogue systems have followed this approach (Hosseini-Asl et al., 2020; Peng et al., 2020b). We represent the example from Figure 5.1 as follows, serializing turns and using special start and end tokens to encapsulate each data field:

> Context: [c] I am looking to ... [/u] What is your ... [/r] I'll be leaving ... [/u] [/c]
> Belief: [b] [train] destination Cambridge, day Tuesday, arrive 12:30, departure London [/b]
> Database: [db] [train] match 1, status not booked [/db]
> Act: [a] [train] inform arrive, inform leave, offer reservation [/a]
> Response: [r] There is a train that leaves at [value_time] and arrives at [value_time]. Should I book it? [/r]

Given this text representation, the discriminative approach models $p(\mathcal{B}, \mathcal{A}, \mathcal{R}|\mathcal{C})$, where $\mathcal{C}$, $\mathcal{B}$, $\mathcal{A}$, and $\mathcal{R}$ represent dialogue context, belief state, dialogue act, and delexicalized response, respectively.[2] We use the serialized text of the dialogue context as input, and the concatenation of belief state, dialogue act, and response as target output, making the task amenable to the application of an autoregressive sequence-to-sequence model. $\mathcal{B}$, $\mathcal{A}$ and $\mathcal{R}$ can be generated sequen-

[2]We do not model the probabilities of database state or lexicalized response, as these are deterministic given the belief state and delexicalized response, respectively.

tially with direct decoding methods, such as greedy decoding and beam search. We use a sequence-to-sequence transformer (Vaswani et al., 2017) to implement $p(\mathcal{B}, \mathcal{A}, \mathcal{R}|\mathcal{C})$. This distribution will also be used to build the noisy channel model in Section 5.4.

5.4 Noisy Channel Model for Dialogue

While direct decoding is an effective approach for decoding belief states (Hosseini-Asl et al., 2020), it may be sub-optimal for generating responses. First, it favors short and generic responses (Bao et al., 2021). As a result, the decoded responses are bland and lack diversity (Li et al., 2016). Second, it suffers from the explaining-away effect (Klein and Manning, 2002), where inputs are "explained-away" by highly predictive output prefixes. For example, if there is one hotel matching the user's intent as encoded in the belief state, the model is nevertheless prone to decoding "no" given the output prefix "there is", ignoring the input information.

In this work, we propose using the neural noisy channel model (Yu et al., 2017) to mitigate the above problems for response generation. Given an input sequence x and output sequence y, the noisy channel formulation (Shannon, 1948) uses Bayes' rule to rewrite the model $p(y|x)$ as $\frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y)$. It was originally applied to speech recognition, where $p(y|x)$ is a conditional model of the source text given a noisy observation. The *channel model* $p(x|y)$ estimates the probability of the observation given the source, while $p(y)$ is an unconditional *language model* (or *source model*), which can be trained on unpaired data. More recently it has been applied to machine translation, where y is a translation of input text x.

Abstracting away from belief states and dialogue acts, for task-oriented dialogue

we want to estimate $p(\mathcal{R}|\mathcal{C})$, the probability of a response given a context. The channel model $p(\mathcal{C}|\mathcal{R})$, given a response, predicts a distribution over contexts which might have elicited that response. The source model $p(\mathcal{R})$ is an unconditional language model. In this extension of the noisy channel approach to task-oriented dialogue, the "channel" can be understood as connecting dialogue contexts with suitable responses.

For the full task, we develop a noisy channel model for $p(\mathcal{B}, \mathcal{A}, \mathcal{R}|\mathcal{C})$. Using the chain rule, $p(\mathcal{B}, \mathcal{A}, \mathcal{R}|\mathcal{C}) = p(\mathcal{B}|\mathcal{C}) \cdot p(\mathcal{A}, \mathcal{R}|\mathcal{C}, \mathcal{B})$. Following Hosseini-Asl et al. (2020), we use the direct model described in Section 5.3 to parameterize $p(\mathcal{B}|\mathcal{C})$ and decode $\mathcal{B}$, which our preliminary experiments confirmed to be advantageous.

We use the noisy channel formulation to parameterize $p(\mathcal{A}, \mathcal{R}|\mathcal{C}, \mathcal{B})$. Using Bayes' Rule, $p(\mathcal{A}, \mathcal{R}|\mathcal{C}, \mathcal{B}) \propto p(\mathcal{C}, \mathcal{B}|\mathcal{A}, \mathcal{R}) \cdot p(\mathcal{A}, \mathcal{R})$. The channel model $p(\mathcal{C}, \mathcal{B}|\mathcal{A}, \mathcal{R})$ and source model $p(\mathcal{A}, \mathcal{R})$ are implemented as transformers.

We choose to use the noisy channel formulation for decoding $\mathcal{A}$ based on preliminary experiments which showed improved overall accuracy over direct decoding, possibly because poor dialogue act prediction by the direct model led to worse quality responses. The serialized text of $\mathcal{A}$ and $\mathcal{R}$ are concatenated during training, and the decoded sequence is split into $\mathcal{A}$ and $\mathcal{R}$ with the special start/end tokens during decoding.

We suggest that the noisy channel model has three advantages over the direct model for response generation: (1) The channel model can penalize short and generic responses. Such responses can be mapped to a large number of contexts, resulting in a flat distribution over contexts. This leads to a lower channel model score for short and generic responses (Zhang et al., 2020c). (2) The channel model ensures that $(\mathcal{A}, \mathcal{R})$ must explain the corresponding $(\mathcal{C}, \mathcal{B})$, alleviating the explaining-away effect (Yu et al., 2017). (3) The source model, an uncondi-

tional distribution over $\mathcal{A}$ and $\mathcal{R}$, can make use of abundant non-dialogue textual data for pretraining, further improving the fluency of generated sequences (Brants et al., 2007). We leave exploration of this last advantage for future work, as we pretrain all sub-modules with the same data.

5.4.1 Decoding

Since exact decoding from the noisy channel model $\arg\max_{\mathcal{A},\mathcal{R}} p(\mathcal{C},\mathcal{B}|\mathcal{A},\mathcal{R}) \cdot p(\mathcal{A},\mathcal{R})$[3] is computationally intractable, we experiment with two approximation methods, noisy channel reranking and noisy channel online decoding. Since these methods rely on $p(\mathcal{A},\mathcal{R}|\mathcal{C},\mathcal{B})$ as a proposal distribution for approximation, and both $p(\mathcal{A},\mathcal{R}|\mathcal{C},\mathcal{B})$ and $p(\mathcal{B}|\mathcal{C})$ are parameterized with the direct model introduced in Section 5.3, our noisy channel model therefore has three sub-modules: a direct model $p(\mathcal{B},\mathcal{A},\mathcal{R}|\mathcal{C})$, a channel model $p(\mathcal{C},\mathcal{B}|\mathcal{A},\mathcal{R})$, and a source model $p(\mathcal{A},\mathcal{R})$.

Noisy channel reranking: Noisy channel reranking first decodes $\mathcal{B}$ and then continues decoding a list $\mathcal{S}$ of $(\mathcal{A},\mathcal{R})$ pairs by beam search with the direct model, prior to utilizing the noisy channel model to rerank $(\mathcal{A},\mathcal{R})$ pairs. In particular, during beam search, partial sequences are expanded and pruned with $p(\mathcal{A},\mathcal{R}|\mathcal{C},\mathcal{B})$ (from the direct model in Section 5.3). The pairs after decoding are reranked using the following model combination:

$$\begin{aligned}(\mathcal{A}',\mathcal{R}') = \arg\max_{(\mathcal{A},\mathcal{R})\in\mathcal{S}} &\log p(\mathcal{A},\mathcal{R}|\mathcal{C},\mathcal{B}) + \\ &\lambda_1 \cdot \log p(\mathcal{C},\mathcal{B}|\mathcal{A},\mathcal{R}) + \\ &\lambda_2 \cdot \log p(\mathcal{A},\mathcal{R}) + \\ &\lambda_3 \cdot |\mathcal{A},\mathcal{R}|,\end{aligned} \tag{5.1}$$

[3] Although exact decoding is also computationally intractable for the direct model, approximating $\arg\max_{\mathcal{B}} p(\mathcal{B}|\mathcal{C})$ is well-studied, e.g. beam search. The decoding for $\mathcal{B}$ is therefore omitted here.

Algorithm 4: Online decoding for the noisy channel.

Input : Context $\mathcal{C}$
Output: Belief, act and response $(\mathcal{B}, \mathcal{A}, \mathcal{R})$
Decode $\mathcal{B}$ given $\mathcal{C}$ with $p(\mathcal{B}|\mathcal{C})$
Beam: $\mathcal{S} = \{([a])\}$
while $\texttt{end}(\mathcal{S})$ is **False do**
  $\mathcal{S}' = \emptyset$
  for $\mathcal{O}$ **in** $\mathcal{S}$ **do**
    if $\mathcal{O}.\texttt{last}()$ is [/r] **or** $|\mathcal{O}| > l$ **then**
      $\mathcal{S}'.\texttt{add}(\mathcal{O})$
      continue
    end
    Get k_1 tokens $o^1, ..., o^{k_1}$ from the direct model $p(O_{|\mathcal{O}|+1}|\mathcal{C}, \mathcal{B}, \mathcal{O})$
    for o^i **in** $(o^1, ..., o^{k_1})$ **do**
      $\mathcal{S}'.\texttt{add}((\mathcal{O}, o^i))$
    end
  end
  $\mathcal{S} = \underset{\mathcal{O} \in \mathcal{S}'}{\text{top_k}_2} \log p(\mathcal{O}|\mathcal{C}, \mathcal{B}) + \lambda_1 \cdot \log p(\mathcal{C}, \mathcal{B}|\mathcal{O}) + \lambda_2 \cdot \log p(\mathcal{O}) + \lambda_3 \cdot |\mathcal{O}|$
end
Select $\mathcal{O} \in \mathcal{S}$ with the largest score using Eq. 5.1 and return $(\mathcal{B}, \mathcal{A}, \mathcal{R})$

where $|\mathcal{A}, \mathcal{R}|$ denotes the length of $(\mathcal{A}, \mathcal{R})$, and λ_1, λ_2 and λ_3 are hyperparameters. Besides the channel model $p(\mathcal{C}, \mathcal{B}|\mathcal{A}, \mathcal{R})$ and the source model $p(\mathcal{A}, \mathcal{R})$, we additionally use the direct model $p(\mathcal{A}, \mathcal{R}|\mathcal{C}, \mathcal{B})$ and a length bias $|\mathcal{A}, \mathcal{R}|$ to encourage responses with high direct model likelihood and discourage short responses, respectively.

Noisy channel online decoding: In contrast to reranking, online decoding applies the noisy channel model during beam search for pruning partial sequences, thus exploring a larger search space.

As shown in Algorithm 4, we first decode the belief state with $p(\mathcal{B}|\mathcal{C})$, which comes from the direct model in Section 5.3. Then, starting with a beam $\mathcal{S}$ containing a single sequence [a] (the dialogue act start token), we continuously expand the sequences in $\mathcal{S}$ until $\texttt{end}(\mathcal{S})$ is met, i.e. all sequences in $\mathcal{S}$ either end

with [/r] or have lengths larger than l. In each iteration, we first expand the sequences in the beam, then prune the expanded beam. To expand a partial act and response sequence (denoted as $\mathcal{O}$ in Algorithm 4), a naive way is to use the noisy channel model to score $|V|$ (the vocabulary size) possible expansions, which is computationally expensive. Instead, we use the probability of the next token $p(O_{|\mathcal{O}|+1}|\mathcal{C}, \mathcal{B}, \mathcal{O})$ (where $|\mathcal{O}|$ denotes the length of $\mathcal{O}$) to select k_1 candidates to be scored by the noisy channel model. This next token probability is from the direct model introduced in Section 5.3. One straightforward way to select k_1 expansions from $p(O_{|\mathcal{O}|+1}|\mathcal{C}, \mathcal{B}, \mathcal{O})$ is using the top-k maximization, but we can also take advantage of the advances in sampling from a categorical distribution for text generation (e.g. top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2019)). After the expansion, we prune the expanded beam $\mathcal{S}'$ to obtain a smaller beam with k_2 partial sequences based on the model combination in Eq. 5.1. Compared to noisy channel reranking, online decoding applies the noisy channel model during beam search, which is potentially less biased towards the direct model.

In summary, we note that beam search for both the direct model and the online decoding for our noisy channel model decodes $(\mathcal{B}, \mathcal{A}, \mathcal{R})$ autoregressively. Thus both approaches are end-to-end models for task-oriented dialogue. The key difference is that noisy channel online decoding uses Eq. 5.1 for pruning, while the direct model uses $p(\mathcal{A}, \mathcal{R}|\mathcal{C}, \mathcal{B})$.

5.5 Pretraining with External Context

We use three transformer (Vaswani et al., 2017) networks to parameterize the direct model $p(\mathcal{B}, \mathcal{A}, \mathcal{R}|\mathcal{C})$, the channel model $p(\mathcal{C}, \mathcal{B}|\mathcal{A}, \mathcal{R})$ and the source model $p(\mathcal{A}, \mathcal{R})$, respectively. The input to each transformer is the sum of four embed-

dings: word embeddings, position embeddings, role embeddings (user/system), and turn embeddings (each word corresponds to a turn number). Cross entropy is used as the loss function.

Given training samples $(\mathcal{C}, \mathcal{B}, \mathcal{A}, \mathcal{R})$, if we train the channel model using complete $(\mathcal{A}, \mathcal{R})$ pairs as input, a significant discrepancy arises between training and decoding for noisy channel online decoding. Since the channel model is used to score partial act and response pairs, i.e. $p(\mathcal{C}, \mathcal{B}|\mathcal{O})$ in Algorithm 4, the channel model trained with complete $(\mathcal{A}, \mathcal{R})$ pairs is unsuited to scoring partial sequences. In order to manually create partial sequences during training that are better matched for online decoding, we truncate the $(\mathcal{A}, \mathcal{R})$ pairs with a truncation length uniformly sampled from 1 to the sequence length (inclusive). The direct model and the source model are trained with complete sequences, as partial sequences occur naturally in their standard autoregressive training procedure.

As in-domain dialogue data are usually scarce, we use a two-stage pretraining strategy to enhance the noisy channel model with external context. Although the effectiveness of pretraining with Reddit data has been validated for open-domain dialogue (Zhang et al., 2020c; Bao et al., 2020; Adiwardana et al., 2020), relatively little work has applied such data to task-oriented dialogue.[4] In the first stage, we explore Reddit pretraining (where the Reddit data is pre-processed into $(\mathcal{C}, \mathcal{R})$, i.e. context-response, pairs as described below). In the second stage, we use two task-oriented dialogue datasets, Taskmaster[5] (Byrne et al., 2019) and Schema-Guided Dialogue[6] (Rastogi et al., 2020), to specialize the Reddit-pretrained models. Since the Reddit data consists of open-domain-style dialogues (where belief states and dialogue acts are missing), pretraining on these datasets can familiarize the models

[4] One exception is Henderson et al. (2019), who use Reddit data to improve response retrieval and selection. We focus on response generation in this work.

[5] https://cutt.ly/xkuUHUa

[6] https://cutt.ly/QkuUZUu

with the sequence-to-sequence representation of task-oriented dialogue. Three models, a context-to-response model, a response-to-context model and a response language model, are pretrained to initialize the direct model, the channel model and the source model, respectively.

5.5.1 Implementation Details

Models: All models are implemented with JAX (Bradbury et al., 2018) and Haiku (Hennigan et al., 2020). For the direct model introduced in Section 5.3, we use a transformer model with hidden size 512, 12 encoder-decoder layers, and 16 self-attention heads. The model has 114M parameters. For the noisy channel model, we use a base setting and a large setting. The base setting reduces the number of layers to 5, hidden size to 384 and self-attention heads to 12. Its sub-modules, a direct model, a reverse model and a language model, have 43M, 43M and 30M parameters, respectively. We employ the base setting for a fair comparison with a single direct model using roughly the same number of parameters (116M vs. 114M). For the large setting, we use the same hyperparameters as the direct model (114M), so that its sub-modules, a direct model, a reverse model and a language model, have 114M, 114M and 64M parameters, respectively. We use this large setting to explore the limits of the noisy channel model. The large noisy channel model (292M) is 2.56 times larger compared to the direct model (114M). This illustrates another advantage of the noisy channel model during training. While training a direct model with 292M parameters will overflow the memory of 16GB TPUs (v3) without using model parallelism, training the sub-modules of the large noisy channel model can easily fit into 16GB TPUs, as these modules are independently trained with no need to load three modules for training. This enables us to train a noisy channel model with more parameters compared to training a direct model using the same hardware. For inference, we still need to load the

sub-modules into a TPU. Since gradients are not required during inference, we are able to load the three sub-modules of the large noisy channel model (292M) into a single TPU with 16GB memory for decoding. The large noisy channel model (292M) still consumes more memory than the direct model (114M) during inference.

Pretraining settings: The maximum sequence length l is set to 1024, and sequences with longer lengths are truncated. We reuse the vocabulary from GPT-2 (Radford et al., 2019), which contains 50,257 BPE tokens. We use PreNorm (Nguyen and Salazar, 2019) for faster convergence. GELU (Hendrycks and Gimpel, 2016) is applied as the activation function. Following ALBERT (Cer et al., 2017), dropout is disabled during pretraining. We use the normal distribution truncated to the range $[-0.01, 0.01]$ to initialize the input embeddings, while other parameters are initialized using the normal distribution with zero mean and standard deviation 0.1. The batch size is set to 256. The LAMB optimizer (You et al., 2020) ($b_1 = 0.9$ and $b_2 = 0.999$) is employed for optimization. The initial learning rate is 1e-7, and we apply 4000 warmup steps to increase the learning rate to 1e-3, before utilizing cosine annealing to decay the learning rate. Gradient clipping with clipping value 1 is applied to avoid gradient explosion. We use gradient accumulation with accumulation step 20.

Pretraining: For Reddit pretraining, we download a Reddit dump (with Reddit posts ranging from 2005-12 to 2019-09) from PushShift.[7] Since the comments of a Reddit post are organized into a tree, we extract paths from a tree as dialogue turns. The last comment of each comment path is regarded as the response, while the others are used as the dialogue context. We pretrain each model for 400,000 steps, consuming 102,400,000 (400,000 $\times$ 256) comment paths in total. For the task-oriented pretraining, we combine the two datasets, Taskmaster and Schema-

[7] https://pushshift.io/

Dataset	# Dialog	# Turn	Avg. Turn/Dialog	Avg. Token/Turn	# Domain	Multi-Task	# Unique Slot	# Unique Value
Taskmaster	17,304	341,801	19.75	7.87	7	✗	281	66,659
Schema	22,825	463,284	20.3	9.86	17	✓	123	23,889
CamRest676	676	5,488	8.12	10.71	1	✗	4	89
MultiWOZ	10,438	143,048	13.7	15.03	7	✓	46	11,828
SMCalFlow	41,517	170,590	4.11	8.77	4	✓	-	-

Table 5.1: Statistics of task-oriented dialogue datasets. We define a multi-task dialogue as a dialogue involving multiple tasks, e.g. hotel and restaurant booking, while its counterpart handles a single task, e.g. hotel booking. Taskmaster and CamRest676 do not contain any multi-task dialogues.

Guided Dialogue, and pretrain for 1e5 steps. The statistics of the task-oriented dialogue datasets are shown in Table 5.1.

We train each model using 64 TPU chips with 16GB memory each. The pretraining takes around 4 days to complete.

5.6 Experiments

We fine-tune and evaluate the pretrained models on three dialogue datasets: MultiWOZ 2.0, CamRest676 and SMCalFlow (Andreas et al., 2020). In this section we describe the datasets (Section 5.6.1), fine-tuning (Section 5.6.2), decoding (Section 5.6.3) and evaluation metrics (Section 5.6.4). Results are presented in Section 5.7, and analysis and ablation studies in Section 5.8.

5.6.1 Datasets

MultiWOZ[8] is a multi-domain dataset consisting of dialogues annotated with $\mathcal{C}, \mathcal{B}, \mathcal{A}, \mathcal{R}$ in the following seven domains: attraction, hotel, hospital, police, restaurant, train, and taxi. Since its release, MultiWOZ has been one of the most commonly used task-oriented dialogue datasets.

[8] https://cutt.ly/0kuUCRS

CamRest676[9] is annotated similarly to MultiWOZ and consists of dialogues in a single domain: restaurant reservations. Though CamRest676 is smaller than MultiWOZ and predates it, it still provides a widely used benchmark for evaluating task-oriented dialogue models.

SMCalFlow consists of dialogues in four domains: calendar, weather, places, and people. Unlike MultiWOZ and CamRest676, SMCalFlow uses dataflow graphs instead of slot-value pairs to represent belief states and does not annotate dialogue acts. We refer readers to Andreas et al. (2020) for a detailed description of the dataflow representation. We follow Andreas et al. (2020) to convert dataflow graphs into sequences to apply seq2seq models. This dataset is newer and offers fewer prior models to compare with, but we use this dataset to study the robustness of the noisy channel model under different annotation schemas.

We use the public splits for these datasets, where MultiWOZ, CamRest676 and SMCalFlow are split to 8438/1000/1000, 404/136/136 and 32647/3649/5211 dialogues for training, development and testing, respectively. However, since SMCalFlow's test set has not been publicly released, we randomly select 500 dialogues from its training set to tune hyperparameters and use its development set for testing.

Preprocessing: We use the standard preprocessing procedures for each dataset in order to facilitate fair comparison with previous methods.[10] [11] [12] In particular, for MultiWOZ and CamRest676, delexicalization is used to reduce lexical variation, while SMCalFlow does not use delexicalization. During delexicalization, slot values are replaced by generic placeholders based on a pre-defined dictionary. During decoding, following prior work, our dialogue models generate delexicalized

[9] https://cutt.ly/SkuUNfE
[10] https://cutt.ly/TkuU1oM
[11] https://cutt.ly/zkuU0Ht
[12] https://cutt.ly/vkuU9bT

responses. These delexicalized responses are re-lexicalized in post-processing by replacing placeholders with their corresponding slot values based on belief states and database information. Since there is no public code for lexicalization,[13] we implement our own functions for lexicalization with regular expressions, for the purpose of displaying example responses. However, this does not affect reported results, as the standard metrics for MultiWOZ and CamRest676 which we adopt here are calculated using delexicalized responses.

5.6.2 Fine-Tuning

We apply label smoothing with parameter 0.1. Dropout is used on input embeddings and hidden representations, with dropout rate 0.1. The Adam optimizer (Kingma and Ba, 2015) ($b_1 = 0.9$ and $b_2 = 0.999$) is adopted. We use a fixed learning rate 1e-4 with gradient clipping for fine-tuning.

5.6.3 Decoding

We use direct decoding for belief state. For dialogue act and response, we study three decoding methods: direct decoding, noisy channel reranking and noisy channel online decoding. Since all of these decoding methods require choosing k_1 tokens from a categorical distribution during expansion, we compare four methods, top-k maximization, sampling without replacement, top-k sampling, and nucleus sampling. Nucleus sampling with cumulative probability 0.98 performs marginally better and is adopted. We perform a range search with the range $[1, 20]$ on development sets for the beam sizes k_1 and k_2, and we set $k_1, k_2 = 4$, $k_1, k_2 = 15$ and $k_1, k_2 = 4$ for MultiWOZ, CamRest676 and SMCalFlow, respectively. For noisy channel reranking and noisy channel online decoding, a grid search with range $[0, 2]$ is performed for λ_1, λ_2 and λ_3. We set ($\lambda_1 = 0.8$, $\lambda_2 = 1$,

[13] We confirmed this with the dataset authors by email.

$\lambda_3 = 0.8$), ($\lambda_1 = 1.2$, $\lambda_2 = 1.2$, $\lambda_3 = 0.8$) and ($\lambda_1 = 0.4$, $\lambda_2 = 1$, $\lambda_3 = 0.2$) for MultiWOZ, CamRest676 and SMCalFlow, respectively.

5.6.4 Evaluation Metrics

For MultiWOZ and CamRest676, following previous work, we adopt three automatic evaluation metrics: inform, success and BLEU score. Peng et al. (2020a) showed that these metrics are well correlated to human evaluation. The evaluators[14] [15] provided with the datasets are used for calculating these metrics. To calculate the inform score for a dialogue, the evaluator first checks whether certain placeholders (e.g. `[restaurant_name]`) appear in decoded responses. If so, decoded belief states are converted to database queries to retrieve database records. These database records are compared with the records retrieved with ground-truth belief states. The inform score is one if these two sets of database records match. The success score takes all the requestable slots (e.g. postcode, phone number and address) from a decoded response and compares these requestable slots with the ones in the ground-truth response. The success score is one if generated requestable slots coincide with the ground-truth ones. BLEU score (BLEU-4) compares the n-grams of generated responses and human responses, and is a widely used metric in NLP for evaluating text quality. Following Budzianowski et al. (2018), we also calculate a combined score, which is (Inform + Success) / 2 + BLEU. For SMCalFlow, inform and success scores are not applicable since calculation of these scores relies on delexicalization placeholders, and this dataset does not use delexicalization. We use SacreBLEU[16] and TER[17] to directly measure the quality of responses. As prior work on this dataset has

[14] https://cutt.ly/VkuU3FA
[15] https://cutt.ly/MkuU88u
[16] https://cutt.ly/BkuU7dL
[17] https://pypi.org/project/pyter/

Model	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Sequicity (Lei et al., 2018)	66.4	45.3	15.54	71.39
HRED-TS (Peng et al., 2019)	70.0	58.0	17.50	81.50
DSTC8 Track 1 Winner (Ham et al., 2020)	73.0	62.4	16.00	83.50
DAMD (Zhang et al., 2020b)	76.4	60.4	16.60	85.00
SimpleTOD (Hosseini-Asl et al., 2020)	84.4	70.1	15.01	92.26
SOLOIST (Peng et al., 2020a)	85.5	72.9	16.54	95.74
UBAR (Yang et al., 2020)†	**88.2**	**79.5**	16.43	100.28
Randomly Initialized				
Direct decoding (114M)	81.0	54.7	15.12	82.97
Noisy channel reranking (116M)	82.7	57.1	15.29	85.19
Noisy channel online decoding (116M)	82.9	58.9	15.33	86.23
Noisy channel reranking (292M)	82.1	58.1	15.37	85.47
Noisy channel online decoding (292M)	**83.9**	**60.9**	**15.57**	**87.97**
Reddit Pretraining				
Direct decoding (114M)	81.0	69.2	17.06	92.16
Noisy channel reranking (116M)	81.3	70.1	19.01	94.71
Noisy channel online decoding (116M)	81.6	71.1	19.31	95.66
Noisy channel reranking (292M)	82.2	70.9	19.89	96.44
Noisy channel online decoding (292M)	**82.4**	**71.7**	**20.49**	**97.54**
Task-Oriented Pretraining				
Direct decoding (114M)	85.2	72.9	17.00	96.05
Noisy channel reranking (116M)	85.6	73.8	19.38	99.08
Noisy channel online decoding (116M)	85.9	74.8	19.76	100.11
Noisy channel reranking (292M)	86.5	74.9	20.31	101.01
Noisy channel online decoding (292M)	**86.9**	**76.2**	**20.58**	**102.13**

Table 5.2: MultiWOZ test results (end-to-end modeling with generated beliefs) with seq2seq approaches. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding. † This work also reports a combined score of 105.1 with an alternative context and evaluation setting, contributions orthogonal to our work and the other benchmarks reported here.

focused on belief tracking rather than end-to-end response generation, we are the first to use these metrics on this dataset.

We perform significance tests, where we use t-test for inform, success and TER scores and use permutation test for BLEU.

5.7 Results

MultiWOZ: Results on the MultiWOZ test set are shown in Table 5.2. We observe several trends. First, the base noisy channel model (116M) performs better than direct decoding (114M), despite having a similar number of parameters,

Model	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Sequicity (Lei et al., 2018)	92.3	85.3	21.40	110.20
GPT-2 fine-tuned (Wu et al., 2021a)	-	86.2	19.20	-
ARDM (Wu et al., 2021a)	-	**87.1**	25.20	-
SOLOIST (Peng et al., 2020a)	94.7	**87.1**	25.50	116.40
Randomly Initialized				
Direct decoding (114M)	78.1	83.5	21.58	102.38
Noisy channel online decoding (116M)	79.8	84.1	22.83	104.78
Noisy channel online decoding (292M)	**80.9**	**84.9**	**23.19**	**106.09**
Reddit Pretraining				
Direct decoding (114M)	93.3	83.9	23.41	112.01
Noisy channel online decoding (116M)	93.7	84.5	25.14	114.24
Noisy channel online decoding (292M)	**93.9**	**84.7**	**25.38**	**114.68**
Task-Oriented Pretraining				
Direct decoding (114M)	93.4	84.3	24.92	113.77
Noisy channel online decoding (116M)	94.3	85.2	25.98	115.73
Noisy channel online decoding (292M)	**95.4**	**85.3**	**26.89**	**117.24**

Table 5.3: CamRest676 test results (end-to-end modeling with generated beliefs) with seq2seq approaches. Noisy channel reranking performs comparable with noisy channel online decoding, and the results are not shown. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding.

showing that the noisy channel factorization is beneficial for task-oriented dialogue. The large noisy channel setting improves further over the base setting. Second, Reddit pretraining provides benefits over random initialization, validating the use of large open-domain dialogue-genre pretraining for task-oriented dialogue, while the models with a second stage of task-oriented pretraining obtain further improvements. This effect is consistent across both direct and noisy channel decoding. Finally, we observe that online decoding consistently outperforms reranking, indicating the benefits of tighter model integration during decoding.

Our model performs better on combined score than SOLOIST (Peng et al., 2020a), a closely related baseline which pretrains a GPT2-initialized transformer with Taskmaster and Schema-Guided Dialogue and decodes with nucleus sampling.

CamRest676: Results on the CamRest676 test set are shown in Table 5.3. We observe that the base noisy channel model (116M) obtains better results compared to direct decoding (114M), again demonstrating the effectiveness of the noisy

Model	SacreBLEU ↑	TER ↓
Randomly Initialized		
Direct decoding (114M)	51.30	89.13
Online decoding (116M)	53.66	74.18
Online decoding (292M)	**54.39**	**73.18**
Reddit Pretraining		
Direct decoding (114M)	60.68	61.99
Online decoding (116M)	63.29	47.16
Online decoding (292M)	**63.91**	**46.43**
Task-Oriented Pretraining		
Direct decoding (114M)	61.02	59.84
Online decoding (116M)	63.72	46.27
Online decoding (292M)	**64.29**	**45.81**

Table 5.4: SMCalFlow results. Reranking performs worse than online decoding, and the results are not shown. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding.

channel model. Reddit pretraining again provides a large benefit over random initialization for both direct decoding and noisy channel decoding, while task-oriented pretraining provides a further boost. Our model again performs better than SOLOIST.

SMCalFlow: Results on the SMCalFlow development set are shown in Table 5.4. As end-to-end models have not previously been tested on this dataset, we use it to demonstrate that the noisy channel model, which we developed primarily on MultiWOZ, continues to be effective on task-oriented dialogue datasets with different annotation schema. The results are consistent with MultiWOZ and CamRest676. The noisy channel model outperforms the direct model by a large margin, demonstrating that dialogue act annotations are not essential for the noisy channel model, and that it remains effective across diverse dialogue representations.

Reddit pretraining confers a similar large benefit on SMCalFlow as on the other datasets, but we observe that task-oriented pretraining brings only marginal further improvements. This may be due to differences in domain or format between our

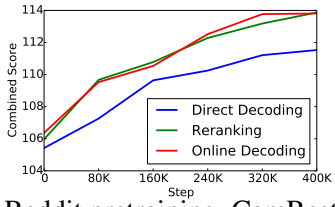
Model	CamRest676	MultiWOZ
Direct decoding	115.17	96.73
Noisy Channel Online Decoding		
Direct + Channel	115.63	98.54
Direct + Source	115.91	99.12
Direct + Length	115.56	97.57
Channel + Source	115.82	99.18
Channel + Length	115.60	98.71
Source + Length	115.62	99.19
All - Direct	115.96	100.89
All - Channel	116.56	100.93
All - Source	116.38	99.92
All - Length	116.52	101.11
All	**116.91**	**102.62**

Table 5.5: Ablation results for model combination on development sets (combined score). Results for reranking are similar and are not shown. 'All', 'Direct' 'Source', and 'Channel' denote no ablation, direct model, source model and channel model, respectively. Rows with '+' are combinations of two sub-modules, while the rows with '-' are combinations of three sub-modules.

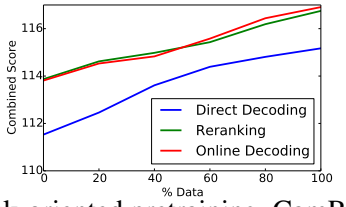
pretraining datasets and SMCalFlow. Alternatively, task-oriented pretraining may help more on task-specific metrics, such as inform and success scores, than on text quality metrics such as BLEU and TER scores. This hypothesis is further supported by the MultiWOZ results in Table 5.2.

5.8 Analysis

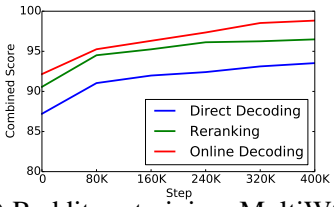
In this section, we use MultiWOZ and CamRest676 to perform ablation studies on the effects of model combination, large-scale pretraining, and sample efficiency; as well as analyzing the runtime requirements of our model and the reasons for its success.



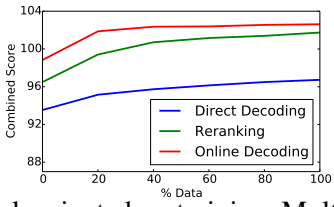
(a) Reddit pretraining, CamRest676



(b) Task-oriented pretraining, CamRest676



(c) Reddit pretraining, MultiWOZ



(d) Task-oriented pretraining, MultiWOZ

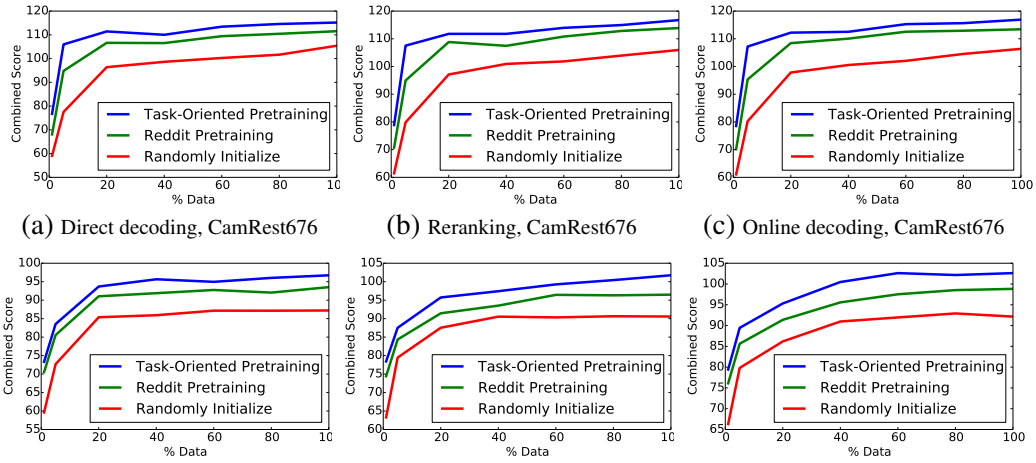
Figure 5.2: Results showing the effect of pretraining scale.

5.8.1 Ablation on Model Combination

Noisy channel decoding involves a combination of four sub-modules, as in Eq. 5.1: the direct model, channel model, language model, and length bias. We perform an ablation study to determine whether all model components are important to the result, using the large model. Results on the development sets of CamRest676 and MultiWOZ are presented in Table 5.5. Note that the ablation is performed after applying the direct model to obtain k_1 expansions at each beam search step for noisy channel online decoding. We find that the combination of all four sub-modules performs the best, followed by combinations of three and then two sub-modules. The results are significant when comparing 'All' and the baselines ($p < 0.01$). This result demonstrates the effectiveness of the noisy channel factorization, and the importance of each model component.

5.8.2 Effect of Pretraining Scale

We investigate the importance of scale for both our pretraining stages. We select different checkpoints for Reddit pretraining, and truncate the two task-oriented



(a) Direct decoding, CamRest676 (b) Reranking, CamRest676 (c) Online decoding, CamRest676

(d) Direct decoding, MultiWOZ (e) Reranking, MultiWOZ (f) Online decoding, MultiWOZ

Figure 5.3: Pretraining improves sample efficiency during fine-tuning.

dialogue datasets for task-oriented pretraining. We fine-tune these models using the full training data of CamRest676 or MultiWOZ. The results of three decoding methods (with the large noisy channel model) on the development sets are shown in Figure 5.2. In Figure 5.2 (a) and (c), the combined scores of all three decoding methods improve with more Reddit pretraining steps, demonstrating the advantage of increasing amounts of data in the open-domain dialogue pretraining stage. In Figure 5.2 (b) and (d), the combined scores further increase with more task-oriented data, confirming that additional task-oriented pretraining data is useful.

5.8.3 Sample Efficiency of Fine-Tuning

We investigate whether pretraining can improve sample efficiency during fine-tuning. We gradually increase the amount of fine-tuning data and evaluate the randomly-initialized, Reddit pretrained and task-oriented pretrained models. The results on the development sets are shown in Figure 5.3. Combined scores increase with more training data under all conditions. Crucially, Reddit pretrained models show better performance with a smaller amount of fine-tuning data than randomly

initialized models, and task-oriented pretrained models better still. We conclude that both our pretraining stages can improve sample efficiency, which is especially important when the target task has little training data.

5.8.4 Decoding Runtime

In Table 5.6, we report the average clock time for decoding one turn (including its belief state, dialogue act and response). Noisy channel reranking is slightly slower compared to direct decoding, with overhead due to the reranking step in Eq. 5.1. Noisy channel online decoding is significantly slower, since it needs to apply Eq. 5.1 at each beam search step. In future work we will investigate ways to improve the efficiency of online decoding.

Model	**CamRest676**	**MultiWOZ**
Direct decoding	**4.89**	**6.48**
Reranking	5.43	6.92
Online decoding	8.73	10.97

Table 5.6: Average decoding time (in seconds) for each turn with different decoding methods.

5.8.5 Decoding Properties

In this section we analyze why the noisy channel model performed better than direct decoding.

Length: In Table 5.7 we show the average length of generated responses. Direct decoding produces shorter responses than the ground truth, confirming that the direct model prefers short and generic responses. Adding a length bias to direct decoding (with lambda tuned on the development sets) produces responses longer than the ground truth, which may be a disadvantage. The noisy channel models produce responses with average length closest to the ground truth.

Model	CamRest676	MultiWOZ
Ground truth	14.50	16.91
Direct decoding	12.07	12.85
Direct decoding + Length	15.98	17.73
Reranking	**15.09**	17.47
Online decoding	15.14	**17.32**

Table 5.7: The average length of responses with different decoding methods (on test set). The value closest to the ground truth is bold.

Zipf: Table 5.8 shows the Zipf scores of responses. We find that the word distributions of responses generated by the noisy channel models are closer to the word distribution of ground-truth responses.

Model	CamRest676	MultiWOZ
Ground truth	1.07	1.22
Direct decoding	0.84	0.91
Reranking	0.87	0.99
Online decoding	**0.89**	**1.03**

Table 5.8: The Zipf scores of responses with different decoding methods (on test set). The value closest to the ground truth is bold.

Repetition: In Table 5.9 we examine the likelihood of falling into repetition loops (Holtzman et al., 2019) for different decoding methods. Repetition loops are rare for all decoding methods, but noisy channel decoding can further decrease their likelihood. The channel model can discount a sequence with a repetition loop, since it conveys less information than a natural sequence of the same length, making it harder to "explain" the context.

Model	CamRest676	MultiWOZ
Direct decoding	0.24	0.31
Reranking	0.12	0.14
Online decoding	**0.08**	**0.11**

Table 5.9: The likelihood (%) of falling into repetition loops for different decoding methods (on test set).

Examples: Some examples of responses are shown in Table 5.10. We observe that noisy channel models decode longer responses compared to direct decoding,

Model	Delexicalized Response	SacreBLEU ↑
	Dialogue: MUL1624, Turn Number: 4	
Ground truth	Sure. The trainid is [train_id]. It leaves at [value_time] and arrives at [value_time]. Would you like to make a reservation?	-
Direct decoding	The travel time is [value_count] minutes.	2.03
Reranking	[train_id] leaves [value_place] at [value_time] and arrives in [value_place] at [value_time]. Would you like me to book that for you?	46.33
Online decoding	[train_id] leaves at [value_time] and arrives at [value_time]. Would you like me to book that?	**60.90**
	Dialogue: MUL1276, Turn Number: 6	
Ground truth	The [restaurant_name] has a postal code of [restaurant_postcode]. Will this work for you? I can book it if you please.	-
Direct decoding	There are [value_count] places. What type of food would you like?	1.63
Reranking	How about [restaurant_name] located at [restaurant_address], postcode [restaurant_postcode].	**24.90**
Online decoding	How about [restaurant_name] located at [restaurant_address], postcode [restaurant_postcode]?	22.00
	Dialogue: MUL1898, Turn Number: 2	
Ground truth	Yes, [attraction_name] is on [attraction_address] and is in the [value_area] side of town. Is there anything else you need to know?	-
Direct decoding	[attraction_name] is located in the [value_area] part of town and has free admission.	27.53
Reranking	[attraction_name] is located in the [value_area] of town at [attraction_address]. The entrance fee is free. Can I help you with anything else?	41.66
Online decoding	[attraction_name] is located in the [value_area] part of town at [attraction_address]. Can I help you with anything else?	**42.38**

Table 5.10: Case study on the responses decoded by direct decoding, noisy channel reranking and noisy channel online decoding. The large noisy channel model is used.

and that the responses can explain their dialogue contexts well to meet users' requirements.

5.9 Summary

In this chapter, we introduced the noisy channel model for task-oriented dialogue. Large-scale pretraining with external context was further adopted to tackle data scarcity in downstream tasks. Extensive experiments on MultiWOZ, CamRest676 and SMCalFlow demonstrated that (1) the noisy channel models significantly outperform direct decoding; (2) models with pretraining improve over randomly-initialized models; (3) the models are robust to different dialogue schema annotations; (4) the noisy channel models can decode responses closer to ground-truth responses than direct decoding.

6 | Conclusions

In this thesis, we aim to better condition on context for natural language processing. Throughout the thesis, we have utilized three different sources of context including local context of each input sequence, global context from training data and external context from the Web. We apply three different sources of context for augmenting NMT datasets, language modelling, and task-oriented dialogue, respectively.

In Chapter 3, we studied utilizing local context for augmenting NMT datasets. We proposed a data augmentation method, which introduces a causal inductive bias. Our method uses language models to answer the counterfactual "What would Y_j have looked like, had $X_i = \hat{x}_i$ instead of x_i, given that Y_j is aligned to X_i, and *all other phrases* $\mathcal{X}_{-i}, \mathcal{Y}_{-j}$ *had been held constant*?". We designed a masked language model and a translation language model to answer the counterfactual. The masked language model takes advantage of source context to replace X_i, while the translation language model takes advantages of both source and target contexts to predict the aligned phrase Y_j.

In Chapter 4, we presented RELATIONLM, a language model that is augmented with relational memory constructed with global context. We showed how to use OpenIE to obtain a relevant knowledge graph for a given corpus and how to combine them with a state-of-the-art language model such as transformer-XL. We demonstrated that our model improves performance and coherence on WikiText-103, WMT19 and enwik8.

In Chapter 5, we introduced the noisy channel model for task-oriented dialogue. Large-scale pretraining with external context was further adopted to tackle data scarcity in downstream tasks. We used a two-stage pretraining strategy. In the

first stage, we applied Reddit pretraining. In the second stage, we used two task-oriented dialogue datasets, Taskmaster and Schema-Guided Dialogue, to specialize the Reddit-pretrained models. We found that the noisy channel models significantly outperform direct decoding. Pre-trained models improve over randomly-initialized models.

For future work, we would like to explore other approaches to retrieve and condition on context. One direction is to use self-play to improve NLP tasks that involve multiple parties, e.g. task-oriented dialogue. Self-play is a natural fit for simulating two-party dialogues, during which a dialogue model converses with a user simulator to achieve a latent goal. We can sample and condition on dialogue goals for generating synthetic dialogues. By sampling dialogue goals from other domains, we can also simulate new scenarios to improve transfer learning, such as cross-domain transfer and single-to-multiple domain transfer.

Another direction is to study better retrieval methods for incorporating global context. In Chapter 4, we used a simple tf-idf scoring method to retrieve global contexts for selected entities. In future work, we will explore dense retrieval methods as it is more powerful to understand polysemous words. However, dense retrieval methods need to retrieve from a large data storage during evaluation, which may significantly increase inference overhead. We will study improving the efficiency of dense retrieval methods in future work.

Beyond local context, global context, and external context pre-downloaded from the Web, we will explore using search engine to obtain contextual information. Search engine maintains more up-to-date information. Besides, search engine services like Google continuously improve their ranking algorithms, providing a powerful tool for information retrieval. To effectively interact with a search engine, the model needs a query producer to generate search queries. We will

study how to efficiently and effectively query and utilize information provided by search engines.

Bibliography

Adiwardana, D., Luong, M., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., and Le, Q. V. (2020). Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977.

Agarwal, O., Ge, H., Shakeri, S., and Al-Rfou, R. (2021). Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3554–3565. Association for Computational Linguistics.

Aharoni, R., Johnson, M., and Firat, O. (2019). Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3874–3884. Association for Computational Linguistics.

Ahn, S., Choi, H., Pärnamaa, T., and Bengio, Y. (2016). A neural knowledge language model. *arXiv preprint arXiv:1608.00318*.

Andreas, J., Bufe, J., Burkett, D., Chen, C., Clausman, J., Crawford, J., Crim, K., DeLoach, J., Dorner, L., Eisner, J., Fang, H., Guo, A., Hall, D., Hayes, K., Hill, K., Ho, D., Iwaszuk, W., Jha, S., Klein, D., Krishnamurthy, J., Lanman, T., Liang, P., Lin, C. H., Lintsbakh, I., McGovern, A., Nisnevich, A., Pauls, A., Petters, D., Read, B., Roth, D., Roy, S., Rusak, J., Short, B., Slomin, D., Snyder, B., Striplin, S., Su, Y., Tellman, Z., Thomson, S., Vorobev, A., Witoszko, I.,

Wolfe, J. A., Wray, A., Zhang, Y., and Zotov, A. (2020). Task-oriented dialogue as dataflow synthesis. *Trans. Assoc. Comput. Linguistics*, 8:556–571.

Angeli, G., Premkumar, M. J. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354. The Association for Computer Linguistics.

Annervaz, K. M., Chowdhury, S. B. R., and Dukkipati, A. (2018). Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 313–322. Association for Computational Linguistics.

Austin, J. L. (1975). *How to do things with words*, volume 88. Oxford University Press.

Bahdanau, D., Bosc, T., Jastrzebski, S., Grefenstette, E., Vincent, P., and Bengio, Y. (2017). Learning to compute word embeddings on the fly. *CoRR*, abs/1706.00286.

Bao, S., He, H., Wang, F., Wu, H., and Wang, H. (2020). PLATO: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*

2020, Online, July 5-10, 2020, pages 85–96. Association for Computational Linguistics.

Bao, S., He, H., Wang, F., Wu, H., Wang, H., Wu, W., Guo, Z., Liu, Z., and Xu, X. (2021). PLATO-2: Towards building an open-domain chatbot via curriculum learning. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2513–2525. Association for Computational Linguistics.

Bapna, A. and Firat, O. (2019). Non-parametric adaptation for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1921–1931. Association for Computational Linguistics.

Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. (2019). Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Barzilay, R. and Lapata, M. (2005). Modeling local coherence: An entity-based approach. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 141–148. The Association for Computer Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Bojar, O. and Tamchyna, A. (2011). Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 330–336. Association for Computational Linguistics.

Bollacker, K. D., Cook, R. P., and Tufts, P. (2007). Freebase: A shared database of structured general human knowledge. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1962–1963. AAAI Press.

Bordes, A., Usunier, N., García-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-Milne, S. (2018). JAX: Composable transformations of Python+NumPy programs.

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 858–867. ACL.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Budzianowski, P. and Vulić, I. (2019). Hello, it's GPT-2 - how can I help you? Towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Budzianowski, P., Wen, T., Tseng, B., Casanueva, I., Ultes, S., Ramadan, O., and Gasic, M. (2018). MultiWOZ - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics.

Byrne, B., Krishnamoorthi, K., Sankar, C., Neelakantan, A., Goodrich, B., Duckworth, D., Yavuz, S., Dubey, A., Kim, K., and Cedilnik, A. (2019). Taskmaster-1: Toward a realistic and diverse dialog dataset. 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 2019, Hong Kong, China, November 3-7, 2019, pages 4515–4524. Association for Computational Linguistics.

Cao, N. D., Izacard, G., Riedel, S., and Petroni, F. (2021). Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Cer, D. M., Diab, M. T., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Chen, W., Chen, J., Qin, P., Yan, X., and Wang, W. Y. (2019). Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709, Florence, Italy. Association for Computational Linguistics.

Cheng, Y., Jiang, L., and Macherey, W. (2019). Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4324–4333. Association for Computational Linguistics.

Cheng, Y., Jiang, L., Macherey, W., and Eisenstein, J. (2020). AdvAug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5961–5970. Association for Computational Linguistics.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Clark, E., Ji, Y., and Smith, N. A. (2018). Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2250–2260. Association for Computational Linguistics.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Currey, A., Barone, A. V. M., and Heafield, K. (2017). Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 148–156. Association for Computational Linguistics.

Dai, A. M. and Le, Q. V. (2015). Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3079–3087.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

de Masson d'Autume, C., Ruder, S., Kong, L., and Yogatama, D. (2019). Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13122–13131.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., and Weston, J. (2019). Wizard of Wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H. (2019). Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648. The Association for Computational Linguistics.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Fadaee, M., Bisazza, A., and Monz, C. (2017). Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver,*

Canada, July 30 - August 4, Volume 2: Short Papers, pages 567–573. Association for Computational Linguistics.

Fan, A., Gardent, C., Braud, C., and Bordes, A. (2021). Augmenting transformers with KNN-based composite memory for dialog. *Trans. Assoc. Comput. Linguistics*, 9:82–99.

Fan, A., Lewis, M., and Dauphin, Y. N. (2018). Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Gao, F., Zhu, J., Wu, L., Xia, Y., Qin, T., Cheng, X., Zhou, W., and Liu, T. (2019). Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5539–5544. Association for Computational Linguistics.

Goutte, C., Cancedda, N., Dymetman, M., and Foster, G. (2009). *Learning machine translation*. Massachusetts Institute of Technology.

Grave, É., Joulin, A., Cissé, M., Grangier, D., and Jégou, H. (2017). Efficient softmax approximation for GPUs. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1302–1310. PMLR.

Grave, E., Joulin, A., and Usunier, N. (2016). Improving neural language models with a continuous cache. *CoRR*, abs/1612.04426.

Guo, D., Tang, D., Duan, N., Zhou, M., and Yin, J. (2018). Dialog-to-action:

Conversational question answering over a large-scale knowledge base. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2946–2955.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. (2020). REALM: Retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Ham, D., Lee, J.-G., Jang, Y., and Kim, K.-E. (2020). End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.

Hao, Y., Dong, L., Wei, F., and Xu, K. (2019). Visualizing and understanding the effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4141–4150. Association for Computational Linguistics.

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

Hayashi, H., Hu, Z., Xiong, C., and Neubig, G. (2020). Latent relation language models. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7911–7918. AAAI Press.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., and Ma, W. (2016). Dual learning for machine translation. In *Advances in Neural Information Process-*

ing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 820–828.

Heck, M., van Niekerk, C., Lubis, N., Geishauser, C., Lin, H., Moresi, M., and Gasic, M. (2020). Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 35–44. Association for Computational Linguistics.

Henderson, M., Thomson, B., and Young, S. J. (2013). Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference, The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 22-24 August 2013, SUPELEC, Metz, France*, pages 467–471. The Association for Computer Linguistics.

Henderson, M., Vulic, I., Gerz, D., Casanueva, I., Budzianowski, P., Coope, S., Spithourakis, G., Wen, T., Mrksic, N., and Su, P. (2019). Training neural response selection for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5392–5404. Association for Computational Linguistics.

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*.

Hennigan, T., Cai, T., Norman, T., and Babuschkin, I. (2020). Haiku: Sonnet for JAX.

Hewitt, J. and Manning, C. D. (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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4129–4138. Association for Computational Linguistics.

Hill, F., Bordes, A., Chopra, S., and Weston, J. (2016). The Goldilocks principle: Reading children's books with explicit memory representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Hixon, B., Clark, P., and Hajishirzi, H. (2015). Learning knowledge graphs for question answering through conversational dialog. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 851–861. The Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Holtzman, A., Buys, J., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Hosseini-Asl, E., McCann, B., Wu, C., Yavuz, S., and Socher, R. (2020). A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Huang, X., Zhang, J., Li, D., and Li, P. (2019). Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 105–113. ACM.

Hutter, M. (2012). The human knowledge compression contest. *URL http://prize. hutter1. net*, 6.

Inan, H., Khosravi, K., and Socher, R. (2016). Tying word vectors and word classifiers: A loss framework for language modeling. *CoRR*, abs/1611.01462.

Izacard, G. and Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Jelinek, F. (1980). Interpolated estimation of Markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice, 1980*.

Ji, Y., Tan, C., Martschat, S., Choi, Y., and Smith, N. A. (2017). Dynamic entity representations in neural language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1830–1839. Association for Computational Linguistics.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

Karpukhin, V., Oguz, B., Min, S., Lewis, P. S. H., Wu, L., Edunov, S., Chen, D., and Yih, W. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Khandelwal, U., Fan, A., Jurafsky, D., Zettlemoyer, L., and Lewis, M. (2020a). Nearest neighbor machine translation. *CoRR*, abs/2010.00710.

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., and Lewis, M. (2020b). Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kiddon, C., Zettlemoyer, L., and Choi, Y. (2016). Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 329–339. The Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Klein, D. and Manning, C. D. (2002). Conditional structure versus conditional estimation in NLP models. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Philadelphia, PA, USA, July 6-7, 2002*, pages 9–16.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 452–457. Association for Computational Linguistics.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. (2019). Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4364–4373. Association for Computational Linguistics.

Krause, B., Kahembwe, E., Murray, I., and Renals, S. (2018). Dynamic evaluation of neural sequence models. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2771–2780. PMLR.

Krause, B., Kahembwe, E., Murray, I., and Renals, S. (2019). Dynamic evaluation of transformer language models. *CoRR*, abs/1904.08378.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Lake, B. M. and Murphy, G. L. (2020). Word meaning in minds and machines. *CoRR*, abs/2008.01766.

Landauer, T. K. and Dumais, S. T. (2008). Latent semantic analysis. *Scholarpedia*, 3(11):4356.

Lee, H., Jo, S., Kim, H., Jung, S., and Kim, T. (2020). SUMBT+LaRL: End-to-end neural task-oriented dialog system with reinforcement learning. *CoRR*, abs/2009.10447.

Lei, W., Jin, X., Kan, M., Ren, Z., He, X., and Yin, D. (2018). Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1437–1447. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020a). 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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020b). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2016). A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Li, X., Tang, M., Guo, F., Li, Y., Cao, K., Song, Q., Wu, X., Sun, S., and Zhou, J. (2022). DDNet: 3D densely connected convolutional networks with feature pyramids for nasopharyngeal carcinoma segmentation. *IET Image Process.*, 16(1):39–48.

Lin, S., Hilton, J., and Evans, O. (2021). TruthfulQA: Measuring how models mimic human falsehoods. *CoRR*, abs/2109.07958.

Lippe, P., Ren, P., Haned, H., Voorn, B., and de Rijke, M. (2020). Diversifying task-oriented dialogue response generation with prototype guided paraphrasing. *CoRR*, abs/2008.03391.

Liu, B. and Lane, I. R. (2018). End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 2-4, 2018, Student Research Workshop*, pages 67–73. Association for Computational Linguistics.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019a). Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-*

HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 1073–1094. Association for Computational Linguistics.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Liu, Q., Yu, L., Rimell, L., and Blunsom, P. (2021b). Pretraining the noisy channel model for task-oriented dialogue. *Transactions of the Association for Computational Linguistics*, 9:657–674.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., and Wang, P. (2020). K-BERT: Enabling language representation with knowledge graph. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2901–2908. AAAI Press.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Logan, R. L., Liu, N. F., Peters, M. E., Gardner, M., and Singh, S. (2019). Barack's wife Hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5962–5971. Association for Computational Linguistics.

Luong, T., Socher, R., and Manning, C. D. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Sev-*

enteenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013, pages 104–113. ACL.

Maddison, C. and Tarlow, D. (2017). Gumbel machinery.

Maddison, C. J., Tarlow, D., and Minka, T. (2014). A* sampling. In *Advances in Neural Information Processing Systems*, pages 3086–3094.

Mehri, S., Srinivasan, T., and Eskénazi, M. (2019). Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 165–177. Association for Computational Linguistics.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T., Yih, W., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751. The Association for Computational Linguistics.

Minervini, P., Bosnjak, M., Rocktäschel, T., Riedel, S., and Grefenstette, E. (2020). Differentiable reasoning on large knowledge bases and natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*

2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 5182–5190. AAAI Press.

Moon, S., Shah, P., Kumar, A., and Subba, R. (2019). OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 845–854. Association for Computational Linguistics.

Mrksic, N., Séaghdha, D. Ó., Wen, T., Thomson, B., and Young, S. J. (2017). Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1777–1788. Association for Computational Linguistics.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Neelakantan, A., Yavuz, S., Narang, S., Prasad, V., Goodrich, B., Duckworth, D., Sankar, C., and Yan, X. (2019). Neural assistant: Joint action prediction, response generation, and latent knowledge reasoning. *CoRR*, abs/1910.14613.

Neubig, G., Watanabe, T., Sumita, E., Mori, S., and Kawahara, T. (2011). An unsupervised model for joint phrase alignment and extraction. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 632–641. The Association for Computer Linguistics.

Nguyen, T. Q. and Salazar, J. (2019). Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation, IWSLT 2019, Hong Kong, November 2-3, 2019*. Association for Computational Linguistics.

Norouzi, M., Bengio, S., Chen, Z., Jaitly, N., Schuster, M., Wu, Y., and Schuurmans, D. (2016). Reward augmented maximum likelihood for neural structured prediction. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1723–1731.

Nouri, E. and Hosseini-Asl, E. (2018). Toward scalable neural dialogue state tracking model. *CoRR*, abs/1812.00899.

Nye, M., Tessler, M. H., Tenenbaum, J. B., and Lake, B. M. (2021). Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. *CoRR*, abs/2107.02794.

Oberst, M. and Sontag, D. A. (2019). Counterfactual off-policy evaluation with Gumbel-max structural causal models. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4881–4890. PMLR.

Ostendorff, M., Bourgonje, P., Berger, M., Schneider, J. M., Rehm, G., and Gipp, B. (2019). Enriching BERT with knowledge graph embeddings for document classification. In *Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, Erlangen, Germany, October 9-11, 2019*.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. (2016). The LAMBADA dataset:

Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Parvez, M. R., Chakraborty, S., Ray, B., and Chang, K. (2018). Building language models for text with named entities. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2373–2383. Association for Computational Linguistics.

Pearl, J. (2003). Causality: Models, reasoning, and inference. *Econometric Theory*, 19:675–685.

Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal inference in statistics: A primer*. John Wiley & Sons.

Pei, J., Ren, P., and de Rijke, M. (2019). A modular task-oriented dialogue system using a neural mixture-of-experts. *CoRR*, abs/1907.05346.

Peng, B., Li, C., Li, J., Shayandeh, S., Liden, L., and Gao, J. (2020a). SOLOIST: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *CoRR*, abs/2005.05298.

Peng, B., Zhu, C., Li, C., Li, X., Li, J., Zeng, M., and Gao, J. (2020b). Few-shot natural language generation for task-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 172–182. Association for Computational Linguistics.

Peng, S., Huang, X., Lin, Z., Ji, F., Chen, H., and Zhang, Y. (2019). Teacher-student framework enhanced multi-domain dialogue generation. *CoRR*, abs/1908.07137.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Peters, M. E., Neumann, M., IV, R. L. L., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. (2019). Knowledge enhanced contextual word representations. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 43–54. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P. S. H., Bakhtin, A., Wu, Y., and Miller, A. H. (2019). Language models as knowledge bases? 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Press, O. and Wolf, L. (2017). Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 157–163. Association for Computational Linguistics.

Provilkov, I., Emelianenko, D., and Voita, E. (2020). BPE-Dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1882–1892. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ramachandran, P., Liu, P. J., and Le, Q. V. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen,*

Denmark, September 9-11, 2017, pages 383–391. Association for Computational Linguistics.

Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, volume 242, pages 29–48. Citeseer.

Rastogi, A., Hakkani-Tür, D., and Heck, L. P. (2017). Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*, pages 561–568. IEEE.

Rastogi, A., Zang, X., Sunkara, S., Gupta, R., and Khaitan, P. (2020). Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, February 7-12, 2020*, pages 8689–8696. AAAI Press.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, Boulder, Colorado, USA, June 4-5, 2009*, pages 147–155. ACL.

Raux, A., Langner, B., Bohus, D., Black, A. W., and Eskénazi, M. (2005). Let's go public! Taking a spoken dialog system to the real world. In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*, pages 885–888. ISCA.

Rebele, T., Suchanek, F. M., Hoffart, J., Biega, J., Kuzey, E., and Weikum, G. (2016). YAGO: A multilingual knowledge base from Wikipedia, Wordnet, and Geonames. In *The Semantic Web - ISWC 2016 - 15th International Seman-*

tic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II, volume 9982 of *Lecture Notes in Computer Science*, pages 177–185.

Reddy, S., Chen, D., and Manning, C. D. (2019). CoQA: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Reif, E., Yuan, A., Wattenberg, M., Viégas, F. B., Coenen, A., Pearce, A., and Kim, B. (2019). Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8592–8600.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kociský, T., and Blunsom, P. (2016). Reasoning about entailment with neural attention. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y., and Weston, J. (2021). Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 300–325. Association for Computational Linguistics.

Schlichtkrull, M. S., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Seneff, S. and Polifroni, J. (2000). Dialogue management in the mercury flight reservation system. In *ANLP-NAACL 2000 Workshop: Conversational Systems*.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 371–376. The Association for Computer Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016c). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., and Wu, H. (2019). ERNIE: Enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Tanaka, K., Takayama, J., and Arase, Y. (2019). Dialogue-act prediction of future responses based on conversation history. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*, pages 197–202. Association for Computational Linguistics.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics.

Thulke, D., Daheim, N., Dugast, C., and Ney, H. (2021). Efficient retrieval augmented generation from unstructured knowledge for task-oriented dialog. *CoRR*, abs/2102.04643.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Tseng, B., Cheng, J., Fang, Y., and Vandyke, D. (2020). A generative model for joint natural language understanding and generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1795–1807. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural*

Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

Verga, P., Sun, H., Soares, L. B., and Cohen, W. W. (2021). Adaptable and interpretable neural memory over symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3678–3691. Association for Computational Linguistics.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019a). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Wang, C., Liu, X., and Song, D. (2020a). Language models are open knowledge graphs. *CoRR*, abs/2010.11967.

Wang, H., Zhang, F., Xie, X., and Guo, M. (2018a). DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1835–1844. ACM.

Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., and Guo, M. (2019b). Multi-task feature learning for knowledge graph enhanced recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2000–2010. ACM.

Wang, L., Li, Y., Aslan, O., and Vinyals, O. (2021a). WikiGraphs: A Wikipedia text - knowledge graph paired dataset. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 67–82, Mexico City, Mexico. Association for Computational Linguistics.

Wang, Q., Pan, X., Huang, L., Zhang, B., Jiang, Z., Ji, H., and Knight, K. (2018b). Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 10–21. Association for Computational Linguistics.

Wang, W., Bi, B., Yan, M., Wu, C., Xia, J., Bao, Z., Peng, L., and Si, L. (2020b). StructBERT: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., and Tang, J. (2021b). KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguistics*, 9:176–194.

Wang, X., Pham, H., Dai, Z., and Neubig, G. (2018c). Switchout: An efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 856–861. Association for Computational Linguistics.

Wen, T., Miao, Y., Blunsom, P., and Young, S. J. (2017a). Latent intention dialogue models. *CoRR*, abs/1705.10229.

Wen, T., Vandyke, D., Mrksic, N., Gasic, M., Rojas-Barahona, L. M., Su, P., Ultes, S., and Young, S. J. (2017b). A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the Euro-*

pean Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers, pages 438–449. Association for Computational Linguistics.

Wu, C., Hoi, S. C. H., Socher, R., and Xiong, C. (2020). TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 917–929. Association for Computational Linguistics.

Wu, C., Madotto, A., Hosseini-Asl, E., Xiong, C., Socher, R., and Fung, P. (2019). Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819. Association for Computational Linguistics.

Wu, Q., Zhang, Y., Li, Y., and Yu, Z. (2021a). Alternating recurrent dialog model with large-scale pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1292–1301. Association for Computational Linguistics.

Wu, Z., Galley, M., Brockett, C., Zhang, Y., Gao, X., Quirk, C., Koncel-Kedziorski, R., Gao, J., Hajishirzi, H., Ostendorf, M., and Dolan, B. (2021b). A controllable model of grounded response generation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14085–14093. AAAI Press.

Xia, M., Kong, X., Anastasopoulos, A., and Neubig, G. (2019). Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5786–5796. Association for Computational Linguistics.

Yang, B. and Mitchell, T. M. (2019). Leveraging knowledge bases in LSTMs for improving machine reading. *CoRR*, abs/1902.09091.

Yang, Y., Li, Y., and Quan, X. (2020). UBAR: Towards fully end-to-end task-oriented dialog systems with GPT-2. *CoRR*, abs/2012.03539.

Yang, Z., Blunsom, P., Dyer, C., and Ling, W. (2017). Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1850–1859. Association for Computational Linguistics.

Yao, L., Mao, C., and Luo, Y. (2019). KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Yasunaga, M., Ren, H., Bosselut, A., Liang, P., and Leskovec, J. (2021). QA-GNN: Reasoning with language models and knowledge graphs for question answering. *CoRR*, abs/2104.06378.

Yee, K., Dauphin, Y. N., and Auli, M. (2019). Simple and effective noisy channel modeling for neural machine translation. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5695–5700. Association for Computational Linguistics.

Yogatama, D., de Masson d'Autume, C., and Kong, L. (2021). Adaptive semi-parametric language models. *Trans. Assoc. Comput. Linguistics*, 9:362–373.

You, Y., Li, J., Reddi, S. J., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C. (2020). Large batch optimization for deep learning: Training BERT in 76 minutes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yu, L., Blunsom, P., Dyer, C., Grefenstette, E., and Kociský, T. (2017). The neural noisy channel. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yu, L., Sartran, L., Stokowiec, W., Ling, W., Kong, L., Blunsom, P., and Dyer, C. (2020). Better document-level machine translation with Bayes' rule. *Transactions of the Association for Computational Linguistics*, 8:346–360.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. (2019). Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9051–9062.

Zhang, F., Yuan, N. J., Lian, D., Xie, X., and Ma, W. (2016). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 353–362. ACM.

Zhang, J., Hashimoto, K., Wu, C., Wang, Y., Yu, P. S., Socher, R., and Xiong, C. (2020a). Find or classify? Dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics, *SEM@COLING 2020, Barcelona, Spain (Online), December 12-13, 2020*, pages 154–167. Association for Computational Linguistics.

Zhang, M. and Chen, Y. (2018). Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5171–5181.

Zhang, S., Tay, Y., Yao, L., and Liu, Q. (2019). Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2731–2741.

Zhang, Y., Dai, H., Kozareva, Z., Smola, A. J., and Song, L. (2018). Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6069–6076. AAAI Press.

Zhang, Y., Ou, Z., and Yu, Z. (2020b). Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence,*

EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 9604–9611. AAAI Press.

Zhang, Y., Sun, S., Galley, M., Chen, Y., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, B. (2020c). DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.

Zhao, T., Xie, K., and Eskénazi, M. (2019). Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1208–1218. Association for Computational Linguistics.

Zhila, A., Yih, W., Meek, C., Zweig, G., and Mikolov, T. (2013). Combining heterogeneous models for measuring relational similarity. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 1000–1009. The Association for Computational Linguistics.

Zhou, L. and Small, K. (2019). Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *CoRR*, abs/1911.06192.

Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W. X., Yu, D., and Wu, H. (2018). Multi-turn response selection for chatbots with deep attention match-

ing network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1118–1127. Association for Computational Linguistics.

Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.