City Research Online

City, University of London Institutional Repository

CITY, UNIVERSITY OF LONDON

DOCTORAL THESIS

Towards Knowledge-Grounded Natural Language Understanding and Generation

---

Author:

Chenxi Jia WHITEHOUSE

Supervisors:

Dr Tillman WEYDE

Dr Pranava MADHYASTHA

School of Science and Technology

Department of Computer Science

April, 2024

Abstract

Towards Knowledge-Grounded Natural Language Understanding and Generation

by Chenxi Jia WHITEHOUSE

This thesis investigates how natural language understanding and generation with transformer models can benefit from grounding the models with knowledge representations. Currently, the most prevailing paradigm for training language models is through pre-training on abundant raw text data and fine-tuning on downstream tasks. Although language models continue to advance, especially the recent trend of Large Language Models (LLMs) such as ChatGPT, there seem to be limits to what can be achieved with text data alone and it is desirable to study the impact of applying and integrating rich forms of knowledge representation to improve model performance.

The most widely used form of knowledge for language modelling is structured knowledge in the form of triples consisting of entities and their relationships, often in English. This thesis explores beyond this conventional approach and aims to address several key questions:

- Can knowledge of entities extend its benefits beyond entity-centric tasks such as entity linking?

- How can we faithfully and effectively extract such structured knowledge from raw text, especially noisy web text?

- How do other types of knowledge, beyond structured knowledge, contribute to improving NLP tasks?

To this end, we study various tasks including multimodal and multilingual applications and consider a wide spectrum of knowledge, structured knowledge that is typically represented as triples with entities and their relations, and unstructured knowledge including parametric knowledge preserved in language models, knowledge distilled from Large Language Models, etc.

Knowledge-grounding with structured knowledge.

We begin by investigating the integration of structured knowledge into language models. Knowledge of entities has shown benefits for entity-centric tasks such as entity linking and relation extraction, however, most studies have been limited to monolingual settings. We expand knowledge-grounding with structured knowledge, specifically entities, in two directions of research.

Firstly, we study whether knowledge of entities can benefit real-world fake news detection. We hypothesise that the world knowledge embedded in entities can contribute to assessing the truthfulness of news statements. Evaluation of various knowledge integration approaches on distinct datasets reveals that knowledge-enhanced language models improve fake news detection when incorporated with a relevant and up-to-date knowledge base.

The second direction expands beyond English and focuses on multilingual entities. We introduce `EntityCS`, where we first construct a code-switched (CS) training corpus from Wikipedia, by switching entities in English to their counterparts in other languages. Then we intermediate-train a pretrained multilingual model on this corpus for joint masked language modelling and entity prediction. Subsequent fine-tuning of the model on entity-centric downstream tasks consistently improves zero-shot cross-lingual transferability, demonstrating the benefit of integrating knowledge of multilingual entities.

Extracting structured knowledge from web text.

We continue by studying effective, faithful, and robust extraction of structured knowledge from web text. Most existing information extraction (IE) datasets are constrained to Wikipedia articles, and models trained on such a rich factual text corpus

show poor performance when applied to more noisy text from the web. To address these challenges, we introduce `WebIE`, a new dataset that takes raw sentences as input and structured triples as output. `WebIE` emphasises data quality by introducing negative examples and undergoing rigorous human annotation. We also propose faithful generative information extraction pipelines. Our experiments with entity planning training and prefix-trie decoding show improvement in accurately extracting knowledge on the web.

Knowledge-grounding beyond structured knowledge.

To address our last research question, we study the impact of a broader sense of knowledge, including parametric knowledge (knowledge stored in the latent parameters of the models) derived from a model's self-explanations and knowledge distilled from LLMs via data augmentation.

We expand the application to multimodal language models and study knowledge-intensive visual question answering (VQA). We introduce a unified approach for fine-tuning multimodal models for jointly generating answers and explanations. Our experiments demonstrate enhancement in both answer accuracy and explanation quality.

Lastly, as LLMs continue to advance in performance and size, we explore the utility of distilling commonsense knowledge from general-purpose LLMs to benefit smaller task-specific models. We prompt various LLMs to generate diverse examples on several challenging and scarce multilingual commonsense datasets. This augmentation shows consistent enhancements on fine-tuned smaller models, shedding light on data augmentation strategies for scenarios with limited training data.

In summary, this thesis explores the role of knowledge grounding in natural language understanding and generation across a broad spectrum of tasks. We found that incorporating relevant and up-to-date knowledge of entities benefits fake news detection, and entity-focused code-switching significantly enhances zero-shot cross-lingual transfer on entity-centric tasks. In terms of effective and faithful approaches

to extracting structured knowledge, our study found that integrating negative examples and training with entity planning significantly improves performance. Additionally, we established that other general forms of knowledge, such as parametric and distilled knowledge, enhance multimodal and multilingual knowledge-intensive tasks. This research shows the tangible benefits of diverse knowledge integration and motivates further exploration in this direction.

Acknowledgements

I want to express my sincere gratitude to my PhD supervisors for their consistent support and guidance throughout my doctoral journey. Special thanks to Dr Tillman Weyde and Dr Nikos Komninos for selecting me for the PhD studentship. I am genuinely thankful to Dr Pranava Madhyastha, who joined my PhD supervision later on and inspired me to explore and gain a deeper understanding of the field of NLP.

I deeply appreciate all those who supported me during my research internships, which were an integral part of my PhD studies. It was an honour to intern at some of the world's leading research institutes and learn from experienced researchers. Warm thanks to Dr Fenia Christopoulou, my mentor during my first research internship at Huawei, whose dedication and hard work have consistently motivated me. I am grateful to Dr Clara Vania at Amazon for her encouragement and genuine support for my future endeavours. Furthermore, I would also like to extend my gratitude to Dr Fantine Huot and Dr Jasmijn Bastings for the opportunity to intern at Google DeepMind, a dream come true, and to Dr Mostafa Dehghani and Prof. Mirella Lapata for their invaluable assistance and guidance throughout the journey.

I extend a warm thank you to Dr Alham Fikri Aji for his support, from Amazon to MBZUAI, and for encouraging me to expand my research collaboration.

Lastly, my sincere thanks go to Jaime, who has been a steadfast source of support in my life, believing in me and standing by my side through both the best and the most challenging times.

Contents

List of Figures

List of Tables

Dedicated to my parents and Jaime.

Chapter 1

Introduction

Natural Language Processing (NLP) has witnessed significant progress in recent years, particularly since the emergence of transformer-based (Vaswani et al., 2017) language models. Most representative transformer models include BERT (Devlin et al., 2019), RoBERTa (Yinhan Liu et al., 2019), BART (M. Lewis et al., 2020), T5 (Raffel et al., 2020a), to name a few, which are typically pre-trained on raw, unlabelled textual inputs with objectives such as Masked Language Modelling (Devlin et al., 2019), and then fine-tuned on labelled task-specific downstream tasks. Although this pre-training and fine-tuning paradigm has proven effective and achieved new start-of-the-art performance on various NLP tasks (Devlin et al., 2019), it presents limitations in adapting to the ever-evolving world knowledge (Zhengyan Zhang et al., 2019; Weijie Liu et al., 2020). Figure 1.1 shows an example where the knowledge base is crucial for the model generation when the parametric knowledge stored within the model parameters becomes stale.

To address this, a growing need has emerged to explore the integration of diverse knowledge representations and grounding into the language models to enhance their capabilities in tasks particularly requiring intensive knowledge (P. Lewis et al., 2020; Izacard et al., 2023) or in-depth context understanding (T.-Y. Chang et al., 2020; Bauer and Bansal, 2021).

The most commonly used knowledge for such integration is structured knowledge, typically represented in knowledge bases consisting of entities and their relationships (X. Chen, S. Jia, and Y. Xiang, 2020; C. Zhu et al., 2022). Knowledge
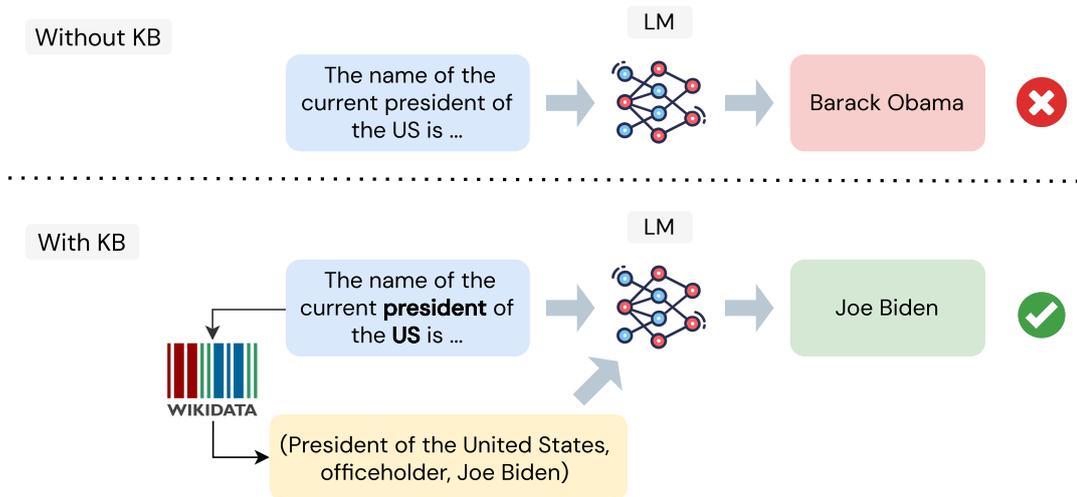
FIGURE 1.1: Illustration of the role of knowledge bases in text generation: When the parametric knowledge stored in the language model becomes outdated (such as with GPT-2, trained in 2017), it is important to incorporate up-to-date knowledge bases for deriving correct answers, especially when using the frozen model at inference time.

bases, such as Wikidata,[1] provide a rich source of human-curated factual knowledge, which can be used as a complement to unlabelled raw text. Considerable research has focused on incorporating structured knowledge, especially entity-based knowledge (i.e., knowledge of entities), into language models (Zhengyan Zhang et al., 2019; Peters et al., 2019; R. Wang et al., 2021; Xiaozhi Wang et al., 2021). Although these prior works have proven the advantages of knowledge integration in specific tasks such as entity linking, named entity recognition, etc., the potential of a wider spectrum of knowledge representations (e.g., unstructured knowledge) and applications (e.g., beyond entity-centric tasks) remains relatively under-explored. This motivates us to conduct a more extensive investigation into the following research questions.

1.1 Research Questions

This thesis aims to address the research questions outlined in the following three aspects.

---

[1] https://www.wikidata.org/wiki/Wikidata:Main_Page

Expanding the Utilisation of Structured Knowledge

The first question we study is: *Can the advantages of entity-based knowledge be extended to multilingual setups and beyond entity-centric tasks?*

Prior work such as KnowBert (Peters et al., 2019), ERNIE (Zhengyan Zhang et al., 2019), K-BERT (Weijie Liu et al., 2020), etc. have demonstrated the success of integrating knowledge of entities into language model pre-training. However, their predominant focuses are limited to monolingual language models (specifically, English only), and downstream tasks that are exclusively related to entities, such as entity linking and named entity recognition. This thesis broadens the application scope of structured knowledge in two ways: (i) we evaluate the effectiveness of knowledge-enhanced language models on the more complex task of fake news detection (Chapter 3), and (ii) we propose the use of multilingual entity knowledge in an entity-centric code-switching method, ENTITYCS, to improve cross-lingual transferability on low-resource languages (Chapter 4).

Effective Extraction of Structured Knowledge

We continue to explore: *How can structured knowledge be extracted effectively and accurately from diverse sources, particularly noisy web text?*

Prior work introduced above as well as our first studies (Whitehouse et al., 2022; Whitehouse, Christopoulou, and Iacobacci, 2022) demonstrate the benefit of structured knowledge in various NLP tasks. Yet the challenge lies in how to effectively obtain such knowledge, which is essential to adding new facts, keeping the relevance and accuracy of existing facts, etc. Hence, there is a compelling need to develop models capable of automatically extracting structured knowledge from vast textual sources (Y. Yao et al., 2019; Y. Yao et al., 2021; Ormandi et al., 2021), which requires high-quality information extraction datasets. The majority of existing knowledge or information extraction datasets that are used to train such models are constructed based on clean and fact-rich resources such as Wikipedia (Trisedya et al., 2019; Huguet Cabot and Navigli, 2021; Seganti et al., 2021). As a result, models

trained on them may encounter difficulties when applied to noisy web text, as revealed in our preliminary studies. Therefore, this thesis addresses the critical challenge by proposing a more generalised dataset, WEBIE, as well as modelling approaches that are suitable for knowledge extraction from the web text (Chapter 5).

Exploration of Diverse Knowledge Forms

The final question explored in this thesis focuses on: *How can broader forms of unstructured knowledge contribute to enhancing language models?*

The preceding two directions focus on the investigation of structured knowledge. However, there exists a diverse range of unstructured knowledge representations, such as those found in raw text (C. Zhu et al., 2022; J. Z. Pan et al., 2023), knowledge derived from the language models after pre-training (i.e., stored in the latent parameters) (Petroni et al., 2019; Zhiyuan Zhang et al., 2020; Xintao Wang et al., 2022; Neeman et al., 2023). In this part of the thesis, we aim to broaden the scope beyond structured knowledge and explore unstructured knowledge, including parametric knowledge (Neeman et al., 2023) embedded within language models, through grounded answer and explanation generation in knowledge-intensive Visual Question Answering (VQA) (Chapter 6), and knowledge distilled from Large Language Models via data augmentation (Chapter 7). Emphasis is placed on the application of these studies in multimodal and multilingual use cases.

1.2   Structure of the Thesis

The thesis begins with an introduction that outlines the background of transformers, knowledge in NLP, and knowledge-enhanced language models in Chapter 2, which sets out the foundation for all the subsequent studies conducted. The main body of the thesis is structured around five chapters, each focusing on a published conference paper.

The first focus is on knowledge-grounding with entities and structured knowledge. Chapter 3 concentrates on knowledge-enhanced language models for the application of fake news detection. We evaluate the effectiveness of various approaches

that incorporate entity knowledge in language models on fake news detection accuracy, covering news datasets in different domains (i.e., politics and COVID-19) with different linguistic features, where we find the benefit of incorporating knowledge that is relevant and up-to-date.

Chapter 4 focuses on studying the effectiveness of multilingual entity knowledge in zero-shot cross-lingual transfer. We propose ENTITYCS, where we utilise entity-centric code-switching with Wikipedia and Wikidata knowledge base to intermediate train a cross-lingual pre-trained language model. Subsequent fine-tuned models on entity-centric tasks, e.g., named entity recognition, slot filling, and fact retrieval, demonstrate strong performance compared to the baseline.

Chapter 5 addresses the extraction of faithful and robust structured knowledge from the web domain. Information extraction is essential for knowledge base construction and population, which also provides reliable knowledge sources for training knowledge-enhanced models. In this chapter, we introduce the collection of a new dataset, WEBIE, via crowdsourcing and propose various joint training strategies for mitigating the hallucination issues in generative information extraction.

The subsequent two chapters explore knowledge-grounding beyond structured knowledge. Chapter 6 expands the application to multimodal language models and studies knowledge-intensive VQA, utilising parametric knowledge derived from a model's self-explanations. We propose a novel direction of jointly generating answers and explanations with multimodal generative models making use of artificial special tokens, enabling the distinction of different tasks and datasets while learning shared semantics among different tasks.

As LLMs become ever more powerful, in Chapter 7, we explore the utility of distilling commonsense knowledge from general-purpose LLMs to benefit smaller task-specific models. We prompt various LLMs to generate diverse examples on several challenging and scarce multilingual commonsense datasets. This augmentation shows consistent enhancements on fine-tuned smaller models, compared to those trained on limited human-created data.

In Chapter 8, we provide a summary of the thesis with insights and key takeaways, as well as motivations for promising future research directions.

1.3   Publications

The main work proposed in this thesis has been published as conference papers. Chapter 3-7 contain existing, improved or extended results to the following publications:

1. Evaluation of Fake News Detection with Knowledge-Enhanced Language Models (Whitehouse et al., 2022)

   - Main content included in Chapter 3

   - *Proceedings of the Sixteenth International AAAI Conference on Web and Social Media (**AAAI-ICWSM** 2022)*

2. EntityCS: Improving Zero-Shot Cross-lingual Transfer with Entity-Centric Code Switching (Whitehouse, Christopoulou, and Iacobacci, 2022)

   - Main content included in Chapter 4

   - *Findings of the Association for Computational Linguistics: **EMNLP** 2022*

   - Work conducted as Research Intern at Huawei Noah's Ark Lab, London, United Kingdom, 2021

3. Towards a Unified Model for Generating Answers and Explanations in Visual Question Answering (Whitehouse, Weyde, and Madhyastha, 2023)

   - Main content included in Chapter 6

   - *Findings of the Association for Computational Linguistics: **EACL** 2023*

4. WebIE: Faithful and Robust Information Extraction on the Web (Whitehouse et al., 2023b)

   - Main content included in Chapter 5

   - *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (**ACL** 2023)*

   - Work conducted as Research Intern at Amazon Alexa AI, Cambridge, United Kingdom, 2022

5. LLM-powered Data Augmentation for Enhanced Cross-lingual Performance (Whitehouse, Choudhury, and Aji, 2023)

   - Main content included in Chapter 7

   - *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (**EMNLP** 2023)*

   - Work conducted in collaboration with MBZUAI and Microsoft

Other Publications during the PhD

Throughout the course of my doctoral studies, there are several additional papers that are not incorporated in the thesis, from my collaboration with researchers from Google DeepMind and MBZUAI.

6. Low-Rank Adaptation Multilingual Summarization: An Empirical Study (Whitehouse et al., 2023a)

   - *Findings of the Association for Computational Linguistics: **NAACL** 2024*

   - Work conducted as Research Intern at Google DeepMind, Amsterdam, Netherlands, 2023

7. M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection (Yuxia Wang et al., 2024)

   - *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (**EACL** 2024)*

   - Work conducted in collaboration with MBZUAI, 2023

8. Inducing Generalization across Languages and Tasks using Featurized Low-Rank Mixtures (C.-C. Lin et al., 2024)

   - *Preprint*

   - Work conducted as Research Intern at Google DeepMind, Amsterdam, Netherlands, 2023

Chapter 2

Background: Transformer, Knowledge, Knowledge-Enhanced PLMs

This section provides a background overview of three fundamental aspects: (i) the Transformer. All the modelling approaches in this thesis are based on the transformer architecture, (ii) knowledge sources and knowledge representation, and (iii) knowledge-enhanced Pre-trained Language Models (PLMs). These topics collectively establish the foundation for the core focus of this thesis: knowledge-grounding in language models.

2.1 The Transformer

The transformer architecture, proposed by Vaswani et al. (2017), has revolutionised the NLP field. Compared to its predecessors, including RNN and LSTM (Hochreiter and Schmidhuber, 1997), transformers distinguish themselves in the following key aspects. Firstly, transformers employ an attention mechanism to alleviate the limitations associated with long-term dependencies. Secondly, unlike previous sequential models, transformers can process input sequences in parallel, making them more efficient. Transformers are also highly scalable. The stacked architecture can be easily scaled to be optimised for larger datasets and more complex tasks by adding more layers or units, which is a key factor in the development of recent powerful LLMs (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020a; Anil et al., 2023).

Thanks to their advantages, transformers have become the cornerstone of the current NLP field. All the models included in this thesis are based on transformers. We first provide an overview of the transformer architecture and then introduce several representative model categories, including encoder-only models, decoder-only models, and sequence-to-sequence models. Finally, we briefly touch upon multimodal vision-language models.

2.1.1 Transformer Architecture

The originally proposed transformer architecture, as illustrated in Figure 2.1, follows a sequence-to-sequence structure, which takes a sequence as input and outputs a new sequence. Its key innovation is the self-attention mechanism, which allows the model to weigh the relevance of each word in a sequence to every other word, enabling the capture of long-range dependencies and context in text data.

At its core, the transformer architecture contains a stack of encoder layers (left) and decoder layers (right). Both the encoder stack and decoder stack have their corresponding Embedding layers to process the input, represented by the `Input Embedding` and `Output Embedding` layers in Figure 2.1. Within each layer, *Multi-Head Attention* is applied independently to all subwords or tokens in the input sequence, and the outputs are passed through `Feed Forward` neural network layers. In the end, the output of the last layer of the decoder stack is fed to a `Linear` layer and a `Softmax` layer to obtain the final output sequence.

2.1.1.1 Attention Mechanism

Given a set of key-value pair vectors and a query vector, attention is a technique to compute a weighted sum of the values, dependent on the association of the query and the corresponding keys, i.e., the query attends to the values (Vaswani et al., 2017).

The core innovation of the Transformer is the incorporation of the attention mechanism. Unlike traditional models that linearly process sequences, the self-attention
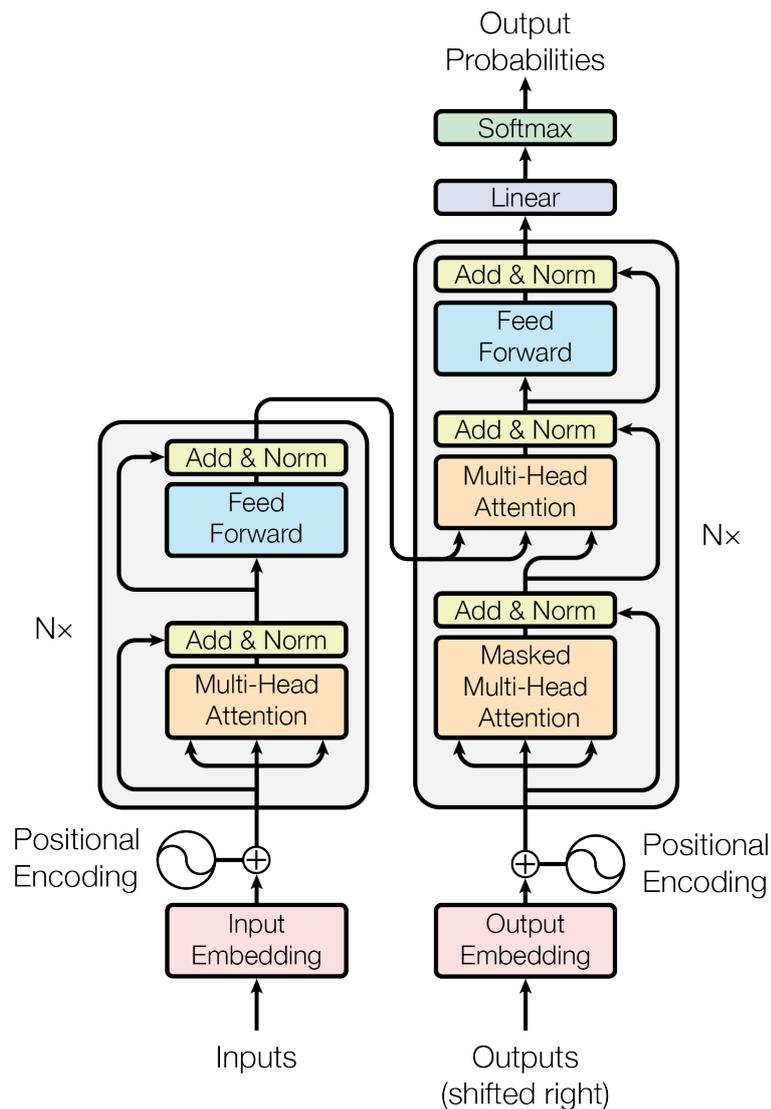
FIGURE 2.1: The Transformer Model Architecture. It is composed of a stack of
N encoders (on the left) and *N* decoders (on the right). Each sublayer comprises
a multi-head attention layer and a feed-forward layer. Layer Normalisation and
Residual connections are applied after each layer. Source: Vaswani et al. (2017).

mechanism allows the model to weigh the significance of different words in a se-
quence when making predictions for a particular word. This mechanism enables the
model to consider global context and dependencies within the input window.

Self-Attention

Vaswani et al. (2017) introduce the `Scaled Dot-Product Attention` mechanism, il-
lustrated in the left plot of Figure 2.2. Input vectors (input text embedding + po-
sitional embedding) are passed through three trainable matrices, *Q*, *K*, and *V*, to

Scaled Dot-Product Attention　　　　　　Multi-Head Attention
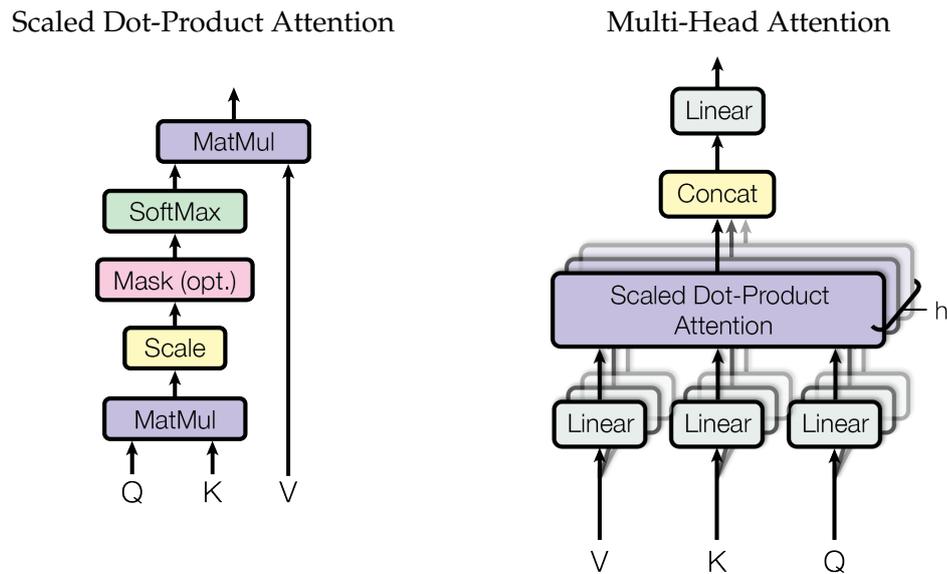
FIGURE 2.2: Illustration of Multi-Head Attention. Input is passed through learnt *query* (Q), *key* (K), and *value* (V) matrices to compute attention scores with respect to other tokens in the sequence. Multi-head attention (on the right) enables the calculation of attention individually and projects the input sequence into different subspaces. The outputs of all the heads are then concatenated and linearly transformed to produce the final output. Source: Vaswani et al. (2017).

compute the Query vector, Key vector, and Value vector. These vectors share the same dimensions as the input vector.

The initial step involves calculating a score that signifies the attention of one token, token_i, in a sequence to another token, token_j. This score is determined by the dot product of the Query vector of token_i and the Key vector of token_j. To prevent the dot products from becoming too large as the dimensionality increases, which can lead to gradients becoming too small during backpropagation, a scaling factor of $\sqrt{d_k}$ is applied to the dot product, with d_k representing the dimension of K, the Key matrix.

After computing the score for each token (including token_i itself), the results are passed through a `Softmax` operation, which normalises the scores to positive numbers and a cumulative sum of 1. The normalised score at each position concerning the query token determines the level of attention that the query token allocates to each position. Finally, the output of the attention layer is computed as the sum of the Value vectors of each position weighted by the normalised scores.

In practice, a set of input vectors is packed into a matrix and simultaneously

multiplied by the *Q*, *K* and *V* matrices. Therefore the attention mechanism can be represented as:

$$Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k})V$$

`Self-attention` denotes applying the attention mechanism to tokens within the same sequence, and is employed in both encoder and decoder layers. The only distinction in the decoder layer lies in the masked attention mechanism, which only permits a token to attend to all the tokens that precede it but not those that follow. This prevents the leakage of information from future tokens during training.

Cross-Attention

In addition to the `self-attention` layers, `cross-attention` is also utilised in the transformer. Here, Query vectors are from the decoder, while both Key and Value vectors are derived from the encoder output. Cross-attention enables each position in the decoder of a sequence-to-sequence model to effectively attend to all positions in the input sequence.

Multi-Head Attention

To enhance the expressive power of self-attention, the transformer uses multiple attention heads, i.e., `Multi-Head Attention`. Instead of applying one attention operation with the entire Query, Key, Value vectors, Vaswani et al. (2017) find that dividing the vectors into multiple chunks with different learned linear projections achieves more efficient use of the model's capacity: e.g., for English-to-German translation, the performance of the model with eight heads is almost 1 BLEU point higher than that of a model of the same size with single-head attention (Vaswani et al., 2017). As illustrated in the right plot of Figure 2.2, each head focuses on different parts of the input sequence, capturing diverse information. Afterwards, the outputs from multiple heads are concatenated and linearly transformed.

2.1.1.2 Other Components in the Transformer

Apart from the multi-head attention mechanism, the following components play important roles in the transformer architecture.

Input Embedding and Position Embedding

The transformer utilises learned embeddings to represent the input. Input sequences are first tokenised into tokens or subwords and then converted into high-dimensional vectors through the input embedding layers. The embedding layers are linear transformation layers at the bottom of both the encoder and the decoder stack (see Figure 2.1).

As the transformer does not inherently capture the sequential order of tokens, positional encoding is incorporated into the input embeddings to convey information about token positions. Vaswani et al. (2017) employ sine and cosine functions of varying frequencies to represent position information, with the positional embeddings typically sharing the same dimensions as the input embeddings. In addition to absolute position embeddings, as seen in Vaswani et al. (2017), various positional embeddings such as relative positional embeddings (Shaw, Uszkoreit, and Vaswani, 2018) and fully learnable embeddings (Devlin et al., 2019; Yinhan Liu et al., 2019; Radford et al., 2019) have also been proposed in the literature.

The two embeddings are often added before being fed to the encoder and decoder stack. Input text embeddings encapsulate the semantic meaning of the input tokens, while positional embeddings capture the sequential order of the tokens in a high-dimensional vector space.

Position-wise Feed-Forward Networks

Each encoder and decoder layer contains a fully connected feed-forward network (FFN) layer, which processes the output from the attention mechanism independently on each position in the sequence. FFN layers typically consist of two linear transformations and a non-linear activation function such as ReLU (Vaswani et al., 2017). FFN layers are crucial in transformers to capture complex, non-linear relationships within the input sequence.

Layer Normalisation and Residual Connections

To enhance training stability, each sub-layer (`multi-head attention` and `FFN` layer) within both the encoder and decoder layers contains layer normalisation and a residual connection. Layer normalisation (Ba, Kiros, and Hinton, 2016) plays a crucial role in mitigating uninformative variations within the hidden vector values. It achieves this by normalising the values to a unit mean and standard deviation within each layer, facilitating more stable and faster training. Residual connection (He et al., 2016a) is beneficial for easier backpropagation and contributes to a smoother loss curve. These components help prevent vanishing or exploding gradients, making it easier to train deep architectures.

Final Output Softmax Layer

In the last output layer of the decoder stack, a learned linear transformation layer is used to convert the output vector into logits. This linear layer at the output shares the same weight matrix with the two input embedding layers of the encoder and the decoder, with the logits representing the scores against each vocabulary item. Subsequently, a `Softmax` function is applied to convert the logits to predict the probabilities of the next tokens. Different decoding strategies can then be employed to sample the tokens to produce the generated sequence.

2.1.2 Transformer Models

In this section, we provide an overview of three fundamental categories of the transformer models: encoder-only models, decoder-only models, and encoder-decoder or sequence-to-sequence models. Additionally, we briefly introduce multimodal transformers, specifically vision-language models.

2.1.2.1 Encoder-Only Models

The first category of the transformer models comprises only the encoder, with BERT (Devlin et al., 2019) standing as one of the most widely adopted models to date.

BERT is trained on the English Wikipedia and Google's BooksCorpus (Y. Zhu et al., 2015), focusing on two primary pre-training objectives: Masked Language Modelling (MLM) and Next Sentence Prediction. Next Sentence Prediction is a binary classification task that determines whether a given pair of consecutive sentences in a text corpus follow each other logically and semantically, while MLM enables bidirectional learning from the text data. MLM is shown to be more potent than either a left-to-right model or a shallow concatenation of left-to-right and right-to-left models (Devlin et al., 2019).

Initially, the training text is tokenised using WordPiece (M. Johnson et al., 2017), encompassing a 30K token vocabulary. To prevent the models from directly seeing future words, Devlin et al. (2019) propose a masking strategy, which randomly masks 15% of the tokens in a sentence and trains BERT to predict the original token. To enhance robustness, the randomly selected tokens to be masked are replaced by [MASK] 80% of the time, by a random token 10% of the time, and left unchanged the remaining 10% of the time. The prediction is optimised using cross-entropy.

MLM has been widely adopted in the training of many models, including popular ones such as RoBERTa (Yinhan Liu et al., 2019), ALBERT (Lan et al., 2020), and ELECTRA (K. Clark et al., 2020), among others.

The tokenised input to the models is first represented as the addition of the token embedding and the positional embedding, then fed to the stack of transformer layers, i.e., the output of the lower layer is treated as the input of the next layer, adopting the same attention mechanism as introduced in Vaswani et al. (2017). BERT is available in two different sizes: a base model with a total of 110M parameters across 12 transformer layers, and a large model with 340M parameters spanning 24 transformer layers.

After pre-training, BERT can then be fine-tuned on downstream tasks. The final representation vector in the last layer is used as the *encoded* representation of the input sentence, which can later be applied to various tasks such as text classification. With BERT setting the new state-of-the-art performance in various downstream NLP tasks, the pre-training and fine-tuning paradigm has proven to be highly effective and remains the predominant method for training language models in the current literature.

2.1.2.2 Decoder-Only Models

The second category of transformer models exclusively features the decoder, where the input data directly enters the decoder without being transformed into a higher, more abstract representation by an encoder. Representative models in this category include GPT (Generative Pre-trained Transformer) from OpenAI, along with its successive iterations like GPT-2 and GPT-3 (Radford et al., 2019; Brown et al., 2020b).

For instance, GPT-2 uses Byte Pair Encoding (BPE) (Sennrich, Haddow, and Birch, 2016) to tokenise the text and is trained on the objective of causal language modelling to predict the subsequent token in a sequence, typically with cross-entropy loss. The decoder in GPT models employs a specific type of attention mechanism known as masked self-attention, which allows a token to attend only to previous tokens, preventing access to future tokens during training.

The left-to-right decoder-only models are effective at generation tasks in an Auto-Regressive (AR) fashion. The process commences by introducing a start token <s> to the model, which traverses through a series of transformer decoder layers, generating token_0 conditioned on <s>. Subsequently, token_0 becomes part of the input to the model, alongside the start token, used for generating the subsequent tokens. This sequence continues until the end-of-sequence token <e> is generated, indicating the completion of the sequence. Fine-tuned GPT models have also demonstrated considerable potential in tasks such as translation, summarisation, and question answering (Radford et al., 2019).

The decoder-only architecture simplifies the model, enhancing its efficiency for specific tasks, such as language modelling. By eliminating the encoder, GPT models can process input data more directly and generate output more rapidly. This design also enables GPT models to be trained on a substantial amount of unlabelled data, a significant advantage in NLP where labelled data is often limited. Thanks to the efficiency and generative capabilities, decoder-only models continue to be developed

with improved size and performance. Most recent LLMs adopt decoder-only transformer architectures, including GPT-3 (175B),[1] OPT (S. Zhang et al., 2022) (125M-175B), PaLM (Chowdhery et al., 2023) (8B-540B), LLaMA (Touvron et al., 2023a) (7B-65B), LLaMA 2 (Touvron et al., 2023b) (7B-70B), to name a few.

2.1.2.3 Sequence-to-Sequence Models

The next category of the transformer model is the Sequence-to-Sequence (Seq2Seq) model, also referred to as the encoder-decoder transformer model, which contains both an encoder and a decoder. This architecture is highly versatile and has demonstrated effectiveness in various NLP applications.

Both the input and output of the Seq2Seq model are sequences. Input sequences first undergo initial processing by the encoder. Subsequently, the high-dimensional representation is passed on to the decoder through the cross-attention mechanism. In each decoding step, the decoder attends not only to its previously generated tokens but also to the most relevant parts of the input sequence. This is achieved through a process similar to self-attention but with an additional step that combines information from both the history of the decoder and the encoded input, enabling the generation of the next token in the output sequence.

Popular models within this category include BART (M. Lewis et al., 2020), T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020a), among others. BART is pre-trained on diverse tasks such as denoising auto-encoding, text infilling, and text generation. This pre-training allows it to excel in various NLP tasks as it learns a broad range of linguistic patterns. T5, on the other hand, is designed with a *text-to-text* framework, unifying diverse NLP tasks into a single system. It showcases flexibility by casting different tasks as text generation problems, allowing a seamless approach to different tasks with a unified mechanism.

Seq2Seq models,are preferred for handling diverse language-related tasks by effectively processing inputs, leveraging cross-attention mechanisms, and generating high-quality output sequences. Compared to decoder-only models, fewer LLMs adopt the encoder-decoder transformers, examples include AlexaTM (Soltan et al., 2022) (20B), Flan-T5 (Chung et al., 2022) (80M-11B).

---

[1]Numbers in the bracket show the model sizes in the number of parameters.

2.1.2.4 Multimodal Transformers

Since the success of the transformer architecture in language processing, it has been adapted to other modalities such as image and audio processing.
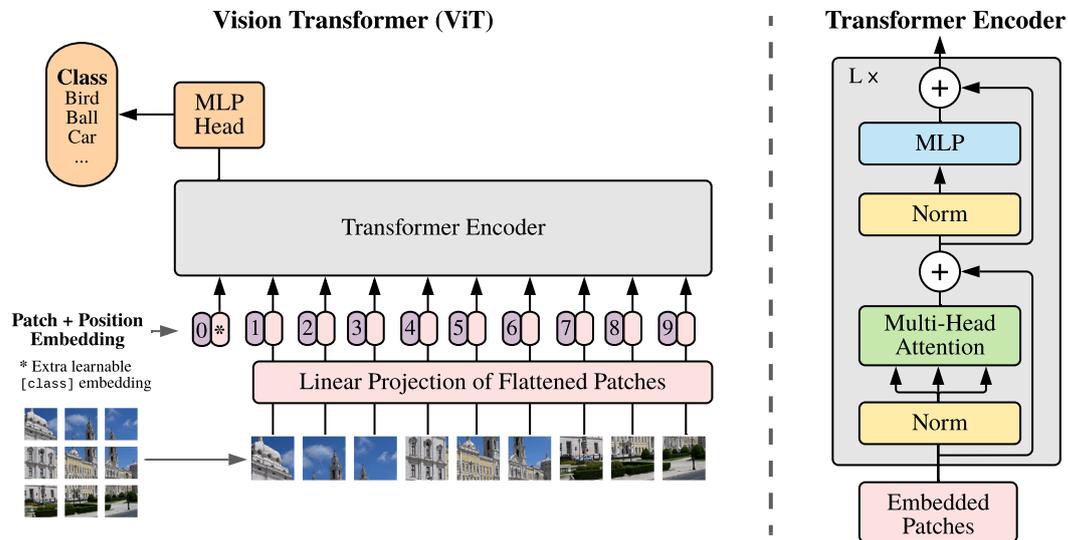
FIGURE 2.3: Vision Transformer. Images are split into fixed-size patches and linearly embedded as sequences. Then position embeddings are added and the resulting vectors are fed to a standard Transformer encoder. Source: Dosovitskiy et al. (2021).

The Vision Transformer (ViT) model is introduced by Dosovitskiy et al. (2021), closely following the design principles of the BERT model. An illustration of ViT is shown in Figure 2.3. Images are split into a sequence of fixed-sized $P \times P$ patches, where P is the patch size. These flattened 2-D image patches are then linearly projected to match the transformer's embedding dimension. ViT is pre-trained using supervised image classification such as ImageNet, which has demonstrated its capability to match or surpass previous stat-of-the-art CNN-based networks like ResNet (He et al., 2016b) on common downstream image classification benchmarks, particularly when ample data is available (Dosovitskiy et al., 2021).

Building on the success of vision transformers, the integration of multiple modalities, such as images and language, has become an active research area. Vision-language transformers, which are typically encoder-decoder transformers, fuse these diverse modalities. In the encoding phase, methods for vision-language models are categorised into two primary architectures: single-stream and dual-stream models.

The single-stream models encode both modalities within the same module. Embedded textual input and image features are either concatenated or combined via cross-modal cross-attention, leveraging Query vectors from one modality while Key and Value vectors come from another. Although different vision-language tasks require different input formats, single-stream architecture is proven effective in models such as VisualBERT (L. H. Li et al., 2019), V-L BERT (W. Su et al., 2020), OSCAR (Xiujun Li et al., 2020), and OFA (P. Wang et al., 2022), etc.

On the other hand, dual-stream models employ two single-modal encoders to independently process vision and text information. They then employ straightforward methods, such as shallow attention layers (K.-H. Lee et al., 2018) or dot products (C. Jia et al., 2021) to project image and text embeddings into the same semantic space. Without the complex cross-attention mechanisms, the dual encoder strategy is often more efficient in modelling vision-language interactions. Models such as ViLBERT (Lu et al., 2019), and LXMERT (H. Tan and Bansal, 2019) fall into this category.

2.2   Knowledge Types and Sources

The precedent sections introduced the advances in transformer-based language models that set the new state-of-the-art performance for a wide variety of NLP tasks, establishing the new paradigm of learning informative contextualised representations for text via training on the large-scale unlabelled corpus. While existing research has showcased the implicit storage of certain knowledge within the latent parameters of language models (Petroni et al., 2019; Roberts, Raffel, and Shazeer, 2020; C. Wang, P. Liu, and Yue Zhang, 2021; B. Cao et al., 2021), PLMs still face challenges in achieving reliable performance in knowledge-intensive tasks such as fact verification (Thorne et al., 2018) and question answering (Kwiatkowski et al., 2019), etc.

Addressing this limitation, the research community has actively explored the integration of explicit knowledge, such as knowledge bases and Wikipedia articles, into large-scale PLMs. In this section, we provide an overview of knowledge types and knowledge sources.

Knowledge comprises a wide spectrum of concepts, e.g., Wikipedia states that

Knowledge is an awareness of facts, a familiarity with individuals and situations, or a practical skill.[2] In the context of NLP, knowledge can be broadly classified into two categories: structured knowledge and unstructured knowledge (C. Zhu et al., 2022).

Structured Knowledge includes Knowledge Bases (KB) or knowledge graphs (KG), dictionaries (W. Yu et al., 2022), syntax-trees (Zhou et al., 2020; Bai et al., 2021), etc. Knowledge Bases are the most common source of structured knowledge utilised in NLP (C. Zhu et al., 2022). Widely-used KBs include Wikidata,[3] DBpedia,[4] WordNet (Miller, 1993), ConceptNet,[5] among others. Both Wikidata and DBpedia are developed around Wikipedia, where structured knowledge is stored as triples, i.e., `<subject, relation, object>`, with subject and object being entities. World knowledge and facts can effectively be represented by such triples, for example, `<United Kingdom, Capital, London>`.
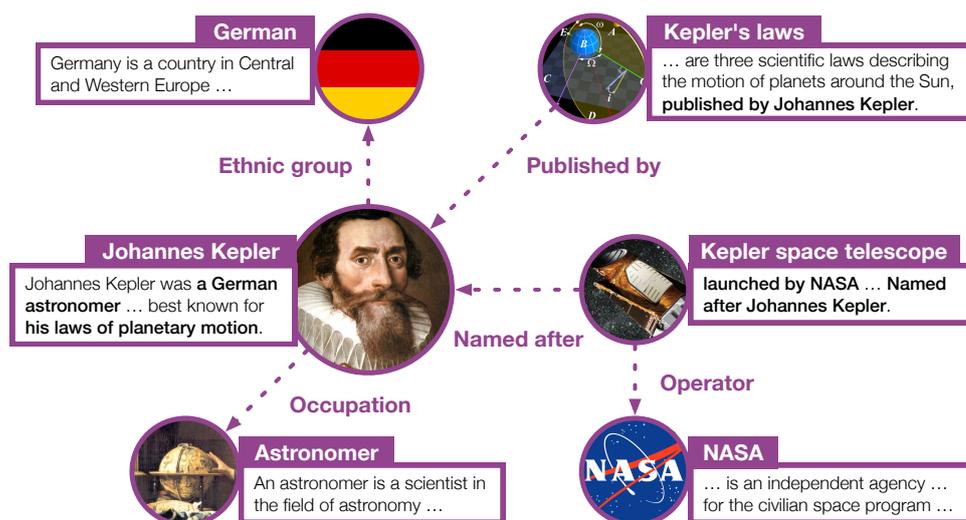
FIGURE 2.4: An example of a knowledge graph in Wikidata, which contains the information of entities, their relations, and the descriptions of the entities. Source: Xiaozhi Wang et al. (2021).

Wikidata is a dynamic and collaborative knowledge base that represents and links entire Wikipedia articles. It provides document-oriented knowledge representation, making it a comprehensive source of diverse information. Wikidata also provides textual descriptions of entities, Figure 2.4 shows an example (source from

---

[2]https://en.wikipedia.org/wiki/Knowledge
[3]https://www.wikidata.org/wiki/Wikidata:Main_Page
[4]https://www.dbpedia.org/
[5]https://conceptnet.io/

Xiaozhi Wang et al. (2021)). DBpedia aims at extracting structured knowledge from Wikipedia and transforming it into a machine-readable format.

WordNet is a lexical database of the English language, which categorises words into synsets, making it particularly useful for tasks such as word sense disambiguation. ConceptNet is a multilingual knowledge base, representing words, phrases, and the commonsense relationships between them. The knowledge in ConceptNet is collected from a variety of resources including crowdsourcing.

Unstructured knowledge takes on various forms, with one prevalent example being text corpora utilised in language model pre-training. Prior research has shown that language models store or memorise knowledge after they are trained, hence can also be treated as an unstructured knowledge source (Petroni et al., 2019; Heinzerling and Inui, 2021).

Neeman et al. (2023) categorise unstructured knowledge into two distinct sources: *parametric knowledge*, which is encoded in or memorised by the model parameters, and *contextual knowledge*, referring to knowledge encapsulated within external textual sources provided to the model at inference time of the downstream tasks, such as paragraphs retrieved based on the question query in question answering.

Knowledge is a theme throughout the thesis. In Chapter 3-5, we primarily consider *Structured Knowledge*, especially the entity and the relations stored in knowledge bases, i.e., WikiData. In Chapters 6 and 7, the exploration of knowledge will be expanded to more flexible forms, including the unstructured *Parametric Knowledge* via self-explanation (Chapter 6), and *Distilled Knowledge* - extraction of knowledge from recent LLMs to power smaller task-specific models via data augmentation (Chapter 7). Particularly in Chapter 4, 5, and 7, we also study zero-shot cross-lingual transfer for multilingual tasks.

2.3   Knowledge-Enhanced Language Models

Now that we have presented the fundamentals of transformer-based language models and knowledge representation, this section continues to introduce the techniques for combining them. Specifically, we categorise knowledge integration methods for pre-trained language models into (i) the incorporation of entity embeddings, (ii) the

utilisation of external memory, (iii) the addition of knowledge-related auxiliary pre-training tasks, (iv) the employment of adapters, and (v) the augmentation of retrieval components into the language models.

2.3.1 Adding Entity Embeddings

The first line of work involves the identification of entities in a sentence, enhancing the original input embeddings with entity knowledge represented in triples by incorporating entity embeddings. Entity embeddings can take the form of pre-trained embedding vectors or share the embedding layer of the transformer model.

Diverse approaches have emerged to representing entities in knowledge bases via entity embeddings. Knowledge graph embedding methods such as TransE (Bordes et al., 2013) and TransR (Y. Lin et al., 2015), employ score functions to measure the distance of two associated entities by their relation. The underlying intuition is that entities linked by a relation are proximal in the vector space, i.e., entity embeddings encapsulate relationship representations from the knowledge graph. Word-entity co-occurrence methods such as Wikipedia2Vec (Yamada et al., 2020a), concurrently learn embeddings of words and entities by identifying co-occurring words around a given entity and placing similar words and entities in close proximity within a continuous embedding space. Another category of methods, like BLINK (L. Wu et al., 2020), utilises entity descriptions encoded by transformer-based PLMs to represent entities.

Models that integrate knowledge by specific entity embeddings include ERNIE (Zhengyan Zhang et al., 2019), KnowBert (Peters et al., 2019), K-BERT (Weijie Liu et al., 2020), etc. These models typically use an existing entity linker or jointly train an entity linker to identify mentions of entities in knowledge bases in the input text, and then retrieve the pre-trained entity embeddings or embed the corresponding entities. Note that the dimensions of the pre-trained entity embeddings are generally much lower than those of the transformer encoders, where a fusion layer is typically added to effectively incorporate entity embeddings from a different embedding space and combine context and entity information.
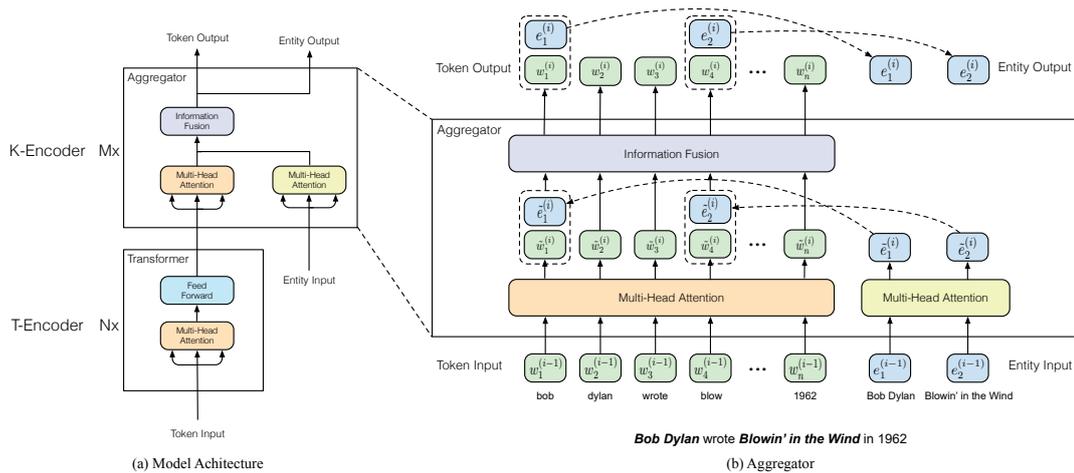
FIGURE 2.5: Illustration of ERNIE. ERNIE comprises a conventional text encoder and an additional knowledge encoder. An aggregator (on the right) is applied for the mutual integration of the input tokens and entities. The information fusion layer takes both token embedding and the concatenation of the token embedding and entity embedding as input, and outputs new token embeddings and entity embeddings for the next layer. Source: Zhengyan Zhang et al. (2019).

ERNIE, for instance, employs TAGME (Ferragina and Scaiella, 2010) to link entities to Wikidata. TAMGE identifies entity mentions in the input text, links them to associated TransE entity embeddings, and fuses them into corresponding positions in the text, as illustrated in Figure 2.5. The knowledge-based learning objective involves predicting correct token-entity alignments, enabling ERNIE to outperform BERT in tasks including entity typing and relation classification (Zhengyan Zhang et al., 2019). However, challenges arise from the need for pre-annotated and linked entities, introducing potential noise through entity linkers like TAGME.

KnowBert, on the other hand, extends BERT by jointly training an entity linker using a knowledge attention and re-contextualisation mechanism, as illustrated in Figure 2.6. It identifies entity spans in the input text and incorporates an integrated entity linker to retrieve entity embeddings from a knowledge base. The entity linker is responsible for entity disambiguation, considering 30 entity candidates and using their weighted average embedding. Knowledge-enhanced entity-span representations are then re-contextualised with a word-to-entity attention technique. When entity-linking supervision is available, the model is trained with an additional
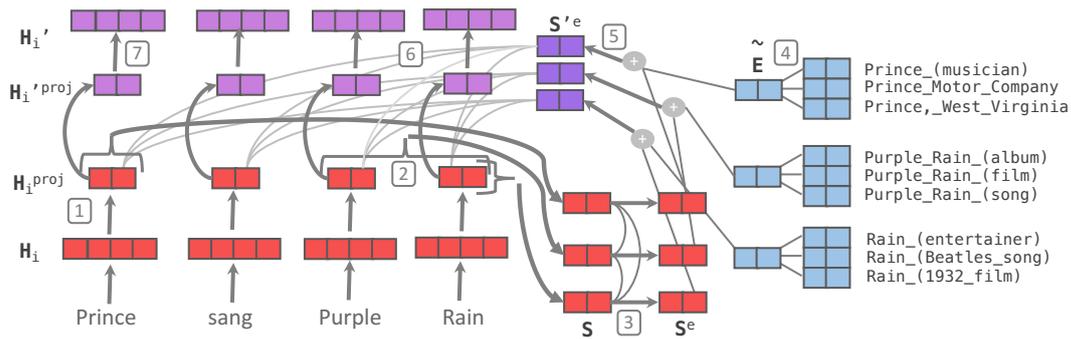
FIGURE 2.6: Illustration the Knowledge Attention and Recontextualisation (KAR) component in KnowBert. BERT word piece representations ($\mathbf{H}_i$) are first projected to $\mathbf{H}_i^{\text{proj}}$ (1), then pooled over candidate mentions spans (2) to compute $\mathbf{S}$, and contextualised into $\mathbf{S}^e$ using mention-span self-attention (3). An integrated entity linker computes weighted average entity embeddings $\tilde{\mathbf{E}}$ (4), which are used to enhance the span representations with knowledge from the KB (5), computing $\mathbf{S}'^e$. Finally, the BERT word piece representations are recontextualised with word-to-entity-span attention (6) and projected back to the BERT dimension (7) resulting in $\mathbf{H}'_i$. Source: Peters et al. (2019).

knowledge-aware log-likelihood or max-margin objective. KnowBert has demonstrated improvements in relationship extraction, entity typing, and word sense disambiguation over BERT and ERNIE (Peters et al., 2019).

K-BERT expands language representation with adaptable knowledge bases via a knowledge layer. This layer detects and injects relevant triples from a knowledge base into the input sentence, converting it into a knowledge-rich sentence tree. The enriched sentence is then processed by the embeddings layer. Soft-position and visible matrix techniques are employed to control the utilisation of knowledge while maintaining the semantic meaning of the sentence close to the original input. K-BERT showcases advantages in knowledge-driven domain-specific tasks, such as question answering and named entity recognition in fields like law, finance, and medicine (Weijie Liu et al., 2020). Moreover, K-BERT also offers flexibility in adapting to different knowledge bases without the need for re-training when changing the KB integrated.

2.3.2 Using External Memory

Another category of approaches involves the use of external memory to seamlessly integrate factual knowledge into language models. This external memory typically

acts as a key-value store, providing access to knowledge base triples or contextual information. This methodology can better facilitate the incorporation and updating of knowledge from knowledge bases into PLMs, often without the need for extensive re-training.

Models that leverage external memory include KGLM (Logan et al., 2019), KNN-LM (Khandelwal et al., 2020), and others. The key idea of KGLM is to condition the language model on a knowledge graph. Unlike most of the models discussed in this thesis, KGLM uses LSTM instead of transformer-based PLMs. In addition to predicting the next word given the previous words in the sequence, KGLM also predicts the next entity given the previous words and entities in the sequence. As KGLM iterates over the sequence, it builds a local knowledge graph which is a subset of the full KG with only entities relevant to the sequence. LSTM is then used to predict the next word as well as its type: related entity, a new entity, or not an entity. KGLM is found to outperform GPT-2 on fact completion tasks. Qualitatively, KGLM tends to predict more specific tokens compared to GPT-2.

KNN-LM learns the similarities between text sequences and stores all representations of text sequences in a nearest neighbour data store. During inference, KNN-LM identifies the k most similar sequences in the data store for a given input, retrieves the corresponding target (e.g., the next word) from these k sequences, and combines the KNN probabilities with the probabilities computed by the language model for the final prediction.

2.3.3 Adding Knowledge-Related Auxiliary Pre-training Tasks

The next set of methods focuses on integrating knowledge by designing auxiliary knowledge-related pre-training tasks, extending beyond traditional training objectives like masked language modelling. These tasks encompass Masked Entity Prediction, Entity Prediction from Descriptions, Entity-Relation Discrimination, etc.

Many models in the literature fall within this category. Since the model architecture is typically not modified, these methods provide flexibility and entail no additional inference overhead during deployment. Below we review some representative models in this category.
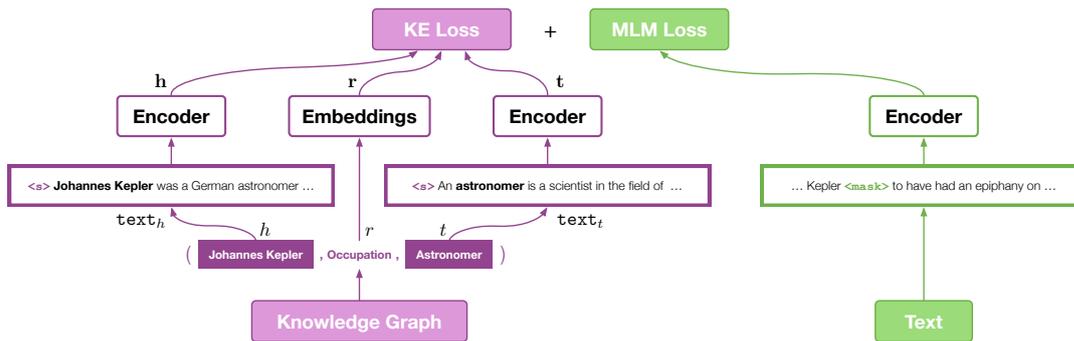
FIGURE 2.7: Illustration of the KEPLER framework. A language model is jointly trained on knowledge embedding (KE) and masked language modelling (MLM) objectives, where three types of entity embeddings are studied for the knowledge embedding task: entity descriptions as embeddings, entity and relation descriptions as embeddings, and entity embeddings conditioned on relations. Source: Xiaozhi Wang et al. (2021).

Xiong et al. (2020) train a model, WKLM, using Wikipedia articles. The mentions in the text are replaced with different entities of the same type from WikiData to create negative knowledge statements that are still linguistically correct. An entity replacement loss is introduced to train WKLM to distinguish between true and false knowledge, together with the MLM objective for pre-training. Sun et al. (2019) adopt a masking strategy at both the phrase and entity levels during training. This introduces tasks that necessitate factual knowledge for model comprehension. Roberts, Raffel, and Shazeer (2020) leverage salient span masking introduced by Guu et al. (2020), to mask out salient spans such as entities and dates. This approach demonstrates enhancements over T5, particularly in question answering.

KEPLER (Xiaozhi Wang et al., 2021), illustrated in Figure 2.7, introduces a knowledge embedding objective with supervision from a knowledge base and optimises jointly with language modelling objectives. KEPLER is specifically trained to encode entities from their contextual descriptions, enhancing the ability of PLMs to extract knowledge from text. Calixto, Raganato, and Pasini (2021) train a multilingual model using Wikipedia articles in 100 languages together with BabelNet (Navigli and Ponzetto, 2012), a multilingual sense-inventory for Word Sense Disambiguation, by predicting Wikipedia hyperlinks.

One of our works from this thesis, EntityCS (Whitehouse, Christopoulou, and Iacobacci, 2022), detailed in Chapter 4, also utilises the entity prediction objectives
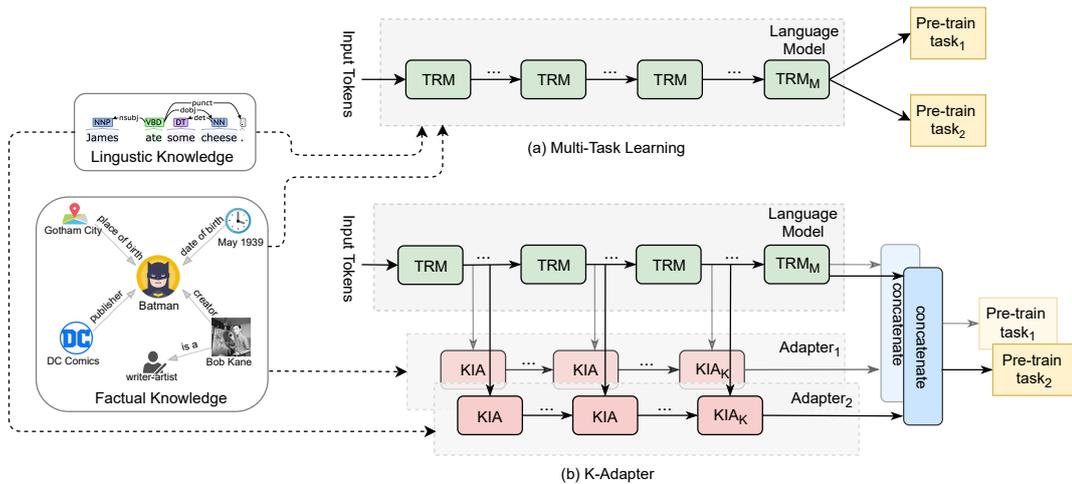
FIGURE 2.8: Illustration of K-ADAPTER. Compared to conventional multi-task learning for injecting knowledge (a) that may result in catastrophic forgetting, K-ADAPTER (b) injects knowledge by training adapters independently on different pre-train tasks, enabling continual knowledge infusion. When injecting new kinds of knowledge, the existing knowledge-specific adapters will not be affected. KIA represents the adapter layer and TRM represents the transformer layer. Source: R. Wang et al. (2021).

on a code-switched training corpus constructed from Wikipedia and Wikidata.

2.3.4 Adding Adapters

Adapters are typically light-weighted or smaller transformer blocks with fewer layers. Keeping the pre-trained language model parameters frozen, the adapter-based models provide efficiency in training and show advantages in mitigating the risk of catastrophic forgetting (T. Vu et al., 2022). In addition to the common use case of adapter for parameter-efficient fine-tuning (Houlsby et al., 2019), researchers have also developed models utilising adapters to integrate knowledge into PLMs. We briefly introduce several such models below.

K-ADAPTER (R. Wang et al., 2021), as illustrated in Figure 2.8, adds learnable adapters to RoBERTa that are trained in a multi-task setting on relation prediction and dependency-tree prediction. Two types of knowledge adapters are developed: factual knowledge obtained from automatically aligned text triples on Wikipedia and Wikidata, and linguistic knowledge obtained via dependency parsing. Both

adapters have demonstrated effectiveness in improving relation classification, entity typing, and question answering (R. Wang et al., 2021). Y. Hou et al. (2022) introduce a lightweight adapter set to enhance multilingual PLMs with cross-lingual entity alignment and facts from multilingual KB for many languages. Their experiments showcase the benefits of incorporating multilingual factual knowledge, particularly for low-resource languages. Many works focus on enhancing tasks requiring domain-specific knowledge with adapters, for instance, Biomedical NLP (Lai, Zhai, and H. Ji, 2023), Task-Oriented Dialogue Systems (Emelin et al., 2022), as well as the exploration of the mixture of domain adapters (Diao et al., 2023).

2.3.5 Retrieval-Augmented Language Models

Retrieval-augmented approaches have gained popularity for expanding contextual knowledge without the need for extensive model parametrisation. It involves training a retriever that dynamically retrieves relevant knowledge, such as passages from Wikipedia or sub-graphs from a KB at runtime. Without the necessity of storing vast amounts of knowledge within the model, retrieval-augmented language models enable more efficient and convenient updates of the evolving knowledge.

Retrieval-augmented models are extensively applied to question-answering and text-generation tasks. M. Joshi et al. (2020) encode questions and passages alongside dynamically retrieved textual encyclopedic background knowledge from multiple documents, particularly focusing on entities mentioned in the text. This method exhibits effectiveness in tasks emphasising factual reasoning, such as reading comprehension.

RAG, Retrieval-Augmented Generation, proposed by P. Lewis et al. (2020), generates answers by retrieving relevant spans across external texts based on pre-trained sequence-to-sequence models. As illustrated in Figure 2.9, given a query, RAG leverages the input sequence to retrieve the top K relevant passages and generates output by conditioning on these latent documents together with the input. REALM, an extension of RAG proposed by Guu et al. (2020), augments the language model by retrieving and attending over documents from a large corpus. It consists of two key components: a neural knowledge retriever implemented with the BERT framework,
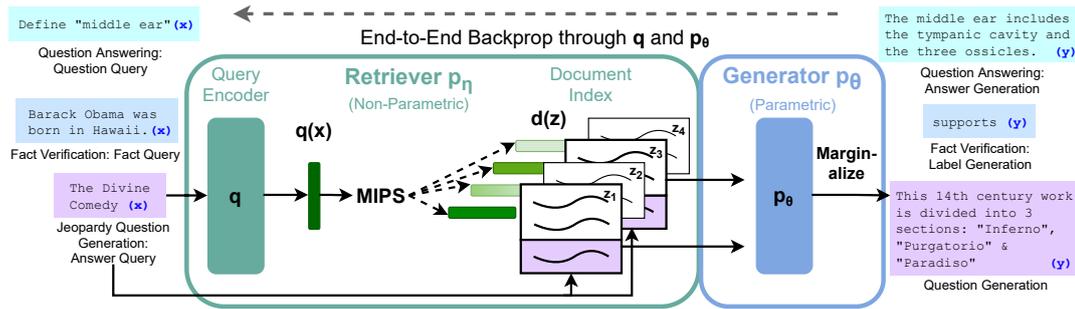
FIGURE 2.9: Illustration of RAG. RAG combines a pre-trained retriever (Query Encoder + Document Index) with a pre-trained Seq2Seq model (Generator) and fine-tune end-to-end. For query x, Maximum Inner Product Search (MIPS) is utilised to find the top-K documents z. For the final prediction y, the retrieved documents are treated as latent variables and marginalised over Seq2Seq predictions given different documents. Source: P. Lewis et al. (2020).

responsible for encoding input data and retrieving potentially helpful documents, and a knowledge-augmented encoder implemented with a Transformer, used to infuse entities in documents and predict words for question-answering tasks.

Retrieval augmentation has also shown advantages in enhancing the few-shot learning capability of language models. Izacard et al. (2023) introduce a retrieval-augmented language model, Atlas, which is designed to handle knowledge-intensive tasks with very few training examples. Atlas demonstrates robust few-shot performance across a diverse range of tasks, including KILT (Petroni et al., 2021), Natural Questions (Kwiatkowski et al., 2019), etc.

Many new retrieval-augmented approaches have been proposed in the light of LLMs (B. Wang et al., 2023; Gao et al., 2023; Asai et al., 2024), among which, self-RAG (Asai et al., 2024) train a LM with reflection tokens. The reflection tokens determine if retrieval would be helpful and criticise its own output to choose the best generation path in terms of factuality and overall quality. Thanks to the effectiveness of extending non-parametric and dynamic knowledge, retrieval-augmented language models maintain their popularity even amid the latest trends of advanced Large Language Models (J. Liu et al., 2023; Patil et al., 2023), alleviating the need for frequent and expensive re-training of ever-larger language models.

2.4 Summary

The background section provides an overview of the transformer architecture, representative transformer-based models, knowledge types and sources, and various knowledge-enhanced language models. The subsequent chapters focus on distinct research aspects, with knowledge as a theme throughout, starting with the next chapter on the application of knowledge-enhanced language models for fake news detection.

Chapter 3

Knowledge-Enhanced Language Models for Fake News Detection

This chapter focuses on evaluating the effectiveness of knowledge-enhanced language models on fake news detection tasks. Specifically, we consider structured knowledge, such as entity knowledge represented in knowledge bases.

The main content is an extended version of the paper "Evaluation of Fake News Detection with Knowledge-Enhanced Language Models" (Whitehouse et al., 2022), published in *the Sixteenth International AAAI Conference on Web and Social Media*.

3.1 Background and Introduction

This chapter studies the fake news detection task, which includes misinformation, disinformation, rumours, hoaxes, and other forms of rapid spread and factually inaccurate information (Sharma et al., 2019a). Due to the wide reach of social media, fake news has been observed to severely impact political processes (Allcott and Gentzkow, 2017). Misinformation related to medical issues, such as the COVID-19 pandemic, can cost lives (O'Connor and Murphy, 2020). There is a growing desire to develop automated methods for fake news detection and mitigation, however, it remains a technically challenging problem (Thorne and Vlachos, 2018).

We focus on content-based fake news detection: methods that assess the truthfulness of news items based only on textual information without using metadata. State-of-the-art models for this task are driven by advances in large-scale pre-trained language models (PLMs) (C. Liu et al., 2019; Kaliyar, Goswami, and Narang, 2021),

which are trained on vast amounts of raw web-based text using self-supervised methods (Rogers, Kovaleva, and Rumshisky, 2020). As discussed in Chapter 2, a major limitation of these models is the lack of explicit grounding to real-world entities and relations, which makes it difficult to recover factual knowledge (Bender et al., 2021). On the other hand, knowledge bases (KBs) provide a rich source of structured and human-curated factual knowledge, often complementary to what is found in raw text. This has recently led to the development of KB-augmented language models (Zhengyan Zhang et al., 2019; Peters et al., 2019). We posit that fake news detection can particularly benefit from the integration of KBs, making such models less dependent and reliant on surface-level linguistic features.

In this chapter, we empirically analyse the impact of recent state-of-the-art knowledge integration methods, which enhance PLMs with KBs, for content-based fake news detection tasks. We evaluate ERNIE (Zhengyan Zhang et al., 2019), KnowBert (Peters et al., 2019), KEPLER (Xiaozhi Wang et al., 2021), and K-ADAPTER, (R. Wang et al., 2021) on two distinct publicly available datasets: LIAR (W. Y. Wang, 2017), a politically oriented dataset, and COVID-19 (Sharma et al., 2019a), a dataset related to the COVID-19 pandemic. We find that integrating knowledge can improve fake news detection accuracy, given that the knowledge bases are relevant and up-to-date. Our experiments are not designed to find new state-of-the-art models for these datasets, but to investigate the effect of knowledge base integration into PLMs.

The main contributions of this chapter are as follows:

- We systematically assess various Knowledge Base integration methods for fake news detection. To the best of our knowledge, this is the first study towards the effectiveness of knowledge-enhanced language models on fake news detection tasks.

- We investigate both model and data aspects that may hinder the effectiveness of KB integration or pose challenges in its accurate measurement.

- We analyse and discuss the potential real-world applications of knowledge-enhanced language models for fake news detection, including dynamic adaptation, adversarial robustness, and the need for human verification in practical deployment scenarios.

In the following sections, we present a brief overview of four state-of-the-art methods that integrate KBs with the PLMs studied. We then introduce and compare the datasets, the experiments with different knowledge-enhanced models, and the effectiveness of entity linking. We discuss our findings with respect to the necessary conditions for KB integration to be effective and how to assess its effect in application scenarios. Finally, we discuss the challenges in fake news detection and promising future directions.

3.2 Related Work

A detailed review of related work on knowledge-enhanced language models is included in section 2.2. This section presents an overview of related work on fake news detection.

3.2.1 Approaches to Fake News Detection

Fake news, including disinformation, misinformation, rumours, hoaxes, etc. (Sharma et al., 2019b), poses significant risks to society. The widespread influence of social media can shape public opinion and manipulate political elections (Allcott and Gentzkow, 2017), and in the case of misinformation related to medical fields and health problems, such as the COVID-19 pandemic, even lead to direct loss of life (O'Connor and Murphy, 2020).

Automated and accurate fake news detection and mitigation represent critical yet technically challenging problems (Sharma et al., 2019a; J. Su, Cardie, and Nakov, 2023). Over the past decade, various fake news detection methods using deep learning techniques have emerged. These methods can be categorised into three types: content-based approaches, user behaviour or propagation pattern analysis, and hybrid models combining both.

Content-based Methods

Content-based methods focus on the textual statements of fake news (Oshikawa, Qian, and W. Y. Wang, 2020) and analyse the language features of the content. For instance, fake news articles and posts often contain more negative and exaggerated

words, which can be leveraged to assess the truthfulness of the news (Rubin et al., 2016).

The majority of current content-based methods employ neural networks for text analysis. For example, Convolutional Neural Networks, widely used in text classification tasks (Oshikawa, Qian, and W. Y. Wang, 2020), have demonstrated superior results on the LIAR dataset compared to traditional neural networks (W. Y. Wang, 2017). In recent years, larger-scale pre-trained transformer-based models like BERT have significantly advanced the state-of-the-art for many NLP tasks, including content-based fake news detection (Kaliyar, Goswami, and Narang, 2021; Farokhian, Rafe, and Veisi, 2023; J. Su, Cardie, and Nakov, 2023). Most fake news detection approaches either combine text with metadata (Ding, Y. Hu, and H. Chang, 2020) or focus solely on the source of the text (Nørregaard, B. D. Horne, and Adali, 2019).

In terms of the two specific datasets we study, for LIAR, Alhindi, Petridis, and Muresan (2018) extend the data with evidence sentences in a new dataset LIAR-PLUS to enhance detection. Chernyavskiy and Ilvovsky (2020) introduce a Deep Averaging Network to model the discursive structure of the text and use Siamese models on the extended text data. Additionally, C. Liu et al. (2019) predict labels at two levels of granularity. In the context of the COVID-19 dataset, results from the CONSTRAINTS 2021 workshop (Chakraborty et al., 2022) showcase a variety of traditional and neural NLP models. Notably, none of these approaches incorporates external knowledge, suggesting potential benefits from knowledge base integration.

User Behaviour-based Methods

In contrast to content-based fake news detection, user behaviour-based methods contend that algorithms focusing on clues from news content are generally less effective. This is because fake news is often intentionally crafted to mislead users by mimicking true news (Shu, S. Wang, and Huan Liu, 2019). Instead, these methods study the social context during the news dissemination process on social media, including user profiles and user behaviours such as likes or retweets.

Shu, S. Wang, and Huan Liu (2019) propose a tri-relationship embedding framework called TriFN, which simultaneously models publisher-news relations and user-news interactions for fake news detection. Another approach by Monti et al. (2019)

involves an automatic fake news detection model based on geometric deep learning, offering language independence and improved resilience to adversarial attacks.

Hybrid Methods

The third category of fake news detection methods involves hybrid models that combine content and user behaviour analysis during fake news propagation.

An example of a hybrid deep model is the CSI model (Ruchansky, Seo, and Yan Liu, 2017). CSI comprises three modules: Capture, Score, and Integrate, correlating the characteristics of the news text, user response, and publisher behaviour. The Capture module captures temporal patterns of user responses to a specific news article and extracts latent features of the text using RNN. The Score module learns source characteristics based on user behaviour, scoring publishers based on user conduct. In the final Integrate module, the model combines response, text, and source information to classify each news item as fake or real.

3.3 Models and Datasets

After presenting the background and related work regarding fake news detection, we now introduce the models that we use in our experiments and provide details of the datasets and our experimental setup.

3.3.1 Knowledge-Enhanced PLMs

In Chapter 2, we have comprehensively reviewed various methods for integrating knowledge into PLMs, and in this chapter, we specifically focus on the following four models:

ERNIE enhances BERT (Devlin et al., 2019) by introducing knowledge through pre-training on both extensive corpora and KBs. While retaining the text encoder of BERT, ERNIE adds an additional knowledge encoder. The knowledge encoder follows the standard transformer architecture, with each layer applying multi-head attention over entity embeddings and token embeddings. A fusion layer then combines the output of the attention heads. ERNIE uses TAGME (Ferragina and Scaiella,

2010) to link entities to Wikidata. TAMGE identifies entity mentions in the input text and links them to associated TransE (Bordes et al., 2013) entity embeddings, which are then fused into the corresponding positions of the text. Apart from Masked Language Modelling and Next Sentence Prediction pre-training tasks as in BERT, ERNIE also adopts a knowledge-based learning objective which predicts the correct token-entity alignment. ERNIE demonstrate superior performance over BERT in entity typing and relation classification (Zhengyan Zhang et al., 2019).

KnowBert   incorporates KBs into BERT using knowledge attention and contextualisation mechanism. It identifies entity spans in the input text and incorporates an integrated entity linker in the model to retrieve entity embeddings from a KB. The entity linker is responsible for entity disambiguation, which considers 30 entity candidates and uses their weighted average embedding. Knowledge-enhanced entity-span representations are then re-contextualised with a word-to-entity attention technique. When entity-linking supervision is available, the model is trained with an additional knowledge-aware log-likelihood or max-margin objective. KnowBert has shown improvement over BERT in relationship extraction, entity typing and word sense disambiguation (Peters et al., 2019).

KEPLER   integrates factual knowledge into PLMs by adding a knowledge embedding objective with the supervision from a KB and optimising it jointly with language modelling objectives. KEPLER is trained to encode the entities from their contextual descriptions, which enhances the ability of PLMs to extract knowledge from text. By keeping the original structures of PLMs, KEPLER can be used in general downstream NLP tasks without additional inference overhead. Specifically, after training on Wikidata5M,[1] a large KB-aligned dataset with entity descriptions, KEPLER shows improved performance over RoBERTa (Yinhan Liu et al., 2019) in relationship extraction, entity typing and link prediction (Xiaozhi Wang et al., 2021).

K-ADAPTER   retains the PLMs unchanged, but adds learnable adapter features that are trained in a multi-task setting on relation prediction and dependency-tree prediction. Two kinds of knowledge adapters have been developed by R. Wang

---

[1]https://deepgraphlearning.github.io/project/wikidata5m

et al. (2021): factual knowledge obtained from automatically aligned text triples on Wikipedia and Wikidata, and linguistic knowledge obtained via dependency parsing. Both adapters have demonstrated effectiveness in improving relation classification, entity typing, and question answering (R. Wang et al., 2021).

3.3.2 Fake News Datasets

In our experiments, we use `LIAR` and `COVID-19` to study fake news detection. Both datasets consist of short statements, however, they differ in content, collection timelines, and linguistic features.

LIAR was collected in 2017 from Politifact.[2] It includes 12.8k human-labelled short statements about US politics from various contexts, i.e., news releases, TV interviews, campaign speeches, etc. Each statement has been rated for truthfulness by a Politifact editor using a six-grade scale: *pants-fire*, *false*, *barely-true*, *half-true*, *mostly true*, and *true*. `LIAR` also provides metadata (e.g., speaker, context), which we do not use in our experiments. While W. Y. Wang (2017) has been widely cited, we only found three other results for our specific task (no metadata, six classes) (Alhindi, Petridis, and Muresan, 2018; C. Liu et al., 2019; Chernyavskiy and Ilvovsky, 2020), the latter has the best accuracy of 34.5%.

COVID-19 was collected in 2020 after the COVID-19 outbreak. It consists of 10.5k posts related to the pandemic which are obtained from different social media sites including Twitter, Facebook, and Instagram. The fake posts were collected from various fact-checking websites, i.e., Politifact and NewsChecker,[3] and the real posts were from Twitter using verified Twitter handles. Each post has a label, *real* or *fake*. It was used as a shared task in the CONSTRAINT 2021 workshop (Chakraborty et al., 2022) with the best-reported accuracy of 98.69%.

Linguistic Feature Analysis

We perform a linguistic feature analysis following the work in B. Horne and Adali (2017) to investigate the stylistic difference between real and fake news in the datasets.

---

[2]https://www.politifact.com
[3]https://newschecker.in/

(A) Word count per statement

(B) POS, punctuation, numbers in `LIAR`
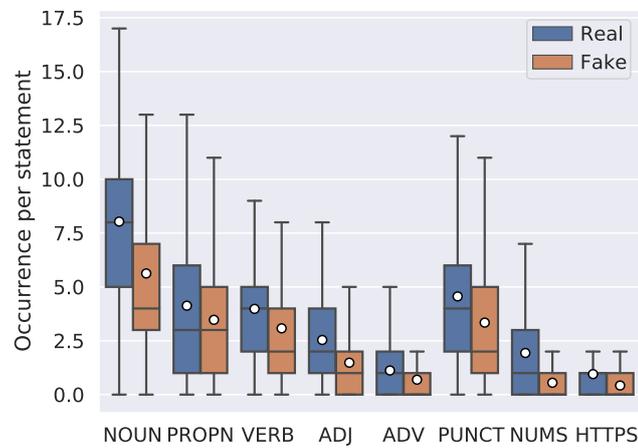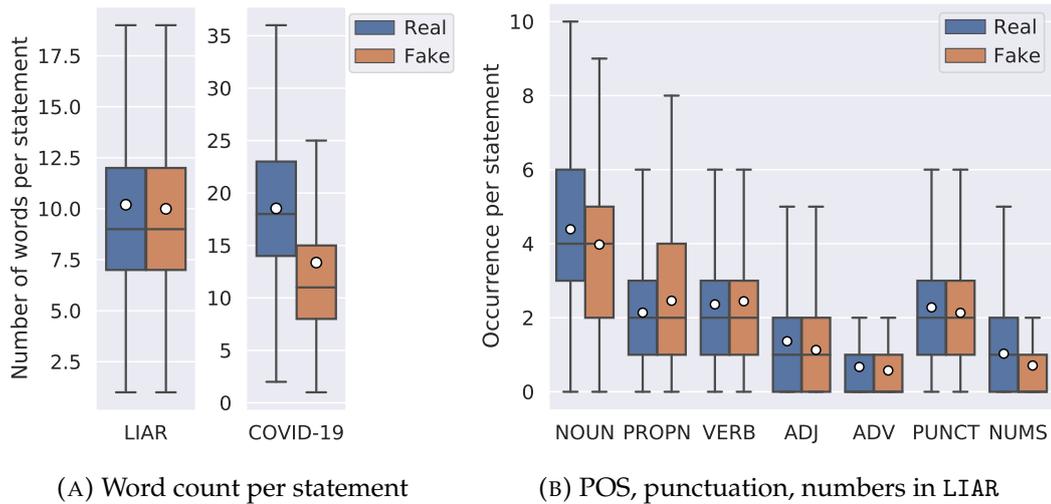
(C) POS, punctuation, numbers, https in `COVID-19`

FIGURE 3.1: Number of words, POS tags, punctuation, and number-like words per statement in `LIAR` and `COVID-19`, as well as number of https-links per statement in `COVID-19`. The mean values are shown as white-filled circles in the plots.

We use SpaCy[4] to parse the statements and to obtain the Part-of-Speech (POS) tags. For `LIAR`, we group *pants-fire*, *false*, and *barely-true* as fake and *half-true*, *mostly true*, and *true* as real. We compare the distribution of various features including word count, POS tags (NOUN, PROPN, VERB, ADJ, ADV), punctuation, and number-like words, in each statement. The results are presented in Figure 3.1.

In terms of statement length, `COVID-19` exhibits notable disparities between real and fake classes, with averages of 32 and 22 words, respectively, as illustrated in Figure 3.1a. Conversely, `LIAR` demonstrates a more balanced distribution of statement lengths, with both real and fake statements comprising an average of 18 and

---

[4]https://spacy.io/

17 words, respectively.

In general, COVID-19 presents distinct linguistic features between classes whereas LIAR shows more comparable features. Notably, COVID-19 incorporates links, predominantly in the form of https links, forming a separate category with a markedly skewed distribution, as depicted in Figure 3.1c.

3.4 Experimental Setup

To assess the influence of external knowledge, we compare the performance of each knowledge-enhanced PLM with the corresponding baseline model. It is noteworthy that ERNIE and KnowBert incorporate entity embeddings linked to the input, allowing us to visualise the entities contributing to fake news detection in ERNIE and design experiments to probe the impact of entity disambiguation in KnowBert.

For the evaluation of model performance in fake news detection, we fine-tune the knowledge-enhanced PLMs on the training set, employing consistent hyperparameter settings. The input text undergoes processing by the PLM, followed by a dropout ($p = 0.1$) and a linear layer. The output is then directed to a softmax layer for classification. The optimisation is carried out using the AdamW optimiser (Loshchilov and Hutter, 2019) with a learning rate of $5e^{-6}$, and cross-entropy serves as the loss function. We set the maximum input length to 128, and the batch size is fixed at 4. Training is conducted for 10 epochs, with convergence typically observed after five epochs. Each experiment is run five times, and the average accuracy, along with the standard deviation, is reported.

As for the models we use, ERNIE and KnowBert are built on BERT-base, whereas KEPLER and K-ADAPTER are enhanced from RoBERTa-base and RoBERTa-large, respectively. In terms of hardware, a Nvidia GTX 1080 GPU with 12GB of VRAM is utilised for experiments involving BERT-base, RoBERTa-base, and models based on them. To maintain a consistent batch size for RoBERTa-large and K-ADAPTER models, a T4 with 16GB VRAM is employed.

3.5   Results and Discussion

We compare ERNIE, three pre-trained KnowBert models with different KBs (Wiki, WordNet, W+W),[5] KEPLER, and K-ADAPTER with three adapters (F, L, F-L)[6] in the published implementation, to the corresponding baselines. This section presents the main results and discussion.

3.5.1   Detection Accuracy

MODEL	BASE	LIAR	COVID-19
BERT-**B**ase (BB)	-	$26.36_{\pm 0.58}$	$97.51_{\pm 0.19}$
RoBERTa-**B**ase (RB)	-	$26.71_{\pm 0.93}$	$97.61_{\pm 0.26}$
RoBERTa-**L**arge (RL)	-	$\mathbf{27.36}_{\pm 0.79}$	$\mathbf{97.92}_{\pm 0.17}$
ERNIE	BB	$27.53_{\pm 0.13}$	$97.30_{\pm 0.18}$
KnowBert-Wiki	BB	$27.64_{\pm 0.09}$	$97.37_{\pm 0.09}$
KEPLER	RB	$26.77_{\pm 1.15}$	$97.58_{\pm 0.15}$
K-ADAPTER-F	RL	$\mathbf{28.63}_{\pm 0.90}{}^{*}$	$\mathbf{97.92}_{\pm 0.10}$
KnowBert-WordNet	BB	$26.95_{\pm 0.45}$	$97.00_{\pm 0.06}$
KnowBert-W+W	BB	$\mathbf{28.95}_{\pm 0.64}{}^{*}$	$97.56_{\pm 0.15}$
K-ADAPTER-L	RL	$28.46_{\pm 0.87}{}^{*}$	$98.07_{\pm 0.09}$
K-ADAPTER-F-L	RL	$27.45_{\pm 0.78}$	$\mathbf{98.11}_{\pm 0.14}$

TABLE 3.1: Detection accuracy results (average of five runs). The first section corresponds to the baseline models. Models in the second section use Wikidata KB. The third section shows models using other KBs and features. The best values within each section per dataset are marked in bold. The subscript numbers with $\pm$ show the standard deviation. Results with * indicate statistically significant improvements over the baseline, both for the mean (t-test, one-sided, $p < .05$) and median (Wilcoxon signed rank test, one-sided, $p < .05$).

The detection accuracy of the knowledge-enhanced PLMs and their corresponding baselines is detailed in Table 3.1.

To disentangle the effectiveness of knowledge integration methods, specific knowledge resources, and baseline model architectures, we categorise the results into three groups: baseline performance (top), models integrated with Wikidata knowledge base (middle), and models utilizing knowledge bases beyond Wikidata (bottom). Comparing models with Wikidata knowledge and their baselines, we observe consistent enhancements on the LIAR dataset. However, no improvement is observed on

---

[5]They refer to Wikipedia, WordNet, and Wikipedia+WordNet as the knowledge base.
[6]They refer to Factual, Linguistic, and Factual+Linguistic adapters.

the `COVID-19` dataset. This discrepancy underscores the positive impact of relevant knowledge bases on transformer-based models, irrespective of the diverse integration approaches employed. Conversely, in scenarios where knowledge bases are outdated and datasets exhibit stylistic imbalances between classes (as outlined in section 3.3.2), external knowledge bases may even detrimentally affect performance.

Comparing models employing the same knowledge integration approach but utilising different knowledge sources reveals intriguing findings. Optimal performance on both `LIAR` and `COVID-19` datasets is achieved by models incorporating multiple knowledge sources. Specifically, KnowBERT-W+W, incorporating both Wikidata and WordNet knowledge bases, exhibits the most substantial overall improvement (a notable increase of +2.59 over BERT-base). Simultaneously, K-ADAPTER, incorporating linguistic adapters, demonstrates the most significant positive impact on the `COVID-19` dataset. This aligns with the observation that `COVID-19` exhibits distinct linguistic stylistic cues between different classes.

Overall, across the `LIAR` dataset, all knowledge-enhanced methods demonstrate improvements over the baseline. In contrast, on the `COVID-19` dataset, only three out of eight models show improvement, and these improvements are marginal

The computational cost varies across different approaches. KEPLER maintains the baseline PLM architecture, incurring no additional overhead. For K-ADAPTER, the RoBERTa-large layers are frozen, resulting in a manageable overhead ranging from 9-23% due to the adapters. However, for KnowBert, the overhead is more substantial, ranging from 40-87%, and for ERNIE, it is even higher at 111-131%.

3.5.2 KB Linking

ERNIE and KnowBert establish links between the text and KB entities at runtime, and the quality of this linking significantly impacts the output. ERNIE utilises TAGME and selects only one entity candidate per text span. In Figure 3.2, we present the 50 most frequently selected KB entities for each dataset.

In the case of `COVID-19`, the most frequently selected entities do not appear content-related, such as "https" and "twitter", while the highly relevant term "COVID-19" is notably absent in the linked entities. Conversely, for `LIAR`, the linked entities seem

(A) Linked Entities in COVID-19

(B) Linked Entities in LIAR

FIGURE 3.2: Word cloud for the 50 most frequent entities linked by ERNIE in `LIAR` and `COVID-19`.

more relevant. This discrepancy may be attributed to the fact that `LIAR` was collected three years earlier, making it potentially better suited for the entity linker and the KB used. Another potential factor influencing the effectiveness of KB integration is the number of linked entities. Unlike ERNIE, KnowBert selects the 30 most probable entities per text span. In a sensitivity study, we restrict KnowBert-W+W to only one entity, leading to a reduction in accuracy on `LIAR` from 28.95% to 27.31%, placing it below the accuracy of ERNIE (27.53%).

3.5.3   Discussion

The consistent improvement in detection accuracy on `LIAR` achieved by integrating PLMs with the Wikidata KB demonstrates the potential of knowledge integration, surpassing results obtained by W. Y. Wang (2017) which integrated multiple types of metadata. However, the improvements, while notable, are not dramatic for `LIAR`, and they lack consistency for `COVID-19`. Two critical aspects contributing to these results are identified in the effective use of knowledge-enhanced models:

- Currentness and Relevance of the Knowledge Base: Since `COVID-19` was collected after most PLMs were trained, certain terms like "COVID-19" may not be present in the Knowledge Base.

- Quality of the Dataset: The `COVID-19` dataset contains confounding factors that provide strong cues, potentially overshadowing the impact of the knowledge base. Notably, the prevalence of https links, occurring in 95.3% of real posts but only 42.3% of fake posts, can act as a *shortcut* to deriving the correct prediction.

There is also potential to achieve more explainability and interpretability with direct KB integration at runtime. For instance, in the statement from `COVID-19`: *"DNA Vaccine: injecting genetic material into the host so that host cells create proteins that are similar to those in the virus against which the host then creates antibodies"*, KnowBert-W+W correctly classifies it as "real", whereas BERT-base fails. We observe that most mention spans in the statement, i.e., *"DNA"*, *"injecting"*, *"genetic"*, *"genetic material"*, *"host"*, *"cells"*, etc. are correctly linked to entities *"DNA"*, *"Injection_(medicine)"*, *"Genetics"*, *"Genome"*, *"Host_(biology)"*, *"Cell_(biology)"*, respectively, suggesting that entity links may have contributed to KnowBert-W+W for this classification. However, the level of explainability remains limited. The entity linking in both models is generally of mixed quality as well, as illustrated in the `COVID-19` example.

Application Aspects

The application of automatic fake news detection in real-world scenarios introduces two dynamic aspects that are challenging to test with static datasets, as highlighted by our experiment on `COVID-19`:

- Dynamic Adaptation: The system needs to adapt to the changing characteristics of real and fake news (Silva and Almeida, 2021). Knowledge-enhanced models that utilise Knowledge Bases at runtime provide an opportunity to update the KB independently of the model. This approach offers the advantage of recognising fake news as contradicting the KB even before specific examples of fake news emerge.

- Adversarial Robustness: Authors of fake news are likely to employ evasive strategies. Adapting the text style is relatively straightforward and can be automated, making the detection using stylistic features challenging (see Zellers et al., 2019b; Schuster et al., 2020).

The deployment of fake news detection in social media is likely to necessitate human verification, especially when users challenge actions taken against them. In this context, Knowledge Base integration can provide a valuable advantage by offering insights into the knowledge used in the detection process, thereby enhancing explainability. This transparency could be crucial in addressing user concerns and building trust in the fake news detection system.

3.6   Conclusion and Future Work

In this chapter, we study the effectiveness of enhancing PLMs with knowledge bases for fake news detection. The findings underscore that the success of integrating knowledge with PLMs is contingent upon the availability of suitable KBs and the quality of the dataset. While better performance is observed on a *static* dataset, there is room for improvement on both the modelling and application levels. To the best of our knowledge, our work is the first examination of knowledge-enhanced models in the context of fake news detection. The positive results on the LIAR dataset provide insights into the effectiveness of considering entity knowledge in areas beyond traditional tasks like entity linking.

For practical application, more insight into the specific knowledge utilised during the detection process could contribute to more transparent and interpretable models. Furthermore, the potential for dynamic adaptation of models and KBs to the evolving characteristics of real and fake news is a promising avenue for exploration. The integration of KBs with PLMs presents an opportunity for more robust and timely fake news detection.

Future work could also investigate the development of a more reliable evaluation approach, for example involving testing scenarios that simulate dynamic knowledge

updates as well as the challenges posed by adversarial and automatic fake news generators. Overall, as the landscape of misinformation evolves, the adaptability and resilience of knowledge-enhanced models demonstrate promise in their effectiveness in real-world applications.

3.7 Limitations

We first discuss the general limitations of incorporating external knowledge resources for improving Transformer-based models, and then we describe the specific limitations of this chapter.

General Limitations of External KBs for Improving Transformer Models

- **Scalability and Efficiency**: Transformer models are already computationally intensive, especially for long contexts, due to the quadratic complexity of the attention calculation within the window length. Integrating external knowledge sources can further increase computational costs, especially when developing adaptive approaches that require up-to-date knowledge on demand.

- **Domain Adaptation and Generalisation**: External knowledge resources often originate from specific domains or sources, which may not fully align with the target task or dataset. This misalignment poses a challenge, particularly as many knowledge integration approaches require linkage to predefined knowledge sources. As large language models evolve toward greater general-purpose utility, there arises a growing need for effective general-purpose knowledge fusion approaches.

- **Selective Knowledge Incorporation**: Existing strategies for leveraging external knowledge in Transformer models often adopt a binary approach, either consistently integrating external knowledge or entirely disregarding it. However, as LLMs evolve and accumulate knowledge within their parameters, the benefit of external knowledge integration becomes context-dependent. There

are instances where querying external knowledge may hinder model performance, given the limitations above (i.e., the computational overhead and potential mismatches between the external knowledge and the task at hand). A recent work, Self-RAG (Asai et al., 2024), which trains the LM to reflect the necessity to retrieve external knowledge and critique its own generation conditioned on the external knowledge, shines a light into this direction.

Limitations of this Chapter

- **Limited Datasets**: The experiments were conducted only on two fake news datasets, both consisting of short statements. Notably, the `COVID-19` dataset already exhibited a very high baseline accuracy. To gain more comprehensive insights into knowledge-enhanced language models, it would be beneficial to include more diverse and complex fake news detection datasets.

- **Limited Knowledge Sources**: The evaluation in this chapter focuses on pretrained models that integrate with knowledge bases primarily sourced from Wikidata and WordNet. However, for the specific use case of COVID-19, there is a potential value in conducting a comparison that involves integrating medical domain-specific knowledge bases.

Chapter 4

Entity-Centric Code-Switching for Enhanced Cross-lingual Transfer

In this chapter, we focus on utilising structured knowledge in a multilingual setup, for the improved cross-lingual transferability of pre-trained cross-lingual language models on entity-centric tasks.

The main content of this chapter is based on the paper "ENTITYCS: Improving Zero-Shot Cross-lingual Transfer with Entity-Centric Code Switching" (Whitehouse, Christopoulou, and Iacobacci, 2022) published in *Findings of the Association for Computational Linguistics: EMNLP 2022.*

4.1   Background and Introduction

Cross-lingual pre-trained Language Models (XLMs), such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020a), have achieved state-of-the-art zero-shot cross-lingual transferability across diverse Natural Language Understanding (NLU) tasks. These models have been notably enhanced through the incorporation of bilingual parallel sentences, along with alignment methods (Yang et al., 2020; Chi et al., 2021a; J. Hu et al., 2021; Gritta and Iacobacci, 2021; Feng et al., 2022). However, acquiring high-quality parallel data is costly, especially for low-resource languages. Therefore, alternative data augmentation approaches have been proposed, one of which is Code Switching (CS).

Code Switching is a phenomenon in which multilingual speakers alternate between languages when they speak, a topic that has been studied for many years

(Gumperz, 1977; Khanuja et al., 2020; Doğruöz et al., 2021). Code-switched sentences consist of words or phrases in different languages, capturing finer-grained cross-lingual expressions compared to parallel sentences. They have been utilised for multilingual intermediate training (Yang et al., 2020) and fine-tuning (Qin et al., 2020; Krishnan et al., 2021). Nevertheless, manually creating large-scale CS datasets is expensive, and only a few natural CS texts exist (Lyu et al., 2015; Barik, Mahendra, and Adriani, 2019; R. Xiang et al., 2020; Chakravarthi et al., 2020; Lovenia et al., 2022). As a result, research has turned to automatic CS data generation.

Some of these approaches generate CS data via dictionaries, often ignoring ambiguity (Qin et al., 2020; Conneau et al., 2020b). Others require parallel data and an alignment method to match words or phrases between languages (Yang et al., 2020; Rizvi et al., 2021). In both cases, what is switched is chosen randomly, potentially resulting in syntactically odd sentences or switching to words with little semantic content (e.g., conjunctions). This is in contrast to observations from prior work that have shown in real code-switched data, such as SEAME (Lyu et al., 2015), where nouns have the highest rate of code-switching (Çetinoğlu, Schulz, and N. T. Vu, 2016). They also find that people may switch grammar rules when they code-switch, making automatic code-switching by randomly replacing words in a sentence less feasible.

On the other hand, entities contain external knowledge and do not alter sentence syntax when replaced with other entities, mitigating the need for parallel data or word alignment tools. Motivated by this, we propose ENTITYCS, a code-switching method that focuses on entities. Resources such as Wikipedia and Wikidata offer rich cross-lingual entity-level information and have shown benefits in XLMs pre-training (Z. Jiang et al., 2020; Calixto, Raganato, and Pasini, 2021; X. Jiang et al., 2022). We use such resources to generate an entity-based CS corpus for the intermediate training of XLMs. Entities in wikilinks[1] are switched to their counterparts in other languages retrieved from the Wikidata KB, thus alleviating ambiguity.

Training models on our synthetic entity-level code-switched data offers several

---

[1] https://en.wikipedia.org/wiki/Help:Link#Wikilinks_(internal_links)

advantages over using naturally occurring code-switched text. Firstly, our synthesised data allows for greater control over language diversity. While natural code-switched text often involves only one or two high-resource non-English languages, our dataset can incorporate a broader range of languages and combinations. Additionally, by focusing on entity-centric training objectives, the models are anticipated to capture finer-grained semantics shares across languages, which is particularly beneficial for downstream tasks that require accurate entity understanding. However, it is important to note that synthetic data may not fully capture the complexities of naturally occurring code-switched text, which often involves subtle shifts in language use influenced by cultural, social, and contextual factors (Doğruöz et al., 2021).

Using the ENTITYCS corpus, we propose a series of masking strategies that focus on enhancing Entity Prediction (EP) for better cross-lingual entity representations. We evaluate the models on entity-centric downstream tasks, including Named Entity Recognition (NER), Fact Retrieval, Slot Filling (SF), and Word Sense Disambiguation (WSD). Extensive experiments demonstrate that our models outperform the baseline on zero-shot cross-lingual transfer, with a +2.8% improvement on NER, surpassing the prior best result that uses large amounts of parallel data, +10.0% on Fact Retrieval, +2.4% on Slot Filling, and +1.3% on WSD.

The main contributions of this chapter include:

- Construction of an entity-level CS corpus, ENTITYCS, based on the English Wikipedia and Wikidata, mitigating the need for parallel data, word-alignment methods, or dictionaries.

- A series of intermediate training objectives, focusing on Entity Prediction.

- Improvement of zero-shot performance on NER, Fact Retrieval, Slot Filling, and WSD.

- Further analysis of model errors, the behaviour of different masking strategies throughout training, as well as the impact across languages, and demonstration of the particular benefit of non-Latin script languages.

4.2   Related Work

We introduce the related work of this chapter in the following three aspects:

4.2.1   Cross-Lingual Pre-Training

Most existing cross-lingual pre-trained language models employ parallel data to enhance multilingual contextualised word representations for different languages (X. Ouyang et al., 2021; Luo et al., 2021; Chi et al., 2021b). Adapters have also been applied to improve zero-shot and few-shot cross-lingual transfer by training only a small set of model parameters (Pfeiffer et al., 2020; Ansell et al., 2021). Additionally, meta-learning techniques (Nooralahzadeh et al., 2020; Tarunesh et al., 2021) have proven highly effective for rapid adaptation to new languages (Dou, K. Yu, and Anastasopoulos, 2019). In comparison to these approaches, our method aims to enhance cross-lingual transferability through intermediate training on an entity-based code-switching corpus created from Wikipedia wikilinks, without requiring parallel data.

4.2.2   Code Switching

Code Switching methods have shown success in cross-lingual model pre-training and fine-tuning across various NLU tasks, including NER (Priyadharshini et al., 2020; L. Liu et al., 2021), Part-of-Speech Tagging (Ball and Garrette, 2018), Machine Translation (Srivastava and Singh, 2020), Intent Classification, and Slot Filling (Krishnan et al., 2021). These methods have also been applied on code-switched datasets (Rizal and Stymne, 2020; Prasad et al., 2021).

A significant challenge in studying Code Switching is the scarcity of training data (D. Gupta, Ekbal, and Bhattacharyya, 2020). Existing code-switched corpora mostly involve English and one other language (e.g., English-Chinese, English-Spanish, English-Hindi), extracted from social media platforms (Barik, Mahendra, and Adriani, 2019; R. Xiang et al., 2020; Chakravarthi et al., 2020; Lovenia et al., 2022).

Methods for generating code-switched data in multiple languages have been proposed. Qin et al. (2020) and Conneau et al. (2020b) create code-switched data from downstream task datasets by randomly switching individual words to a target

language using translations from bilingual dictionaries. However, this introduces ambiguity errors and is prone to switching words without significant content. Krishnan et al. (2021) use Code Switching to improve Intent Classification and Slot Filling. Instead of switching individual words, they obtain phrase information from the slot labels and generate phrase-level code-switched sentences via automatic translations. Yang et al. (2020) create code-switched sentences by randomly substituting source phrases with their target equivalents in parallel sentences after obtaining word alignments. Z. Jiang et al. (2020) select a subset of Wikipedia sentences in four languages that contain multilingual entities from X-FACTR and create code-switched sentences by switching entities from English to non-English entities *and* vice versa, via Wikidata translations.

Our proposed `EntityCS` shares several similarities. The primary distinctions, particularly in contrast to Z. Jiang et al. (2020), can be summarised in three key aspects: (i)*Scale*: While Z. Jiang et al. (2020) focuses on a limited subset, we create a substantial corpus, magnifying the scale by a factor of 1000x, spanning across 93 languages rather than just four; (ii) *Diversity*: In contrast to the 30% sentence code-switching and single non-English entity focus in Z. Jiang et al. (2020), our approach code-switches every sentence to multiple candidate target languages. This modification enhances the model's ability to capture cross-lingual information comprehensively, resulting in improved cross-lingual transferability; (iii) *Entity-Prediction Objective*: A pivotal difference lies in our incorporation of various entity-focused prediction objectives. This strategic design enhances the entity-awareness of the intermediate-trained model, which is further shown to contribute to entity-centric downstream tasks.

4.2.3 Knowledge Integration into Language Models

As detailed in Chapter 2, Pre-trained Language Models may lack explicit grounding to real-world entities and relations, making it challenging to recover factual knowledge (Bender et al., 2021).

We refer the reader to section 2.3 for techniques of knowledge integration into PLMs that focus on monolingual models. Integrating multilingual knowledge into

XLMs has also been recently addressed. X. Jiang et al. (2022) train a model with two knowledge-related tasks: entity prediction and object entailment. They use Wiki-Data description embeddings in one language (English and non-English) to predict an entity in a target language as a classification task, preserving an entity's vocabulary. Calixto, Raganato, and Pasini (2021) use Wikipedia articles in 100 languages together with BabelNet (Navigli and Ponzetto, 2012), a multilingual sense-inventory for WSD, by predicting the WikiData ID of each entity. Another work taking advantage of entities by Ri, Yamada, and Tsuruoka (2022) uses dedicated multilingual entity embeddings on 24 languages and outperforms word-based pre-trained models in various cross-lingual transfer tasks.

4.3 Methodology

In this section, we provide a comprehensive overview of the ENTITYCS corpus construction and detail various entity-oriented masking strategies employed in our experiments.

4.3.1 ENTITYCS Corpus Construction

Wikipedia is a multilingual online encyclopedia available in more than 300 languages.[2] Structured data of Wikipedia articles are stored in Wikidata, a multilingual document-oriented database. With more than six million articles, English Wikipedia has the potential to serve as a rich resource for generating CS data. We use English Wikipedia and leverage entity information from Wikidata to construct an entity-based CS corpus.

To achieve this, we make use of wikilinks in Wikipedia, i.e., links from one page to another. We use the English Wikipedia dump[3] and extract raw text with WikiExtractor[4] while keeping track of wikilinks. Wikilinks are typically surrounded by square brackets in Wikipedia dump, in the format of [[*entity | display text*]], where *entity* is the title of the target Wikipedia page it links to, and *display text* corresponds

---

[2]https://en.wikipedia.org/wiki/Wikipedia
[3]https://dumps.wikimedia.org/enwiki/latest/ (Nov 2021 version)
[4]https://github.com/attardi/wikiextractor

to what is displayed in the current article. We then employ SpaCy[5] for sentence segmentation. To focus on entity-level code-switched instances, only sentences containing at least one wikilink are retained, and sentences exceeding 128 words are excluded from the dataset. This process results in a final ENTITYCS corpus comprising 54.5 million English sentences and 104 million entities.

She was studying [[**computer science**]] and [[**electrical engineering**]] .

WIKIDATA

| computer science | $\longrightarrow$ | Q21198 |
| electrical engineering | $\longrightarrow$ | Q43035 |

Q21198	**Q43035**
zh: 计算机科学	**zh**: 电气工程学
hi: कम्प्यूटर विज्ञान	**hi**: विद्युत अभियान्त्रिकी
fr: Informatique	. . .
. . .	**fr**: Électrotechnique
ar: علم الحاسوب	**ar**: هندسة كهربائية
el: Επιστήμη Υπολογιστών	**el**: Ηλεκτρολογία

She was studying *<e>***computer science***</e>* and *<e>***electrical engineering***</e>*.
She was studying *<e>***计算机科学***</e>* and *<e>***电气工程学***</e>*.
She was studying *<e>***कम्प्यूटर विज्ञान***</e>* and *<e>***विद्युत अभियान्त्रिकी***</e>*.
She was studying *<e>***Informatique***</e>* and *<e>***Électrotechnique***</e>*.
. . .

FIGURE 4.1: Illustration of generating ENTITYCS sentences from an English sentence extracted from Wikipedia. Entities in double square brackets indicate wikilinks.

As depicted in Figure 4.1, our Code Switching process begins with an English sentence containing wikilinks. Each entity within these links is mapped to its corresponding Wikidata ID, and translations for these entities are retrieved from Wikidata. The selection of target languages for Code Switching is based on the availability of translations for all entities within a given sentence.

We consider a set of 92 target languages (non-English), representing the overlap between languages available in Wikidata and those supported by XLM-R (Conneau et al., 2020a), the model utilised for intermediate training. To ensure coherence, all

---

[5]https://spacy.io/

STATISTIC	COUNT
Languages	93
English Sentences	54,469,214
English Entities	104,593,076
Average Sentence Length	23.37
Average Entities per Sentence	2
CS Sentences per EN Sentence	≤ 5
CS Sentences	231,124,422
CS Entities	420,907,878

TABLE 4.1: Statistics of the ENTITYCS Corpus.

entities in a sentence are code-switched to the same target language, mitigating potential noise from introducing too many languages.

To manage the size of the corpus, we generate up to five entity code-switched sentences for each English sentence. Specifically, if fewer than five languages have translations available for all entities in a sentence, we create ENTITYCS instances with all available languages. Otherwise, we randomly select five target languages from the candidates. If no candidate languages are found, we retain the sentence in the English corpus without code-switching.

In the final step, we enclose each entity with entity indicators (<e>, </e>). This ensures clear identification of entities within the code-switched sentences.

The statistical overview of the ENTITYCS corpus is presented in Table 4.1, and a histogram detailing the number of sentences and entities per language (excluding English) in the ENTITYCS corpus is illustrated in Figure 4.2.

4.3.2 Masking Strategies

To assess the efficacy of intermediate training on the generated ENTITYCS corpus, we experiment with various training objectives using an existing pre-trained language model. Initially, we adopt the conventional 80-10-10 Masked Language Modelling (MLM) objective, where 15% of sentence subwords or tokens serve as masking candidates. Among these, we replace tokens with [MASK] 80% of the time, with 10% using random tokens (from the entire vocabulary), and the remaining 10% left unchanged (Same).
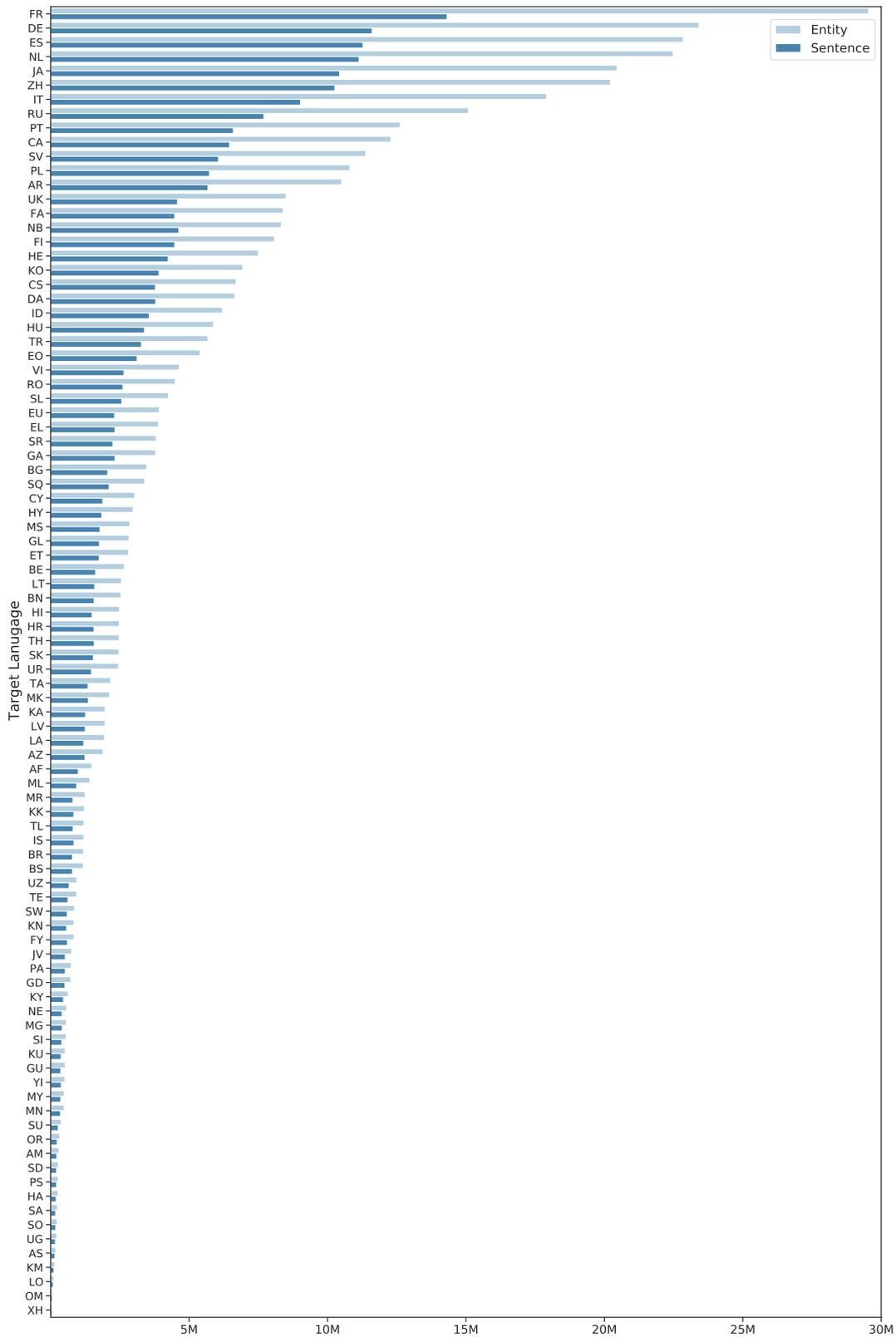
FIGURE 4.2: Number of Code-Switched Entities and Sentences in the ENTITYCS corpus.

(A) **W**hole **E**ntity **P**rediction (WEP)

(B) **P**artial **E**ntity **P**rediction (PEP)

(C) **P**artial **E**ntity **P**rediction with **MLM** (PEP+MLM)
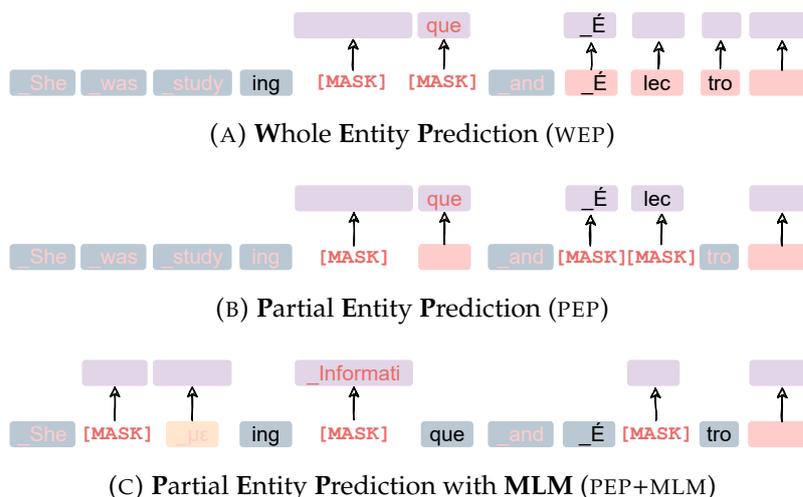
FIGURE 4.3: Illustration of the proposed masking strategies. Random tokens are chosen from the entire vocabulary and thus can be from different languages. (c) shows a case where "study" is replaced with a token in Greek.

To integrate entity-level cross-lingual knowledge into the model, we introduce Entity Prediction objectives, where we exclusively mask tokens belonging to an entity. By predicting the masked entities in ENTITYCS sentences, we anticipate the model capturing the semantics of the same entity in different languages. Two distinct masking strategies are proposed for predicting entities: Whole Entity Prediction (WEP) and Partial Entity Prediction (PEP).

In WEP, inspired by Sun et al. (2019) where whole-word masking is also employed, we consider all the words (and consequently subwords or tokens) inside an entity as masking candidates. Subsequently, 80% of the time, we mask every token inside an entity, leaving 20% unchanged. Notably, to predict the entire masked entity, we refrain from replacing it with random tokens, as it might introduce noise, leading to the model predicting incorrect entities. After masking entities, we remove the entity indicators <e>, </e> from the sentences before feeding them to the model. Figure 4.3a provides an example of WEP.

For PEP, we also consider all entities as masking candidates. Unlike WEP, we do not enforce tokens belonging to one entity to be either all masked or all unmasked. Instead, each individual entity token is masked 80% of the time. For the remaining 20% of masking candidates, we experiment with three different replacements.

PEP$_{MRS}$ corresponds to the conventional 80-10-10 masking strategy, where 10% of the remaining tokens are replaced with random tokens, and the other 10% are left

MASKING STRATEGY		ENTITY (%)				NON-ENTITY (%)			
		p	MASK	RND	SAME	p	MASK	RND	SAME
MLM		15	80	10	10	15	80	10	10
WEP		100	80	0	20	0	–	–	–
PEP_{MRS}		100	80	10	10	0	–	–	–
PEP_{MS}		100	80	0	10	0	–	–	–
PEP_M		100	80	0	0	0	–	–	–
WEP		50	80	0	20	15	80	10	10
PEP_{MRS}	+ MLM	50	80	10	10	15	80	10	10
PEP_{MS}		50	80	0	10	15	80	10	10
PEP_M		50	80	0	0	15	80	10	10

TABLE 4.2: Summary of the proposed masking strategies. p corresponds to the probability of choosing candidate items (entity/non-entity tokens) for masking. MASK, RND, SAME represent the percentage of replacing a candidate with Mask, Random or the Same item. When combining WEP/PEP with MLM (+MLM), we lower p to 50%.

unchanged. In PEP_{MS}, we remove the 10% random tokens substitution, predicting only the 80% masked tokens and 10% Same tokens from the masking candidates. In PEP_M, we further eliminate the 10% Same tokens prediction, essentially predicting only the masked tokens. An example of PEP is illustrated in Figure 4.3b. Previous work has demonstrated the effectiveness of combining Entity Prediction with MLM for cross-lingual transfer (Z. Jiang et al., 2020). Therefore, we investigate the combination of Entity Prediction objectives with MLM on non-entity tokens. Specifically, when combined with MLM, we lower the entity masking probability (p) to 50% to maintain roughly the same overall masking percentage. Figure 4.3c illustrates an example of PEP combined with MLM on non-entity tokens.

A summary of the masking strategies is presented in Table 4.2, along with the corresponding masking percentages.

4.4 Experimental Setup

After constructing the ENTITYCS corpus, we proceed to further train an XLM model, utilising XLM-R-base[6] for our experiments. We explore the effectiveness of WEP, PEP, MLM, and the joint objectives in intermediate training.

---

[6]https://huggingface.co/xlm-roberta-base

We adopt the sampling strategy proposed by Conneau and Lample (2019), where we down-sample high-resource languages and increase the sampling frequency for low-resource languages. Recognising that semantic features are often emphasised in higher layers of pre-trained language encoders (Tenney, Das, and Pavlick, 2019; Rogers, Kovaleva, and Rumshisky, 2020), we restrict training to the embedding layer and the last two layers of the model. This approach helps prevent catastrophic forgetting, a phenomenon observed in preliminary experiments when updating the entire network. We randomly select 100 sentences from each language to serve as a validation set, measuring perplexity every 10K training steps.

4.4.1   Datasets

As the ENTITYCS corpus is constructed with Code Switching at the entity level, we expect our models to mostly improve entity-centric tasks. Thus, we choose the following datasets: WikiAnn (X. Pan et al., 2017) for NER, X-FACTR (Z. Jiang et al., 2020) for Fact Retrieval, MultiATIS++ (W. Xu, Haider, and Mansour, 2020) and MTOP (Haoran Li et al., 2021) for Slot Filling, and XL-WiC (Raganato et al., 2020) for WSD.[7] The details of the datasets are introduced below.

WikiAnn (X. Pan et al., 2017) is a cross-lingual name tagging and linking dataset based on Wikipedia articles, where named entities are annotated as location (`LOC`), organisation (`ORG`) and person (`PER`) tags following the IOB2 format. The original dataset contains 282 languages. We evaluate our models on the 40 languages from WikiAnn that are included in the XTREME benchmark (J. Hu et al., 2020).

X-FACTR (Z. Jiang et al., 2020) is a multilingual fact retrieval benchmark similar to LAMA (Petroni et al., 2019). It probes factual knowledge stored in pre-trained language models by prompt-based fill-in-the-blank cloze queries, covering 23 languages. X-FACTR includes both single- and multi-token entities, and two decoding methods (independent and confidence-based) are proposed.

---

[7]The result reported on the XL-WiC for prior work is our re-implementation based on `https://github.com/pasinit/xlwic-runs`.

MultiATIS++ (W. Xu, Haider, and Mansour, 2020) is an expansion of the Multilingual ATIS (Upadhyay et al., 2018) dataset, which includes nine languages (English, Spanish, German, French, Portuguese, Chinese, Japanese, Hindi and Turkish) from four language families (Indo-European, Sino-Tibetan, Japonic and Altaic). It contains dialogues in a single domain, Air Travel Information Services. While processing the dataset, we noticed that 14 examples in the test set do not have a matching number of tokens and slot labels, which we ignored during the evaluation.

MTOP (Haoran Li et al., 2021) is a Multilingual Task-Oriented Parsing dataset that includes six languages from 11 domains that are related to interactions with a personal assistant. We use the standard flat labels as reported in Haoran Li et al. (2021).

XL-WiC (Raganato et al., 2020) is a cross-lingual word disambiguation dataset (Word in Context), formed as a binary classification problem. Given a target word and two contexts, the goal is to identify if the word is used in the same sense in both contexts. The dataset contains both nouns and verbs as target words, covers 12 languages and was created as an extension to the English WiC dataset (Pilehvar and Camacho-Collados, 2019).

4.4.2 Hyper-Parameter Settings

We introduce below the detailed setting of our intermediate training and downstream tasks fine-tuning.

Intermediate Training

We use 8 Nvidia V100 32GB GPUs for training our models on the ENTITYCS corpus, with the Hugging Face library (Wolf et al., 2020). During fine-tuning, all models were run on a single Nvidia V100 32GB GPU. We set the batch size to 16 and gradient accumulation steps to 2, resulting in an effective batch size of 256. For speedup, we employ half-precision (fp16) in the experiments. In each batch, we allow examples from multiple languages, based on the sampling strategy followed by Conneau and Lample (2019). We train for a single epoch with a maximum learning rate $5e^{-5}$ and linear decay scheduler, no warmup or weight decay, gradient clipping equal to 1.0,

PARAMETER	WIKI-ANN	MULTIATIS++		MTOP		XL-WIC
		SF	*Joint*	*SF*	*Joint*	
LEARNING RATE	$1e^{-5}$	$3e^{-5}$	$3e^{-5}$	$2e^{-5}$	$3e^{-5}$	$1e^{-5}$
WARMUP RATIO	0.1	0.0	0.0	0.1	0.1	0.0
BATCH SIZE	8	8	8	8	8	8

TABLE 4.3: Best hyper-parameters used for the datasets.

and early stopping if perplexity does not drop after 20 consecutive evaluations (we evaluate every 10K training steps).

Downstream Tasks

After intermediate training on the ENTITYCS corpus, we evaluate the zero-shot cross-lingual transfer of the models on downstream tasks by fine-tuning the model on task-specific English training data. For downstream tasks, we evaluate models on the English validation set five times per epoch following Dodge et al. (2020).

For fine-tuning XLM-R-base on WikiAnn, MultiATIS++, MTOP and XL-WiC, we fix the number of training epochs to 10, gradient clipping to 1.0, and maximum sequence length to 128. We select the batch size from $\{8, 32\}$, learning rate from $\{1e^{-5}, 2e^{-5}, 3e^{-5}, 4e^{-6}, 5e^{-6}, 6e^{-6}\}$, and warm up ratio from $\{0, 0.1\}$.

The best hyper-parameters per task are reported in Table 4.3. We choose the checkpoints with the best performance on the English validation set. For all experiments except X-FACTR, we fine-tune models with five random seeds and report average performance and standard deviation.

4.4.3   Languages for Intermediate Training

Given the size of the ENTITYCS corpus, we primarily select a subset from the total 93 languages, that covers most of the languages used in the downstream tasks. This subset contains 39 languages, from WikiAnn, excluding Yoruba.[8] We train XLM-R-base on this subset, and subsequently fine-tune the new checkpoints on the English training set of each dataset, with evaluations spanning all available languages.

---

[8]Yoruba is not included in the ENTITYCS corpus, as we only consider languages that XLM-R is pre-trained on.

MODEL	NER (F1) WIKIANN	FACT RETR. (ACC.) X-FACTR			SLOT FILLING (F1, F1/ACC.) MULTIATIS++		MTOP		WSD (ACC.) XL-WIC
		all	single	multi	SF	SF / Intent	SF	SF / Intent	
XLM-R$^{\text{PRIOR}}$	61.8	3.5	9.4	2.6	–	–	–	–	58.0
XLM-R$^{\text{OURS}}$	61.6 $_{0.3}$	3.5	9.4	2.6	71.8 $_{2.0}$	73.0 $_{0.7}$ / 89.1 $_{1.0}$	73.2 $_{0.9}$	72.5 $_{0.8}$ / 86.0 $_{0.7}$	59.1 $_{1.5}$
MLM	63.5 $_{0.5}$	2.5	6.4	1.7	72.1 $_{2.3}$	74.0 $_{0.7}$ / 89.6 $_{1.4}$	72.8 $_{0.6}$	72.7 $_{0.3}$ / **86.3** $_{0.4}$	59.3 $_{0.4}$
WEP	62.4 $_{0.7}$	**6.1**	**19.4**	3.0	71.6 $_{1.2}$	71.7 $_{0.8}$ / 89.7 $_{1.3}$	72.2 $_{0.6}$	**73.0** $_{0.5}$ / 86.0 $_{0.4}$	**60.4** $_{1.0}$
PEP$_{\text{MS}}$	63.3 $_{0.7}$	6.0	15.0	**4.3**	73.4 $_{1.7}$	**74.4** $_{0.7}$ / **90.0** $_{0.9}$	71.5 $_{0.7}$	72.7 $_{0.6}$ / 86.1 $_{0.5}$	60.2 $_{0.9}$
PEP$_{\text{MS}}$+MLM	**64.4** $_{0.5}$	5.7	13.9	3.9	**74.2** $_{0.4}$	74.3 $_{0.8}$ / 89.0 $_{0.9}$	73.0 $_{0.3}$	72.5 $_{0.6}$ / 85.8 $_{0.8}$	59.8 $_{0.8}$

TABLE 4.4: Average performance across languages on the test set of downstream tasks. XLM-R$^{\text{PRIOR}}$ corresponds to previous reported results with XLM-R-base, referring to Chi et al. (2021b) for WikiAnn, Z. Jiang et al. (2020) for X-FACTR and Raganato et al. (2020) for XL-WiC. XLM-R$^{\text{OURS}}$ shows our re-implemented results with XLM-R-base. Results (excluding X-FACTR) are averaged across five seeds with standard deviation reported as a subscript.

4.5 Main Results

The main results are reported in Table 4.4 where we compare models trained on the ENTITYCS corpus with MLM, WEP, PEP$_{\text{MS}}$ and PEP$_{\text{MS}}$+MLM masking strategies. For MultiATIS++ and MTOP, we report results of training only Slot Filling (SF), as well as joint training of Slot Filling and Intent Classification (SF/Intent).

4.5.1 Named Entity Recognition

In NER, models with CS intermediate training consistently demonstrate improvement on WikiAnn over the baseline, with PEP$_{\text{MS}}$+MLM exhibiting a substantial +2.8% absolute improvement. This outperformance extends to XL-Align[9] (Chi et al., 2021b), which employs a significant amount of parallel data (see Table 4.7). The conventional MLM objective proves similarly effective with PEP, potentially due to the overlap in entities chosen as masking candidates. However, WEP yields lower performance, suggesting that predicting entire entities from the surrounding context poses greater challenges.

Performance per language for PEP$_{\text{MS}}$ and PEP$_{\text{MS}}$+MLM is detailed in Table 4.5.[10] Notably, nearly all languages benefit from training on the ENTITYCS corpus. In the

---

[9] https://huggingface.co/microsoft/xlm-align-base
[10] Per-language results on WikiAnn for other models are reported in Table A.1 in Appendix A.

MODEL	AR	HE	VI	ID	JV	MS	TL	EU	ML	TA	TE	AF	NL	EN
XLM-R^{OURS}	44.6	51.9	68.3	48.6	59.6	63.3	72.5	**61.2**	63.2	54.3	49.3	76.3	80.7	83.4
PEP$_{MS}$	49.6	53.0	70.0	58.5	**62.0**	64.9	**75.7**	59.8	63.3	**57.7**	52.1	76.4	80.9	83.8
PEP$_{MS}$+MLM	**51.5**	**54.0**	**70.9**	**61.1**	59.3	**69.9**	74.6	59.3	**66.3**	57.6	**54.8**	**77.9**	**81.5**	**84.2**

	DE	EL	BN	HI	MR	UR	FA	FR	IT	PT	ES	BG	RU	JA
XLM-R^{OURS}	75.4	74.2	67.9	68.3	61.8	55.8	47.6	78.0	78.2	78.9	76.2	77.3	63.9	22.9
PEP$_{MS}$	75.1	76.3	72.5	70.1	**66.8**	61.5	**55.6**	78.8	78.5	78.6	75.8	78.0	66.4	21.3
PEP$_{MS}$+MLM	**75.5**	**77.1**	**74.6**	**70.7**	66.3	**65.9**	54.2	**79.5**	78.9	**80.1**	**78.2**	**79.6**	**67.7**	**23.2**

	KA	KO	TH	SW	YO	MY	ZH	KK	TR	ET	FI	HU		
XLM-R^{OURS}	66.4	48.8	4.3	**68.3**	45.4	52.7	27.7	44.2	76.9	72.4	75.6	76.9		
PEP$_{MS}$	67.0	50.2	**4.6**	66.9	44.7	55.2	26.9	48.9	77.4	**73.4**	76.6	77.8		
PEP$_{MS}$+MLM	**68.2**	**52.1**	4.0	66.4	**48.4**	**56.1**	29.8	52.0	78.6	71.9	**76.8**	78.8		

TABLE 4.5: F1-score per language on the WikiAnn test set. Results are averaged across five seeds.

optimal setting, PEP$_{MS}$+MLM, languages AR, ID, and UR exhibit the most substantial improvement (approximately +10% in AR and ID). However, EU and SW result in lower performance compared to the baseline.

Given this, we conduct a detailed analysis of the PEP$_{MS}$+MLM model, specifically focusing on NER errors categorised into five types: Tag, Span, Tag+Span, Missing Extraction, and Extra Extraction. In this context, Missing Extraction denotes instances where the model fails to identify an entity, while Extra Extraction refers to errors where a non-entity is incorrectly predicted as an entity. For this analysis, we select EU and SW (with lower F1-scores than the baseline),[11] as well as AR, ID, and UR (languages with the most significant improvement). The delta bar plot in Figure 4.4 illustrates the comparative analysis.

Compared to the baseline, AR, ID, and UR consistently improve in Tag+Span and Extra Extraction categories. All except ID show enhancement in Span detection, while most languages exhibit poorer performance in Missing Extraction. On the flip side, EU and SW result in slightly worse Span and Extra Extraction errors.

We further investigate the reasons for this behaviour. In ID, we observe that around 80% of the Span errors are due to additionally identifying the token *"ALIH"* (means "moving", "changing" in English) as the start of an entity. For example, for the input [*"ALIH"*, *"Indofood"*, *"Sukses"*, *"Makmur"*], the gold entity is *"ORG: Indofood Sukses Makmur"*, whereas the model predicts *"ORG: ALIH Indofood Sukses*

---

[11]Thai is excluded due to problematic tokenisation in WikiAnn, where everything is tokenised into individual *characters*.
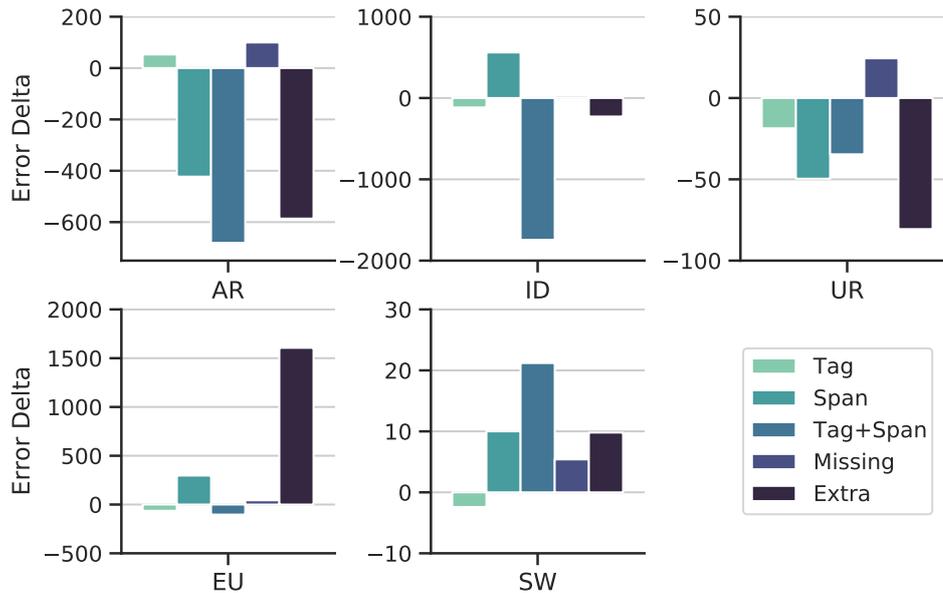
FIGURE 4.4: Error Delta (lower is better) for different types of errors in the WikiAnn test set between vanilla XLM-R-base and PEP$_{MS}$+MLM. We show error count differences for AR, ID and UR, the three languages with the largest F1-score improvement, as well as EU and SW, the two languages that underperform the baseline.

Makmur". This pattern is also observed in XLM-R, accounting for 68% of the span errors. Consultation with a native speaker reveals this may be an inaccuracy of the dataset (*"ALIH"* should not appear before the actual entities in WikiAnn).

As for EU, a lower overlap between WikiAnn entities and Wikipedia (only 47% compared to the average of 57% across all WikiAnn languages) might explain the prediction of additional entities not present in the dataset.

4.5.2 Fact Retrieval

For X-FACTR, models trained with Entity Prediction consistently outperform the baseline, while MLM performs worse than vanilla XLM-R, as anticipated. Notably, WEP achieves the best results for single-token classification, showcasing a remarkable +10% gain over XLM-R. On the other hand, PEP$_{MS}$, which involves masking part of the entity tokens for prediction, excels when dealing with multi-token entities. Interestingly, models trained on large parallel data, such as InfoXLM[12] (Chi et al., 2021a) and XLM-Align (Chi et al., 2021b), demonstrate poor performance with

---

[12]https://huggingface.co/microsoft/infoxlm-base

MODEL	LATIN SCRIPT						NON LATIN SCRIPT			
	ES	DE	FR	PT	TR	*avg*	ZH	JA	HI	*avg*
XLM-R$^{\text{OURS}}$	81.5	79.8	74.8	76.5	43.0	71.1	77.2	56.8	50.6	61.5
MLM	78.8	78.0	74.4	74.6	39.7	69.1	76.4	70.3	61.5	69.4
PEP$_{\text{MS}}$	79.3	79.7	75.3	76.2	45.3	71.1	77.8	69.0	62.9	69.9
PEP$_{\text{MS}}$+MLM	81.3	81.4	78.2	76.1	42.1	**71.8**	78.8	68.8	65.8	**71.1**

TABLE 4.6: F1-score (average across five seeds) for languages with Latin and Non-Latin script on MultiATIS++ test set when using SF-only training.

single-token accuracy of 3% and 5%, respectively, and less than 1% multi-token accuracy. This discrepancy can be attributed to their focus on sentence-level alignment. Results per language for X-FACTR are available in Table A.2 of Appendix A.

4.5.3   Slot Filling

In the case of SF-only training, the most effective model is PEP$_{\text{MS}}$+MLM, showcasing a +2.4% gain over XLM-R on MultiATIS++. This performance is competitive with the best result from XLM-Align (74.4, see Appendix A). Conversely, no improvements are observed in MTOP over the baseline. A manual inspection of the dataset reveals that this disparity can be attributed to domain differences. MultiATIS++ contains entities such as city names, whereas MTOP consists of dialogues with a personal assistant, involving tasks like setting up reminders, where fewer entities occur, limiting the benefits of entity-centric CS training.

When jointly optimising SF and Intent Classification, models also demonstrate improvements over the baseline on SF (+1.4% for MultiATIS++ and +0.5% for MTOP), albeit with lower gains. We speculate that the additional intent labels offer complementary information to the task, mitigating the impact of external information.

We subsequently categorise languages in MultiATIS++ based on whether they share the same script as English (Latin) and scrutinise their performance in SF-only training. As evident in Table 4.6, models trained on the ENTITYCS corpus exhibit significant improvements in languages with non-Latin scripts, achieving an average gain of +9.6%. This suggests that entity-focused training enables models to capture information that proves particularly beneficial for languages featuring scripts

different from English. Language-specific results on MultiATIS++ can be found in Table A.3 of Appendix A.

4.5.4 Word Sense Disambiguation

In the case of XL-WiC, we note the modest improvement across tasks, with WEP yielding the best performance at +1.3% over the baseline. This trend can be attributed to the task's nature, where our entity-based training objectives assume that disambiguation has already been addressed and is treated as implicit external information. Notably, in the evaluation of XLM-Align, which leverages parallel data, we observe no enhancement in disambiguating word-level semantics across languages (57% accuracy). Per-language performance on WSD is included in Table A.4 of Appendix A.

4.5.5 Comparison with Models trained with Parallel Data

We summarise the comparison with InfoXLM and XLM-Align in Table 4.7. Although these results are not fair comparisons for ENTITYCS, given that InfoXLM and XLM-Align utilise parallel data, it is evident that ENTITYCS consistently demonstrates competitive or superior performance across the board.

MODEL	**NER** (F1) WIKIANN	**FACT RETR.** (ACC.) X-FACTR			**SLOT FILLING** (F1) MULTIATIS++	MTOP	**WSD** (ACC.) XL-WIC
		all	*single*	*multi*	*SF*	*SF*	
XLM-R^{OURS}	61.6 $_{0.28}$	3.5	9.4	2.6	70.6 $_{1.55}$	72.3 $_{0.98}$	59.1 $_{1.52}$
INFOXLM *	62.8	1.1	3.3	0.6	73.9 $_{1.95}$	74.7 $_{0.30}$	56.9 $_{0.81}$
XLM-ALIGN **	63.7	1.5	5.0	1.0	**74.4** $_{0.29}$	**74.9** $_{0.36}$	56.9 $_{1.22}$
WEP	62.4 $_{0.68}$	**6.1**	**19.4**	3.0	71.6 $_{1.20}$	73.2 $_{0.89}$	**60.4** $_{0.97}$
PEP$_{MS}$	63.3 $_{0.70}$	6.0	15.0	**4.3**	73.4 $_{1.70}$	71.5 $_{0.67}$	60.2 $_{0.85}$
PEP$_{MS}$+MLM	**64.4** $_{0.50}$	5.7	13.9	3.9	74.2 $_{0.43}$	73.0 $_{0.33}$	59.8 $_{0.75}$

TABLE 4.7: Comparison with models using parallel data. Results for InfoXLM and XLM-Align are obtained from Chi et al. (2021a) and Chi et al. (2021b), respectively.

MODEL		WIKIANN	X-FACTR			MULTIATIS++	
			all	*single*	*multi*	*SF*	*SF / Intent*
XLM-R[OURS]		61.6	3.5	9.4	2.6	70.6	73.0 / 88.9
MLM	EN	61.0	1.1	2.7	0.7	71.5	72.1 / 89.6
	39	**63.5**	2.6	6.4	1.7	72.5	**73.8 / 90.2**
	93	63.3	**2.7**	**6.8**	**1.8**	**72.7**	73.4 / 89.6
WEP	EN	61.9	3.3	8.5	1.6	**71.8**	72.2 / **91.1**
	39	**62.4**	**6.1**	**19.4**	**3.0**	71.1	71.7 / 89.7
	93	59.4	5.8	18.6	2.7	70.4	**72.9** / 90.3
PEP$_{MS}$ +MLM	EN	61.2	2.7	6.6	1.6	71.3	72.3 / 90.7
	39	**64.4**	**5.7**	**13.9**	**3.9**	**73.4**	**74.4 / 90.0**
	93	63.6	5.5	13.2	3.8	72.8	72.7 / **90.8**

TABLE 4.8: Results (average over five seeds) with a different number of pre-training languages.

4.6   Analysis

We conduct further analysis on different masking strategies, examining their impact across languages and training steps during intermediate training on the ENTITYCS corpus. Our primary focus is on WikiAnn, which encompasses the largest number of languages from the datasets we evaluate.

4.6.1   Performance vs Languages in Intermediate Training

For WikiAnn, X-FACTR, and MultiATIS++, we conduct additional experiments by training MLM, WEP, and PEP$_{MS}$+MLM with varying numbers of languages in the ENTITYCS corpus. We explore three scenarios: using English only (no Code Switching), employing the subset of 39 languages (as mentioned in subsection 4.4.3), and incorporating all 93 languages.

As shown in Table 4.8, models trained solely on English sentences do not exhibit a noticeable improvement in average performance across languages (except for Intent Classification accuracy and WikiAnn with WEP over the baseline). However, English-only training proves beneficial for English performance, with an average gain of +23.1% for single-token and +5.6% for multi-token predictions in X-FACTR over the baseline XLM-R, using WEP.

When trained with all 93 languages, models employing all masking strategies demonstrate improved performance compared to XLM-R. However, the prevailing

trend indicates that training on a more restricted set of languages generally results in better performance. This suggests that incorporating a broader array of languages may not necessarily contribute to superior results, underscoring the non-trivial challenge of scaling to too many languages. Notably, the subset of 39 languages already covers most languages in the downstream tasks. In scenarios where additional languages are introduced, an increase in the number of pre-training languages could potentially lead to improved performance.

4.6.2 Performance vs Training Steps

Figure 4.5 provides a comparative analysis of different training objectives on the EN-TITYCS corpus, showcasing their performance across the number of training steps on the WikiAnn test set. The figure reveals that most masking strategies reach a plateau after the middle of training. Notably, objectives involving MLM training exhibit a clear performance increase across the board, underscoring the benefits of joint training for entities and non-entities, not only in terms of performance but also in terms of smoother learning curves. Importantly, all objectives consistently outperform the baseline throughout the training process.

It's worth noting that the observed gains from Code Switching intermediate training do not stem from additional training data. This is substantiated by Table 4.8, where the WikiAnn F1-score trained on English-only sentences (without Code Switching) shows that additional English-only training data does not improve upon the XLM-R baseline (61.6). This observation is further supported by Figure 4.5 at step 200K, corresponding to the steps required for training on the English-only sentences. At this point, all models exhibit an F1 score above 62.3, indicating that the NER performance gain can be attributed to the design of the ENTITYCS corpus and the associated training objectives.

4.6.3 Random and Same Token Prediction

We further investigate the influence of Random token substitution and Same token prediction when implementing PEP. By comparing PEP_{MRS}, PEP_{MS}, and PEP_M in Figure 4.5, a clear trend emerges: incorporating Random token substitution in PEP
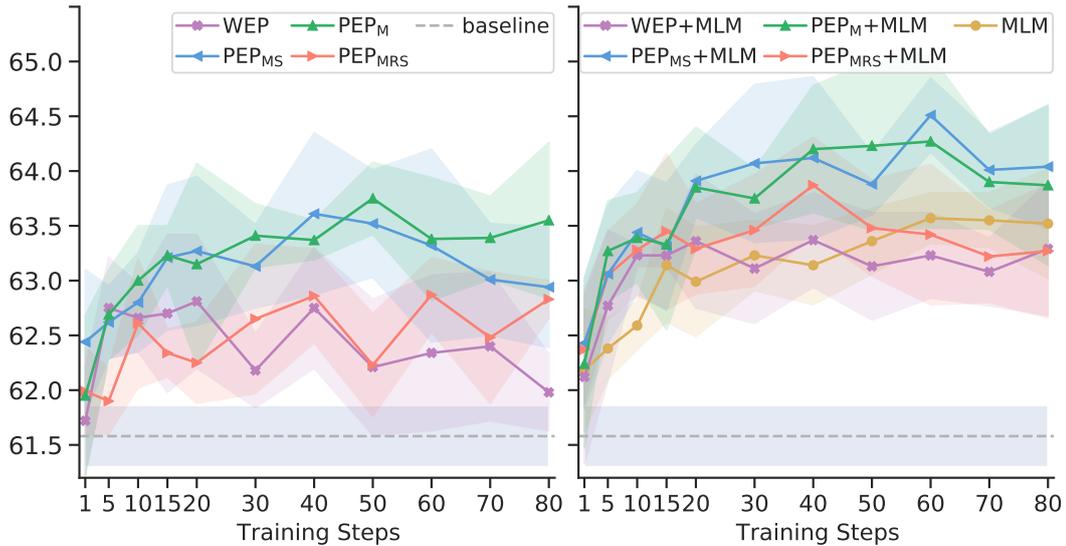
FIGURE 4.5: F1-score comparison on WikiAnn test set (average across five seeds) as a function of the number of training steps (in *ten* thousands) with various masking objectives. EP-only strategies are on the left, and EP + MLM strategies are on the right.

leads to inferior performance while omitting the prediction of Same tokens has a negligible effect.

Intuitively, Random token substitution introduces the risk of predicting an incorrect entity, potentially undermining the model's learning by steering it towards inaccurate predictions from the randomly replaced tokens. On the contrary, predicting the Same token involves a more straightforward task of replicating the input entity to the output. Consequently, models appear to neither gain nor lose performance with or without the Same token prediction.

These observations align with the findings in Wettig et al. (2023), albeit their focus on monolingual settings. A similar pattern can also be observed when combining PEP with MLM.

4.6.4 Entity Masking Percentage

To assess the impact of the percentage of the entity masking candidates during training, we increment the masking probability (p) from 50% to 80% and 100% for the best model, PEP_{MS}+MLM, and evaluate its effects on the WikiAnn test set.

Notably, we observe that a further increase in the masking percentage leads to a performance decline, from a $64.4_{\pm 0.5}$ F1-score to $63.8_{\pm 0.7}$ at 80%, and $64.0_{\pm 0.4}$ at

100%. We posit that masking too many tokens renders the task of entity prediction from the remaining context more challenging. However, the minimal performance difference between percentages is noteworthy, likely stemming from the observation that, on average, only two entities exist per sentence, as demonstrated in Table 4.1.

4.7 Conclusion and Future Work

In summary, this chapter underscores the valuable role of multilingual knowledge sourced from Wikipedia and Wikidata in enhancing zero-shot transfer learning for entity-oriented tasks through entity-level Code Switching. The creation of the EN-TITYCS corpus, leveraging English Wikipedia and Wikidata, facilitates the substitution of entities in wikilinks with their counterparts in multiple languages. By introducing entity-oriented training objectives, we consistently improve performance across various datasets, including Named Entity Recognition, Fact Retrieval, and Word Sense Disambiguation, surpassing baseline models and outperforming previous methods relying on extensive parallel data. Notably, our approach demonstrates competitive performance in Slot Filling.

Our findings highlight the task-specific optimal nature of different masking strategies, revealing that Whole Entity Prediction excels when emphasising single-token factual knowledge, while Partial Entity Prediction is particularly beneficial for entity typing and multi-token factual retrieval. Simultaneously predicting non-entity and entity tokens proves advantageous for tasks where the entire input context plays a crucial role, with a notable impact on languages with non-Latin scripts.

The generic nature of our corpus construction process allows scalability to a more extensive range of languages, underlining the broader applicability of our methodology. Further research could explore Code Switching beyond entities, encompassing verbs and phrases. However, it is important to acknowledge the potential challenges associated with automatically code-switching parts of sentences beyond entities, particularly in the absence of readily available aligned corpora. We hypothesise that a specifically trained module is likely required to discern suitable target candidates, especially when multiple translations are available. This module would need

to understand the context of the sentence and ensure the selection of the most appropriate option for code-switching that does not alter the semantic meaning or the syntax of the original sentence. On the other hand, future works may consider incorporating additional sources beyond Wikipedia and Wikidata (which requires an entity linking module), which provide insights into the approach on the more general domains. Overall, our study reinforces the potential of leveraging multilingual knowledge sources in advancing cross-lingual transfer learning.

4.8   Limitations

The limitations of this chapter are outlined as follows:

- **Morphological Inflexion Oversight**: Before code-switching an entity, its morphological inflexion is not verified. This can potentially introduce errors, as the form of the code-switched entity might not align with the surrounding context (e.g., plural form). Addressing this issue is crucial for future versions of the corpus.

- **Language Diversity Constraint**: The diversity of languages in the ENTITYCS corpus is constrained to the overlap between WikiData and XLM-R pre-training languages. While this decision facilitates a more robust model comparison, there is potential to enrich the corpus with additional languages not covered by XLM-R.

- **Task Specificity in Evaluation**: The proposed approach's primary evaluation is centred on entity-centric tasks. The exploration of broader natural language understanding tasks, such as Natural Language Inference, is feasible, and investigating the impact on such tasks may provide more insights into the generalisability of the models.

- **Unidirectional Code-Switching**: Code-switching is performed only from English to other languages while maintaining the context in English. Although the choice is based on the abundance of English resources for studying cross-lingual transfer, exploring bidirectional code-switching, particularly from non-English articles to English, is a promising direction for future research. We

anticipate that a model trained using bidirectional code-switching could yield similar enhancements, as observed in prior work by Z. Jiang et al. (2020), however, one limiting factor may be the availability of Wikipedia articles in very low-resource languages.

- **Model Size**: The experiments were conducted solely with base-sized models for speed considerations. Extending these experiments to larger models is a logical progression that could provide valuable insights into scalability and performance improvements.

Chapter 5

Faithful and Robust Knowledge Extraction on the Web

In the previous two chapters, we explored how structured knowledge enhances NLP applications, including fake news detection and multilingual entity-centric tasks. This chapter focuses on investigating faithful and robust methods for extracting information or structured knowledge from web text.

The main content of the chapter is based on the paper "WEBIE: Faithful and Robust Information Extraction on the Web" (Whitehouse et al., 2023b) published in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

5.1 Background and Introduction

Information Extraction (IE) is the task of extracting structured information from unstructured text, typically presented in the form of triples *<subject, relation, object>*. It is essential for many Natural Language Processing applications such as knowledge base population, question answering, faithful summarisation, and fake news detection (Trisedya et al., 2019; Huguet Cabot and Navigli, 2021; Narayan et al., 2021; Whitehouse et al., 2022).

Closed IE systems, specifically those extracting triples with predefined entities and relations from a knowledge base (KB), require two essential pieces of information: (i) the entities mentioned in the text and (ii) the relations between each pair of entities. Due to the expense of annotations, most existing IE datasets, like WikiNRE

(Trisedya et al., 2019) or REBEL (Huguet Cabot and Navigli, 2021), are constructed using Wikipedia. Entities are linked through wikilinks, and relations are automatically extracted using a distant supervision (DS) approach (Mintz et al., 2009) based on a KB such as Wikidata, which assumes that if two entities share a relation in a KB, sentences mentioning both entities express that relation.

While models trained only on this fact-rich domain[1] have shown to be useful for IE applications, they have limited capacity when applied to extracting information in other web domains, which often contains noisy text or text without any factual information. For example, AllenAI's C4 dataset,[2] an open-sourced version of Google's C4 (Raffel et al., 2020b) dataset based on Common Crawl, demonstrate this challenge. Our analysis using the DS approach reveals that fewer than 15% of the sentences contain triples (refer to subsection 5.2.3). In other words, the remaining sentences extracted from C4 either do not contain entities or do not have entities that share a relation. In such cases, faithful IE models are expected *not* to output triples. However, a state-of-the-art generative IE model, GenIE (Josifoski et al., 2022), trained on REBEL, generates triples for nearly every sentence, leading to a high false positive rate and hallucination issues.

To address these challenges and facilitate future work on IE on the web, we present WEBIE, the first large-scale, entity-linked closed IE dataset collected from web sources. The WEBIE dataset is collected from the 200 most frequent URL domains from the C4 dataset. First, we use ReFinED (Ayoola et al., 2022), a state-of-the-art Entity Linking (EL) model to identify mention spans of the entities and link them to Wikidata. We then apply the DS approach to extract triples and use a Natural Language Inference (NLI) model to filter out triples not expressed by the sentence. We also include negative examples, i.e., sentences without any factual information, to better reflect the data on the web. Our final dataset consists of 1.6M sentences, and we annotate a subset of ~21K triples through crowdsourcing. The annotated set is exclusively used as part of the test set to allow more reliable evaluation. Finally, we introduce mWEBIE, which contains human-corrected translations of the annotated version of WEBIE in four languages: French, Spanish, Portuguese, and Hindi.

---

[1]We use the term *domain* to refer to the URL domain.
[2]We use the dataset from `https://huggingface.co/datasets/allenai/c4`.

Previous studies have highlighted the superiority of generative models over discriminative pipelines, which often suffer from accumulative errors due to separate Entity Linking and Relation Extraction (RE) steps (Mesquita et al., 2019; Trisedya et al., 2019; Josifoski et al., 2022). Therefore, we primarily benchmark WEBIE with generative, transformer-based encoder-decoder models, BART (M. Lewis et al., 2020) and mBART (Tang et al., 2021). The latter is used for zero-shot cross-lingual transfer performance evaluation on mWEBIE.

Additionally, we propose three training strategies (subsection 5.3.2) that utilise entity linking as an auxiliary task for generative IE: joint generation with the linked-entity prompt (ENTITY-PROMPT), multi-task learning with distinguished artificial prompt tokens (ARTIFICIAL-PROMPT), and training with an additional task-specific language model (LM) head (2LM-HEADS). Our experiments demonstrate that incorporating entity-linking objectives leads to better and more faithful IE results. An illustration of these training strategies is provided in Figure 5.1.

Our experiments reveal that models trained on WEBIE are more robust and generalisable compared to models trained solely on Wikipedia datasets. These models achieve a new state-of-the-art performance on REBEL (section 5.5) and competitive zero-shot performance on WikiNRE. We demonstrate that WEBIE serves as a complementary dataset to existing Wikipedia-based datasets and emphasise the significance of including negative examples to address false positives in generative IE.

Our main contributions are as follows:

- We introduce (m)WEBIE, the first large-scale, entity-linked IE dataset from the web, with a subset annotated by humans and translated into four other languages.

- We propose and assess the effectiveness of using entity linking as an auxiliary task for generative IE through various training strategies.

- Comprehensive experiments demonstrate that models trained on WEBIE exhibit enhanced generalisability in Information Extraction from the web domain, including competitive zero-shot performance on IE tasks on Wikipedia.

BART

Obama was born in Honolulu, Hawaii. → BART → <sub> Barack Obama <rel> place of birth <obj> Honolulu <et> <sub> Barack Obama <rel> place of birth <obj> Hawaii <et> <sub> Honolulu <rel> located in <obj> Hawaii <et> ...

+ ENTITY-PROMPT

Obama was born in Honolulu, Hawaii. → BART → [ENTITY] Obama # Barack Obama | Honolulu # Honolulu | Hawaii # Hawaii [TRIPLE] <sub> Barack Obama <rel> place of birth <obj> Honolulu <et> <sub> Barack Obama <rel> place of birth <obj> Hawaii <et> <sub> Honolulu <rel> located in <obj> Hawaii <et> ...

+ ARTIFICIAL-PROMPT

<#el#> Obama was born in Honolulu, Hawaii.

<#tri#> Obama was born in Honolulu, Hawaii.

→ BART →

Obama # Barack Obama | Honolulu # Honolulu | Hawaii # Hawaii

<sub> Barack Obama <rel> place of birth <obj> Honolulu <et> <sub> Barack Obama <rel> place of birth <obj> Hawaii <et> <sub> Honolulu <rel> located in <obj> Hawaii <et> ...

+ 2LM-HEADS

Obama was born in Honolulu, Hawaii. → BART →

LM-Head 1 → Obama # Barack Obama | Honolulu # Honolulu | Hawaii # Hawaii

LM-Head 2 → <sub> Barack Obama <rel> place of birth <obj> Honolulu <et> <sub> Barack Obama <rel> place of birth <obj> Hawaii <et> <sub> Honolulu <rel> located in <obj> Hawaii <et> ...

FIGURE 5.1: Illustration of the training strategies. The blue and green text refer to *mention span* and its corresponding *Wikipedia title* (used as entity labels). For standard BART training, the target output is the linearised triples (subsection 5.3.1). For ENTITY-PROMPT, the target is the EL output (subsection 5.3.2) concatenated with the linearised triples. In ARTIFICIAL-PROMPT, we prepend an artificial token to the input to indicate the desired output, EL (yellow box) or linearised triples. For 2LM-HEADS, we add an additional task-specific LM head to the decoder for the EL task (grey box).

5.2 Related Work

We review the related work from the following two aspects: datasets and approaches for Information Extraction.

5.2.1 Information Extraction Datasets

The term Information Extraction has been used for different tasks in the literature. Most existing IE datasets are collected from Wikipedia articles aligned with Wikidata, including sentence-level IE datasets such as REBEL, WikiNRE, FewRel (Han et al., 2018), T-REx (Elsahar et al., 2018); document-level Relation Extraction[3] datasets, e.g., DocRED (Y. Yao et al., 2019), CodRED (Y. Yao et al., 2021). SMiLER (Seganti et al., 2021) is a multilingual sentence-level IE dataset that is also based on Wikipedia, covering 14 languages and 36 relations. These sentence-level IE datasets typically do not contain negative examples. Although it is possible to augment these datasets with negative instances sourced from Wikipedia, they do not adequately reflect the true data distribution on the web. To enable more general information extraction models beyond solely relying on Wikipedia articles, we argue for the necessity of datasets that accurately mirror web data – a point later confirmed in our experiments (refer to section 5.5).

Datasets such as TACRED (Yuhao Zhang et al., 2017), RE-TACRED (Stoica, Platanios, and Poczos, 2021), and WebRED (Ormandi et al., 2021) contain negative relation examples, but they lack linkage to knowledge bases. Furthermore, applying entity linking to a knowledge base in a post-hoc fashion is not trivial, as these relation extraction datasets like WebRED primarily focus on *relation*, where the subject and object in a triple can be a name such as *Alice*, which may not be linkable to a KB. Therefore, our proposed dataset WEBIE stands out from existing datasets in that it pertains to the web domain, is entity-linked, and includes negative examples.

5.2.2 Information Extraction Approaches

IE approaches can be classified into two categories: pipeline systems with discriminative models, and sequence-to-sequence systems with generative models. Pipeline models typically include separate modules for Named Entity Recognition (NER), Entity Linking and Relation Extraction (Chaganty et al., 2017; Yamada et al., 2020b).

---

[3]We consider RE dataset as the ones that focus on extracting relations but without entity spans and/or linking information.

Systems that jointly train NER, EL, and RE, have also been explored, taking advantage of the information shared among the tasks (B. Ji et al., 2020; Eberts and Ulges, 2020).

In recent years, generative IE has gained a lot of attention. Nayak and Ng (2020) utilise an LTSM model and propose a pointer network-based decoding. More recent approaches, e.g., as introduced in REBEL and GenIE, train a transformer-based encoder-decoder model with standard maximum-likelihood objectives to convert sentences to linearised output. KnowGL (Rossiello et al., 2023) improves upon REBEL with additional entity type information added to the linearised output. Our work extends GenIE and experiments with three different approaches where we incorporate explicit EL information as an auxiliary task with adapted constraint Trie decoding.

5.2.3   WEBIE Collection

In this section, we provide a detailed explanation of the dataset collection process for (m)WEBIE.

Data Pre-processing

We begin the process with the English segment of AllenAI's C4 dataset, retaining the most frequent 200 URL domains. Subsequently, one million documents are randomly sampled, and SpaCy[4] is employed for sentence segmentation. Sentences containing fewer than 10 words are removed, resulting in approximately 20 million sentences.

Entity Linking and DS Dataset

Next, we run ReFinED (Ayoola et al., 2022), a state-of-the-art EL model on the sentences to identify entity spans and link them to their corresponding Wikidata ID. Besides named entities, ReFinED also extracts *numerical* entities that do not have Wikidata ID. In this work, we only consider numerical entities that express dates and map them to the corresponding *year* for simplicity.[5] Some examples of ReFinED processed output are included in Table B.2 in Appendix B.

---

[4]https://spacy.io/
[5]For example, "October 10, 2018" will be mapped to "2018".

After obtaining the entity-linked sentences, we apply the DS paradigm to retrieve the set of relations that exist between each pair of entities in each sentence using Wikidata (September 2022 dump) as our KB and build a DS dataset. After the above steps, we obtain WEBIE DS dataset consisting of 21.2M entities and 4.8M triples.

Entailment Filtering

One major drawback of the DS approach is that the triples extracted may or may not be expressed by the source sentence (Riedel, L. Yao, and McCallum, 2010). Following previous work on obtaining a cleaner version of the DS dataset (Huguet Cabot and Navigli, 2021; Vania, Grace Lee, and Pierleoni, 2022), we apply an NLI model, `nli-deberta-v3-large`,[6] that is trained on SNLI (S. R. Bowman et al., 2015) and MultiNLI (Williams, Nangia, and S. Bowman, 2018), to filter out triples that do not entail the sentence. Each source sentence is treated as the *premise* and we use manually created templates (similar to Vania, Grace Lee, and Pierleoni (2022)) to convert a DS triple to one or more *hypotheses*.

We then obtain the entailment probability score for each premise-hypothesis pair and take the maximum score for cases with multiple converted hypotheses. We set the threshold to be 0.7, similar to Huguet Cabot and Navigli (2021), and only keep triples with an entailment score above the threshold. We retain 2.1M triples (44% of the previous DS triples, see Table 5.1) after this filtering process.

Negative Examples

After the DS creation and NLI filtering steps, only less than 10% of the original sentences contain triples. To train models for extracting facts from the web and alleviate false positives, we include two kinds of negative examples in WEBIE: (i) sentences with one or zero entities, and (ii) sentences with two or more entities, but without any factual information (i.e., no relation between the entities). We randomly sample negative instances covering both cases evenly and add them to WEBIE. In the end, WEBIE consists of 1.6M sentences, where 50% are negative examples. A summary of

---

[6]https://huggingface.co/cross-encoder/nli-deberta-v3-large achieved superior results among the models we evaluated in our preliminary experiments.

Dataset	Domain	Entity Linked	Relation Type	Sentence	Train†	Val†	Test†	Triple	Anno. Triple	Negative Instance	Language (Test Set)
TACRED	Web	✗	42	106,264	68,124	22,631	15,509	106,264	106,264	79.5%	1
WebRED	Web	✗	523	117,717	–	–	–	117,717	117,717	65%	1
WikiNRE	Wikipedia	✓	158	255,654	224,881	988	29,785	330,005	0	0	1
REBEL	Wikipedia	✓	1146	3,059,894	2,754,387	152,672	152,835	10,311,293	0	0	1
WebIE	Web	✓	661	1,649,167	1,480,223	82,914	86,030	1,905,205	21,113	50%	5

TABLE 5.1: Statistics of WebIE and comparison with other sentence-level RE (top two rows) and IE datasets. Sentences correspond to the number of distinct sentences. We report the publicly available version of WebRED. † shows the number of examples in each split. Anno. Triple represents the number of human-annotated triples.

the statistics of WEBIE with a comparison with other datasets is shown in Table 5.1. The dataset is randomly split into train/validation/test sets using a 90/5/5 split.

5.2.4 Human Annotation

Existing IE datasets, such as REBEL, are often automatically annotated using the DS approach, hence the labels can be noisy. To allow a more reliable evaluation of WEBIE, we randomly sample ~21K triples from the most frequent 200 relations and annotate them with MTurk. Given a sentence, each HIT (Human Intelligence Task) is designed to verify if a DS triple is correctly expressed in the sentence.[7] First, the annotators are asked to verify if the head entity (subject) and tail entity (object) are linked correctly. For each entity, we provide its Wikipedia title and link to its Wikidata page as additional context. After that, the annotators are asked to verify if the triple relation is correctly inferred from the sentence. Here, we provide the relation descriptions and example use cases of each relation. We ask three MTurk workers to annotate each DS triple and take the majority vote as the final label for each triple. A triple is considered valid if both entities are linked to the correct Wikidata entities and the relation is inferred[8] by the sentence. An annotation interface is shown in Appendix B.

To ensure the annotation quality, we set qualifications with additional requirements for MTurk workers (see Appendix B for details). The agreement among the three annotators is high: 99.4% for the head entities, 99.2% for the tail entities, and

---

[7] We ensure *all* DS triples in a selected sentence are annotated.

[8] We ask for *inferred* instead of explicit expression since some relations may not be explicitly expressed in the sentence, e.g., "located in" (London, UK) or "date of birth" XX (1986-2022).

76.1% for the relations have all three annotators agreeing on the same label. After the majority vote, 92.1% of the triples are labelled as inferred and therefore kept as valid triples.

5.2.5 Multilingual WEBIE

To enable zero-shot cross-lingual transfer evaluation on WEBIE, we further extend the annotated subset, with additional negative examples, to four other languages: French, Spanish, Portuguese, and Hindi. First, we use a neural machine translation model, the distilled 1.3B variant,[9] of NLLB-200 (Costa-jussà et al., 2022) to translate the English sentences into the target languages. We then use MTurk to verify the translation and add entity span information in the translated sentences. We provide the English sentence (with the entity spans highlighted) and its translation, and first, ask the annotators to correct the translation. After that, MTurk workers are asked to mark the corresponding entity spans in the target language. We ask two annotators to complete the aforementioned HIT, and an additional worker to select the better translation, which is used in our final dataset. To obtain translations with higher quality, we restrict the region of the workers to countries where the target language is the official language.[10] The final mWEBIE consists of 9K instances in each language, which corresponds to roughly 90% of the 21K annotated triples.

5.3 Generative Information Extraction

This section describes the training strategies we use for benchmarking (m)WEBIE.

5.3.1 Sentence-to-Triples Generation

We use BART and mBART for all of our experiments. Given a sentence *s* as input, we train the model to autoregressively generate the linearised triples *t* as an output. Following the practice from Huguet Cabot and Navigli (2021) and Josifoski et al. (2022), we linearise a triple t_i by converting it into "`<sub> head entity label`

---

[9]https://huggingface.co/facebook/nllb-200-distilled-1.3B
[10]See details for mWEBIE annotations in section B.3.

`<rel>` relation `<obj>` tail entity label `<et>`", where the tags in brackets represent **sub**ject, **rel**ation, **obj**ect, and the **e**nd of **t**riple, respectively. Head/tail entity label refers to the Wikipedia title that the mention span in the sentence is mapped to, which also has a one-to-one correspondence with the Wikidata ID.[11]

For each sentence, we order its linearised triples according to the order in which they appear in the input sentence; first by the order of the appearance of the head entity, and then by the order of the tail entity (for cases when the head entities are the same). The conditional probability of generating t is formulated as $p(t|s) = \prod_{t=0}^{N} p(t_i|t_{<i}, s)$. We use the standard cross-entropy loss and maximise the output sequence likelihood with teacher forcing (Sutskever, Vinyals, and Le, 2014). An example of input and output can be seen at the top of Figure 5.1.

5.3.2   Entity-Linking as an Auxiliary Task

The standard linearised triples output only contains the label of the entity and not the span. As a result, it may be difficult to trace back from which input span an entity is generated, especially in the case when the model hallucinates (e.g., by generating an entity that is not mentioned in the sentence). To encourage models to generate faithful and interpretable output, we also experiment with models that are jointly optimised for generating triples and EL. The goal of the EL task is to identify and extract entity spans from the input sentence and link them to their corresponding KB entities. We posit that adding the EL task as an additional training objective will teach the model to attend to the input spans when generating the output.

We experiment with the following three approaches.

ENTITY-PROMPT

Narayan et al. (2021) and Narayan et al. (2022) have shown that generation with entity-chain planning, i.e., generating the desired entities first before the actual output, is effective in improving the faithfulness and controlling hallucinations in text generation tasks such as abstractive summarisation. For generative IE tasks, EL can be used as an intermediate plan to ground the generation of the linearised triples. We

---

[11]For example, a mention span of "UK" is linked to Wikipedia title "United Kingdom" and mapped to Q145 in Wikidata.

define the Entity-Linking target in the format of "Mention Span$_1$ # Entity Label$_1$ | Mention Span$_2$ # Entity Label$_2$ | ...", where we order the entity spans as they appear in the text. We then prepend the Entity-Linking target to the linearised triples target, using special symbols as separators, i.e., "[ENTITY] Entity-Linking target [TRIPLE] Linearised Triples Target". Here "[ENTITY]" is the start symbol before generating the EL output, and "[TRIPLE]" is the start symbol before generating the linearised triples. Given an input sentence, we essentially train the decoder to first generate the EL chain and then generate the triples, conditioned on both the input sentence and the EL output.[12]

ARTIFICIAL-PROMPT

Artificial Prompt tokens are symbols placed in front of the input sequence, which has previously been explored in areas such as neural machine translation to distinguish the language of the target output translation (M. Johnson et al., 2017). We adapt this approach for jointly training our models for Entity Linking and generative IE. Specifically, we use an artificial prompt token <#el#> at the beginning of the input sentence when training for the Entity-Linking target, and use <#tri#>[13] for the linearised output target. Training instances for both tasks are mixed and randomly shuffled for training.

2LM-HEADS

Finally, inspired by Gontier, Reddy, and Pal (2022), the third approach that we experiment with is the addition of a second language model (LM) head in the decoder, which is initialised with the same weights as the first (standard) LM head. The first LM head is optimised for generating the linearised triples while the second LM head is optimised for the EL task, thus each training instance has two different target outputs. During training, the input sentence is fed to the encoder once, and different

---

[12]The EL target only includes mention spans that contribute to valid triples, consistent with the triples that are later generated conditioned on the linked entities.

[13]Both artificial prompt tokens are added as the special tokens to the tokenizer to avoid bias from pre-trained embeddings, but are intended to be biased to the associated task.

target outputs are given to the *same* decoder. Each task-specific LM head is then responsible for generating output targeted for it. The training loss is then formulated as a weighted sum of the losses from both tasks: $\mathcal{L} = \alpha\mathcal{L}_{\text{IE}} + (1-\alpha)\mathcal{L}_{\text{EL}}$.

5.3.3 Inference with a Constraint Trie

In addition to standard beam search decoding, we experiment with constraint decoding by restricting the generated output to be valid Wikipedia titles and Wikidata relations using a prefix Trie, following the ideas proposed in GENRE (N. D. Cao et al., 2021) and GenIE (Josifoski et al., 2022). The constraint Trie improves generative IE by ensuring that the generated strings are valid entities or relations. We construct two constraint Tries: an entity Trie and a relation Trie, following the same method as introduced by N. D. Cao et al. (2021). The entity Trie is built using all Wikipedia titles (as the entity labels), and the relation Trie is built using all Wikidata relation property labels.

We use four special symbols, `<sub>`, `<rel>`, `<obj>` and `<et>` to define the state of the generation. We apply both constraint Tries as follows. We adopt the constraint Trie so that, in the very first decoding state, the model is allowed to either (i) return an empty string for a negative example, or (ii) generate `<sub>`, which is the start symbol for generating a triple. If the `<sub>` symbol is generated, then we generate the head entity using the entity Trie, i.e., only valid entities will be considered. Once the generation of the head entity is completed, the model proceeds to generate `<rel>` (i.e., the start symbol for generating relation string) and then subsequently generate allowed tokens from the relation Trie which is built from the relations in Wikidata. After that, the model generates `<obj>` and the tail entity, in the same manner, using the entity Trie. After generating the full triple (indicated by `<et>` generated after the tail entity), the decoder can either stop the generation or start a new iteration for generating the next triple.

For the ENTITY-PROMPT models, since the entity mention spans are text from the input sentences and usually are not the same as the entity labels in Wikidata, we propose a *partial* constraint generation approach. Specifically, we start the standard

beam search for the EL target output and only activate the Trie constraints after that when generating the linearised triples.

5.4 Experimental Setup

In this section, we explain the datasets used in the experiments and the detailed modelling setup.

5.4.1 Dataset

In addition to our proposed WEBIE dataset, we also use the following datasets for our experiments.

WikiNRE (Trisedya et al., 2019) is an IE dataset based on Wikipedia which is automatically constructed by aligning Wikipedia sentences to Wikidata triples using the DS approach. The authors apply a coreference resolution model (K. Clark and Manning, 2016) to obtain sentences with implicit entity names and use a paraphrase detection model (Ganitkevitch, Van Durme, and Callison-Burch, 2013; Grycner and Weikum, 2016) to filter out sentences that do not express the DS triples. In our experiments, we only use WikiNRE for zero-shot evaluation.

REBEL (Huguet Cabot and Navigli, 2021) is a large-scale IE dataset constructed automatically from Wikipedia abstracts. Using the Wikipedia hyperlinks in the abstracts, as well as numerical values and dates, they map the entity spans to their corresponding Wikidata entities. They then use the DS approach to identify triples in each sentence. To filter out false positives, the authors use an NLI model by *concatenating* the entities and the relation as the hypothesis. In our experiment, we use the REBEL dataset that is sub-sampled by Josifoski et al. (2022), where 857 relations are considered. Both WikiNRE and REBEL do not contain negative examples and are not annotated by humans.

5.4.2   Models

We experiment with BART using two settings: BART_{PLM} with the pre-trained weights from M. Lewis et al. (2020),[14] and $\text{BART}_{\text{RAND}}$, using the same configuration and architecture but randomly initialised weights. Across the two settings, Josifoski et al. (2022) find that $\text{BART}_{\text{RAND}}$ generates better results than BART_{PLM} on REBEL. For mWEBIE, we experiment with the mBART-50[15] model (for simplicity we refer to it as mBART in this thesis).

To compare models trained on different datasets, we train both BART_{PLM} and $\text{BART}_{\text{RAND}}$ on REBEL (R), WEBIE (W), and both datasets together (R+W). We evaluate the performance of the generated triples by parsing the linearised output to a list of triples and comparing it to the gold label to calculate precision, recall, and F1 scores. For WEBIE, we also calculate the accuracy of the prediction of negative instances, where a prediction is considered correct if the model accurately generates empty strings rather than hallucinating triples.

For training with EL as an auxiliary task, we primarily experiment with the $\text{BART}_{\text{RAND}}$. We prepare the training instances as described in subsection 5.3.2, and train separate models on REBEL and on WEBIE. For the 2LM-HEADS, we conduct experiments with different values of the α parameter in the combined loss function, specifically, we set it to 0.5 and 0.75.

We use 8 GPUs, each with 32G VRAM, for all experiments. We set the batch size to 8 and the accumulated gradient batches to 32. We follow the hyper-parameters settings from Josifoski et al. (2022) and set the learning rate to $3e^{-5}$, weight decay to 0.01, and warmup steps to 5K.[16] We train for up to 30 epochs with early stopping (patience 10), validate twice per epoch, and take the last checkpoint for evaluation. Training one epoch takes approximately 1.5 hours for BART and 2 hours for mBART.

5.5   Results and Analysis

This section presents the main results of (m)WEBIE and compare different training strategies.

---

[14] https://huggingface.co/facebook/bart-large
[15] https://huggingface.co/facebook/mbart-large-50
[16] For $\text{BART}_{\text{PLM}}(\text{W})$ we find it is necessary to use a lower learning rate $5e^{-6}$ for more stable training.

Model	WEBIE (ALL TEST)				WEBIE (ANNO. TEST)				REBEL			WIKI-NRE		
	P	R	F1	Acc-N	P	R	F1	Acc-N	P	R	F1	P	R	F1
UNCONSTRAINED														
BART_RAND (R)	10.83	16.00	12.92	0.00	10.70	13.26	11.84	0.00	64.34	67.90	66.07	15.83	52.09	24.28
BART_PLM (R)	17.58	34.20	23.23	2.28	17.95	30.02	22.47	1.97	63.83	76.66	69.66	18.34	65.04	28.62
BART_RAND (W)	55.06	54.90	54.98	89.67	51.64	44.46	47.78	94.74	22.45	20.42	21.39	10.95	31.49	16.25
BART_PLM (W)	54.81	70.29	61.59	87.59	53.40	62.36	57.53	93.58	28.05	37.28	32.01	15.55	60.45	24.73
BART_RAND (R+W)	51.34	61.22	55.85	86.80	49.64	51.62	50.61	93.15	64.38	69.57	66.87	17.68	65.96	27.89
BART_PLM (R+W)	53.04	75.29	62.23	76.66	53.18	68.41	59.84	82.96	63.49	75.30	68.89	18.93	73.52	30.11
CONSTRAINT TRIE														
BART_RAND (R)	11.93	18.91	14.63	0.00	11.82	15.63	13.46	0.00	66.89	70.37	68.58	27.61	66.73	39.06
BART_PLM (R)	15.24	39.30	21.96	0.00	15.98	34.92	21.93	0.00	66.28	76.78	71.14	25.39	77.45	38.24
BART_RAND (W)	55.47	57.25	56.35	90.07	52.95	46.60	49.57	95.04	27.47	23.13	25.12	18.98	43.75	26.48
BART_PLM (W)	57.92	74.19	64.91	87.99	57.00	65.91	61.13	94.18	35.81	43.00	39.08	24.30	78.01	37.06
BART_RAND (R+W)	52.79	64.15	57.92	87.45	51.89	54.28	53.06	93.71	66.87	72.24	69.45	29.02	82.35	42.91
BART_PLM ((R+W)	54.63	78.43	64.40	76.43	55.22	71.25	62.22	82.59	66.42	78.29	71.87	29.25	86.38	43.70

TABLE 5.2: Experiment results for beam search with and without constraint Trie. P and R refer to precision and recall, respectively, and Acc-N shows the accuracy of the negative examples. BART_RAND corresponds to models with BART configuration but randomly initialised weights. BART_PLM are models with pretrained weights from M. Lewis et al. (2020). (R), (W), (R+W) refer to models trained on REBEL, WEBIE, and both datasets, respectively. For WEBIE we show the overall performance and the accuracy on negative samples. Results in blue shades are zero-shot performance.

5.5.1 Main Results

Table 5.2 shows our benchmarking results on WEBIE. We report results with the constraint Trie in decoding since it overall achieves better results. Contrary to the findings from Josifoski et al. (2022), we find that BART models with pre-trained weights are better than randomly initialised weights. Constraint Trie decoding benefits REBEL, WikiNRE, and the recall performance of WEBIE, but may compromise the precision since the models are also trained to handle negative examples.

Models trained on both REBEL and WEBIE (R+W) obtain overall better F1 scores on the two datasets compared to models trained on each dataset separately. Similar performance can also be observed in the zero-shot performance on WikiNRE. Models trained solely on the REBEL dataset (Wikipedia-domain) show poor generalisability on WEBIE[17] and always generate false positives thus resulting in 0% accuracy for negative instances in WEBIE. This indicates that Wikipedia-domain data is not adequate for training robust models for the web, and the absence of negative examples in these datasets leads to a prominent issue of hallucination when applied to the web.

---

[17] For positive examples it only achieves 20 F1 points.

LANGUAGE	UNCONSTRAINED					CONSTRAINT TRIE				
	P	*R*	*F1*	*Empty-Pos%*	*Acc-N*	*P*	*R*	*F1*	*Empty-Pos%*	*Acc-N*
ENGLISH	57.72	61.26	59.43	2.48	95.69	60.29	64.29	62.22	2.63	96.11
FRENCH	43.27	36.13	39.38	11.89	96.19	46.52	40.26	43.16	12.63	96.64
SPANISH	41.93	34.63	37.93	12.34	96.74	45.13	38.89	41.78	12.80	96.97
PORTUGUESE	41.17	32.37	36.24	14.07	96.91	44.15	36.61	40.02	14.82	97.22
HINDI	4.28	1.62	2.35	67.38	98.64	4.23	1.67	2.40	67.55	98.64

TABLE 5.3: Performance on mWEBIE with mBART. Results for non-English are zero-shot. Empty-Pos(itive)% shows *false* negatives percentage, revealing zero-shot performance has a high rate of empty results for positive examples.

BART$_{\text{PLM}}$ (R+W) also achieves a new state-of-the-art F1 score of 71.87 on REBEL, surpassing the performance of 68.93 from GenIE (Josifoski et al., 2022) and 70.74 from KnowGL (Rossiello et al., 2023), the latter of which trains with additional information including entity type. The results demonstrate the benefit of WEBIE, which contributes to the generalisability of the models.

5.5.2   Cross-lingual Transfer with mBART

We train mBART on the training set of WEBIE and evaluate the zero-shot cross-lingual transfer on mWEBIE. Similar to prior experiments, results in Table 5.3 show that constraint Trie decoding obtains higher performance than standard decoding.[18]

For English, mBART achieves higher overall performance than BART$_{\text{PLM}}$ (see Table 5.2). The zero-shot results reveal that Hindi has a significant decline in performance compared to the other three non-English languages, French, Spanish, and Portuguese. As these three languages utilise the Latin script as in English, which may result in an overlap of entity surface forms. In contrast, the transfer is more difficult for Hindi as it employs a different writing system. Manual analysis indicates that mBART tends to produce a high rate of false negatives in Hindi examples, where the correct extraction mostly occurs when the entities in the sentences share similar surface forms with the English counterparts.

5.5.3   Results with Additional EL Training

Table 5.4 shows the results of training with Entity-Linking as an auxiliary task. For REBEL, the best results are achieved with the 2LM-HEADS approach, where the α

---

[18]We report results using EN as the source language token for mBART, as it produces better performance compared to the actual source language token. See more details in section B.1.

	REBEL						WebIE (Anno.)					
MODEL	UNCONSTRAINED			CONSTRAINT TRIE			UNCONSTRAINED			CONSTRAINT TRIE		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BART$_{\text{RAND}}$	64.34	67.90	66.07	66.89	70.37	68.58	51.64	44.46	47.78	52.95	46.60	49.57
ENTITY-PROMPTS	63.30	63.04	63.17	67.91	67.54	67.72	49.64	51.62	**50.61**	51.90	54.28	**53.06**
ARTIFICIAL-PROMPT	64.23	68.23	66.17	66.41	70.72	68.50	52.33	46.21	49.08	53.86	48.18	50.86
2LM-HEADS	65.16	68.70	**66.88**	67.05	70.88	**68.91**	49.13	47.67	48.39	51.07	49.59	50.32

TABLE 5.4: Comparison of various training with entity linking as an auxiliary task, and beam search with and without constraint Trie decoding. WEBIE results are on the annotated test set. All models use BART configuration with randomly initialised weights. We show in bold the best result among the training objectives.

parameter is set to 0.75. For WEBIE with negative examples, all EL training models achieve better F1 performance than BART$_{\text{RAND}}$, with ENTITY-PROMPT particularly resulting in better recall. This shows the benefit of joint training with EL to improve the faithfulness of web domain data. ARTIFICIAL-PROMPT achieves the best precision in WEBIE but does not show significant differences in performance compared to BART$_{\text{RAND}}$. We also note that the performance on negative examples does not show significant variations among the different training approaches. Nevertheless, all three approaches provide better interpretability, i.e., providing the information of the mention spans in the text that contributes to the IE prediction.

ENTITY-PROMPT and ARTIFICIAL-PROMPT do not require additional architectural adaptation over the standard model. ENTITY-PROMPT also does not introduce training overhead, whereas the other two models may require twice the training time. 2LM-HEADS offers the flexibility of adapting the weighted combination of the main task and the auxiliary task by adjusting α in the joint loss formula, which allows more emphasis on the main target. Regarding extending to more tasks, the ARTIFICIAL-PROMPT approach can be easily extended by adding additional artificial tokens as task identifiers, whereas 2LM-HEADS approach would need to add more LM-heads to include more tasks.

5.6 Conclusion and Future Work

We present (m)WEBIE, the first large-scale, entity-linked closed IE dataset on the web. A subset of the dataset is further annotated by humans and translated into four

other languages, French, Spanish, Portuguese, and Hindi, via crowdsourcing. The main contribution lies in the development of faithful and robust extraction pipelines, coupled with effective training methods, for structured knowledge on the web.

The high approval rate of the triples obtained from the automatic IE pipeline, as confirmed by Human annotation, underscores the efficacy of extracting fact triples through Entity linking, Distance Supervision, and NLI entailment filtering. However, given the recent surge in powerful Language Model (LLM) capabilities, it is intriguing to compare the quality of IE datasets obtained using LLM prompting or via few-shot in-context learning (Y. Ma et al., 2023). While the automatic pipeline excels in efficiency, LLMs offer greater flexibility since they don't necessitate restriction to a single knowledge base as required by distance supervision. Nevertheless, both distance supervision and LLM-generated triples are prone to false positives, hence the post-processing such as the NLI filtering remains essential in both scenarios.

We benchmark WEBIE with generative models and compare the models trained on WEBIE and REBEL (Wikipedia domain). Our results show that models trained on WEBIE have competitive zero-shot performance when applied to REBEL and WikiNRE, whereas models trained only on REBEL have 0% accuracy on the negative examples in WEBIE. This highlights the importance of including negative examples for training more robust models and reducing hallucination in generative IE on the web. Models trained on both REBEL and WEBIE achieve the best performance on both datasets, as well as zero-shot results on WikiNRE, positioning WEBIE as a complementary dataset to existing Wikipedia-centric datasets.

Our exploration of approaches integrating Entity Linking as an auxiliary task reveals that the addition of a task-specific LM head achieves the overall best performance for REBEL. Notably, the ENTITY-PROMPT approach demonstrates significant improvement on WEBIE, particularly enhancing recall.

While our primary benchmarking involves transformer-based encoder-decoder models on WEBIE, future work could also explore pipeline frameworks and larger language models for few-shot performance. This chapter underscores the importance of developing a faithful and robust pipeline for extracting structured knowledge on the web, which can then be incorporated into many other knowledge-intensive applications.

5.7   Limitations

We identify several limitations in this work:

- **False Negatives**: Our current automatic triple extraction pipeline is built using the DS approach followed by filtering using an NLI model. However, Wikidata is not complete (Q. Tan et al., 2022). While some triples may not be completely available in WEBIE, we expect models trained on this dataset can still discover new triples that do not exist in Wikidata.

- **Limited Relations in Annotation**: The human annotation is only conducted on the most frequent 200 relations.

- **Limited Languages in mWEBIE**: As discussed in subsection 5.2.5 and section B.3, the languages in mWEBIE are limited to official languages from geographical regions where there is a reasonable amount of MTurk workers to accept the job.

  An alternative solution would be to use professional translators, especially for low-resource languages.

- **Fixed Dataset**:

  Facts might change in the world (and Wikidata). This can lead to a degraded real-world performance if a system relies exclusively on WebIE for evaluation when the dataset is not updated accordingly.

Chapter 6

Grounded Answer and Explanation in Knowledge-Intensive VQA

In the upcoming two chapters, we shift our focus to a broader format of knowledge beyond structured knowledge. This chapter, in particular, centres around knowledge-intensive Visual Question Answering. We explore better utilisation of the parametric knowledge embedded in pre-trained multimodal language models through self-explanation.

The main content is an extended version of the paper "Towards a Unified Model for Generating Answers and Explanations in Visual Question Answering" (Whitehouse, Weyde, and Madhyastha, 2023) published in *Findings of the Association for Computational Linguistics: EACL 2023*.

6.1   Background and Introduction

The focus of this chapter is on knowledge-intensive Visual Question Answering (VQA) tasks. Contemporary models for Visual Question Answering (VQA) and Visual Commonsense Reasoning are typically trained discriminatively to select the best answers from multiple-choice questions or to classify single-word answers from a predetermined vocabulary (Anderson et al., 2018). However, such settings often have limitations, such as encouraging models to find superficial correlations (Ye and Kovashka, 2021) or penalising model performance, even when the answers are plausible. For example, synonyms, multi-word expressions, and morphological variations are often not considered correct answers.

FIGURE 6.1: Illustration of UMAE: we train a multimodal encoder-decoder model on the mix of VQA tasks for jointly optimising answer and explanation, where we distinguish the training instances and target output with artificial prompt tokens (e.g., <#AOKA#>). The top and bottom examples are from A-OKVQA and VCR, respectively.

Moreover, most current explanation generation models are trained independently of the QA model, and explanations are usually generated after the QA model has provided an answer. Consequently, these explanation models lack access to the process that generated the answer, limiting the grounding of the explanation to the answer text.

We posit that a unified model that simultaneously performs answer prediction and explanation generation is a more effective and consistent approach for VQA. Generative models, such as GPT-3 (Brown et al., 2020a), T5 (Raffel et al., 2020b), or OFA (P. Wang et al., 2022), have demonstrated success in rapidly adapting to downstream tasks and generating high-quality open-ended text, making them suitable candidates for this unified approach.

To address this, we propose a multitask learning approach for transformer-based multimodal encoder-decoder models, creating a Unified Model for Answer and Explanation generation (UMAE). In addition to the prevailing trend of separate answer prediction and explanation generation based on the answers, our approach adds the capability of jointly generating answers and explanations. Inspired by the success of artificial prompt tokens in Neural Machine Translation (NMT) (M. Johnson et al., 2017), we extend and demonstrate the efficacy of the artificial prompt-based method for VQA in a multitask setup. Specifically, we augment training instances with artificial prompt tokens, enabling the model to distinguish different tasks while learning shared semantic features.

After the model is trained, we propose to determine the best option among the multiple choices by directly ranking the options based on the model's perplexity against each one. We find this approach to be more optimal compared to mapping the generated answer to the closest word embedding from the options (D. Schwenk et al., 2022) or using sentence embedding (concatenating the question and answer) with BERTScore (T. Zhang et al., 2020). Our method also addresses the surface form penalisation mentioned earlier (synonyms, multi-word expressions, morphological variations, etc.) that involves exact match metrics (Y. Goyal et al., 2017).

Experiments on a combination of three knowledge-intensive VQA datasets, OK-VQA (Marino et al., 2019), A-OKVQA (D. Schwenk et al., 2022), and VCR (Zellers et al., 2019a), show that UMAE models achieve a new state-of-the-art answer accuracy on A-OKVQA, a new state-of-the-art explanation score on VCR, and competitive out-of-domain performance on VQA-X (Park et al., 2018). UMAE supports the generation of the answer to a question, the explanation for a given question and answer, and both together jointly, making the model efficient and flexible. An illustration of the training setup is shown in Figure 6.1. To the best of our knowledge, our proposal is the first to unit grounded answer and explanation generation for VQA. We specifically focus on knowledge-intensive VQA tasks that are designed to require accessing a knowledge base for answers. By joint training for answer and explanation generation, we hypothesise that the model can more effectively leverage parametric knowledge, potentially enhancing performance even without explicit links to external knowledge sources.

In summary, our main contributions are as follows:

- The UMAE framework where answers and explanations can be generated by a single unified model (subsection 6.3.1).

- A simple and efficient training approach that uses multitask learning with artificial prompts and demonstrates its ability to generalise across domains (section 6.4).

- A method to map generated answers to Multiple-Choice options via evaluating the perplexity of the generation (subsection 6.3.2).

- New state-of-the-art results by UMAE particularly for explanation generation, as well as promising out-of-domain performance (section 6.5).

6.2 Related Work

We introduce the related work to this chapter in the following three aspects.

6.2.1 Multimodal Transformer-based Models

As detailed in Chapter 2, multimodal transformer-based encoder-decoder models achieve state-of-the-art performance on various vision-language tasks (Y.-C. Chen et al., 2020; Xiujun Li et al., 2020; Cho et al., 2021; P. Zhang et al., 2021; Z. Wang et al., 2022). They showcase the possibility of capturing richer multimodal semantic coherence than discriminatively trained models and are further capable of generating self-explanations. Pre-trained on multitask settings with natural language instructions, e.g., *"what does the region describe?"*, models like OFA (P. Wang et al., 2022) are claimed to have the capability to transfer to unseen tasks and domains via similar instructions. However, contrary to these claims, we observe that pre-trained OFA is incapable of generating valid explanations through simple natural language instructions (section 6.5).

6.2.2 Artificial Prompt Tokens

Artificial prompt tokens, introduced by M. Johnson et al. (2017), have primarily found application in Neural Machine Translation (NMT). In their work on Google's multilingual NMT, tokens like 2es are added at the beginning of sentences to indicate the target language for translation is Spanish. By using different prompt tokens, they jointly train a multilingual NMT model on various language pairs, allowing the model to learn shared semantics among different instances. This approach proves beneficial for low-resource languages and is effective for zero-shot performance on previously unseen language pairs (M. Johnson et al., 2017).

Building on the success observed in NMT, our work exploits a similar approach with artificial prompts for answer and explanation generation in VQA with a united

model. This enables the model to learn shared features among tasks and datasets in various domains.

6.2.3 Explanation Generation for VQA

There has been a growing interest in incorporating explanations into Visual Question Answering tasks, including Visual Commonsense Reasoning. Various datasets have been developed where explanations are provided, e.g., VCR (Zellers et al., 2019a), which provides explanations as candidate options in multiple-choice questions, and A-OKVQA (D. Schwenk et al., 2022), which includes multiple explanations for the answer to each question.

Existing VQA datasets have also been extended or corrected with provided textual explanations, for example, VQA-X (Park et al., 2018), CLEVR-X (Salewski et al., 2022), and e-SNLI-VE (Kayser et al., 2021) are developed from VQAv2 (Y. Goyal et al., 2017), CLEVR (J. Johnson et al., 2017), and SNLI-VE (Xie et al., 2019), respectively.

For modelling explanation generation, most recent approaches use separate models to predict answers and generate explanations (Dua, Kancheti, and Balasubramanian, 2021). J. Wu and Mooney (2019) introduce the concept of Faithful Multimodal Explanations (FME) for VQA, wherein textual explanations are linked to relevant image regions attended to by the underlying VQA system. Dua, Kancheti, and Balasubramanian (2021) take a two-step approach, training separate modules for answer generation and explanation generation, with the latter based on previously computed answers. Kayser et al. (2021) develop a model called e-UG, which combines UNITER (Y.-C. Chen et al., 2020) for processing multimodal input and GPT-2 (Radford et al., 2019) for generation. In contrast, we propose using a single united model for more grounded answer and explanation generation.

6.3 Methodology

We introduce the methodology in the following two aspects.

6.3.1   Multitask Learning with Artificial Prompt Tokens

We formulate three generation settings: Q→A: answer prediction; QA→E: explanation generation conditioned on the answer; and Q→AE: *joint* answer and explanation generation for a given question. We hypothesise that by training the model to generate both the answer and its explanation *simultaneously*, the result answer and explanation will be more grounded and consistent.

We use a pre-trained multimodal encoder-decoder transformer as our base model (here we build on the openly released version of OFA as a strong baseline), and fine-tune the model on a mix of VQA datasets from different domains.

Different from OFA, for each image in the VQA datasets, we first extract objects and attributes using a bottom-up top-down attention-based model, which is crucial for open-domain VQA tasks (Anderson et al., 2018). We then add artificial prompt tokens at the beginning of the textual input to signal the generation task (answer, explanation, or both) and the dataset.[1] For Q→AE, we concatenate answers and explanations with a separator in between. Finally, we mix all training instances, each consisting of an image (processed in patches), objects and attributes, and textual input with artificial prompts.

6.3.2   Perplexity as Multiple Choice Metric

To map the generated output to Multiple-Choice options, in previous work the predictions are loosely matched with options or gold answers using embedding-based methods, such as GloVe embedding similarity (D. Schwenk et al., 2022). In contrast to these approaches, we propose to evaluate each option as a *text generation* task, by feeding the model the information that was used to generate the answer as a prompt and calculating the likelihood of each option being generated. Formally, given an option $Y = (y_1, y_2, ..., y_t)$ with t tokens, we calculate the probability of each token y_i being generated by feeding the image, objects, and question, as well as the first $i - 1$ tokens from Y to the model p_θ. The perplexity is then calculated with:

---

[1]Artificial prompt tokens are added as special tokens to the tokeniser to avoid bias in the pre-trained embeddings. However, we note that these tokens may be biased regarding their association with specific tasks after training, which is an intended effect.

$\mathrm{PPL}(Y) = \exp\left\{-\frac{1}{t}\sum_i^t \log p_\theta\left(y_i|y_{<i}\right)\right\}$, which reflects the probability of option Y being generated by the model. Finally, the option with the lowest perplexity is chosen as the answer.

We also compare the performance of our approach, using perplexity as the metric, with GloVe embedding similarity for A-OKVQA (see Table 6.1).

6.4 Experimental Setup

We primarily evaluate our proposed UMAE approach using pre-trained OFA[2] as the base model on three knowledge-intensive VQA datasets: OK-VQA, A-OKVQA and VCR. The datasets are introduced below.

6.4.1 Datasets

OK-VQA (Marino et al., 2019) is a knowledge-based VQA dataset that requires outside knowledge beyond the images to answer the questions. It has train and test splits of size 9,009 and 5,046. Each question is provided answers by five annotators. To use the VQA (Antol et al., 2015) metric, each annotated answer is then repeated twice to form a gold answer set with 10 answers. Since no explanation is provided, we only train `Q`→`A` task on OK-VQA.

A-OKVQA (D. Schwenk et al., 2022) is currently a large-scale knowledge-based VQA dataset split into 17.1K, 1.1K, and 6.7K for train, validation, and test, respectively. The questions cover four knowledge types: visual, commonsense, knowledge bases, and physical. For each question, it provides both multiple-choice answers and 10 free-form answers (annotated by 10 different people), as well as three explanations. Images in both OK-VQA and A-OKVQA are from MSCOCO (T.-Y. Lin et al., 2014), and answers in both datasets are in single words or short phrases.

VCR (Zellers et al., 2019a) is a large multiple-choice dataset for Visual Commonsense Reasoning. The train, validation, and test splits have 191.6k, 21.3k, and 26.5k instances, respectively. Each question has four answer options in sentences, and the

---

[2]https://github.com/OFA-Sys/OFA

correct answer is further provided with four explanation options. Images in VCR are from movie clips (Rohrbach et al., 2017). Bounding boxes of entities are provided associated with mentions such as Person1 in questions, answers and explanations. We follow Zellers et al. (2021) and draw coloured highlights around the referenced entity on the images, where entity names and the coloured highlights are consistent in the entire dataset, expecting the model to learn the association between the coloured bounding box and the entity.

VQA-X (Park et al., 2018) contains a subset from the VQAv2 (Y. Goyal et al., 2017) dataset and further provides three explanations for each question. The image-question pairs are split into train, validation, and test with 29.5k, 1.5k, and 2k instances, respectively. We only use the original test set to evaluate the zero-shot performance of the trained models.

6.4.2   Training Setup and Hyper-parameters

We begin with the pre-trained weights from the original OFA-large,[3] which is trained on vision-only tasks including Image Classification, language-only tasks including Sentence Classification and text Summarisation, as well as various vision-language tasks including Image Captioning, Visual Question Answering and Visual Entailment (P. Wang et al., 2022).

We split the original train set into train and validation sets (95/5 split) for all three datasets. Since the test set is not publicly available for A-OKVQA and VCR, we use the original validation set for experimental analyses. We prepare training instances as introduced in subsection 6.3.1. Specifically, we add <#OKA#> for OK-VQA (only answers are available), <#A#>, <#E#>, <#AE#> for VCR, and <#AOKA#>, <#AOKE#>, <#AOKAE#> for A-OKVQA. Additionally, for VCR, we draw coloured highlighted boxes around the referenced entity on the images as described. To account for the imbalance in size among the datasets, we up-sample instances in OK-VQA and A-OKVQA, and shuffle all instances to train a model denoted as UMAE_{ALL}.

---

[3]https://github.com/OFA-Sys/OFA

For ablation studies, we fine-tune OFA for separate answer prediction ($OFA_{Q->A}$) and explanation generation conditioned on answers ($OFA_{QA->E}$). To better understand the impact of mixing datasets from different domains, we also train models $UMAE_{A-OKVQA}$ and $UMAE_{VCR}$, focusing on all three answer and explanation generation tasks but only using data from a single dataset: either with A-OKVQA or with VCR. Adam is used as the optimiser and cross-entropy is the loss function.

We set the learning rate to $10e^{-5}$, the warm-up ratio to 0.4, and the patch image size to 480. We shuffle all the training examples and use batch size 16. Due to the large size of VCR, we train for 30 epochs on models involving VCR ($OFA_{Q->A}$ for VCR, $UMAE_{VCR}$ and $UMAE_{ALL}$), and up to 100 epochs for other models. We report the empirical performance with checkpoints that perform best on the validation set (the 5% split from the original train set). For A-OKVQA, we additionally report the answer accuracy on the original test set.

6.4.3 NLG Evaluation Metrics

We use beam search for generating answers and additionally experiment with different decoding methods including top-k sampling, Nucleus sampling (Holtzman et al., 2020), and Typical sampling (Meister et al., 2023), for generating explanations. We evaluate answer accuracy as well as explanation quality with automatic NLG metrics and e-ViL scores, following Kayser et al. (2021) for better comparison. Specifically, e-ViL scores consist of S_T (task, i.e., answer accuracy), S_E (explanation score), and overall S_0 (product of S_T and S_E), where S_E is defined by Kayser et al. (2021) as the harmonic mean of NGRAMScore (the harmonic mean of n-gram scores ROUGE-L (C.-Y. Lin and Och, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam, Lawrence Zitnick, and Parikh, 2015), and SPICE (Anderson et al., 2016)) and additionally the BERTScore (T. Zhang et al., 2020), a learned similarity metric over contextual representations of sentences.

We elaborate and discuss different aspects of these NLG metrics. Each of the n-gram scores targets distinct aspects of the output. ROUGE-L focuses on assessing the

overlap of contiguous sequences of words between the generated text and the reference text by measuring the precision, recall, and F1 score of the longest common subsequences between the generated and reference texts. METEOR emphasises semantic similarity and incorporates stemming, synonymy, and word order information to compute a score that better reflects the semantic equivalence between the generated and reference texts. CIDEr assesses the consensus between human judgements and machine-generated texts by calculating a score based on the similarity of n-grams between the generated text and multiple reference texts provided by human annotators. SPICE focuses on the precision and recall of generated linguistic structures and evaluates the structural similarity between the generated and reference texts by analysing syntactic and semantic components. On the other hand, BERTScore is a trained metric that leverages contextual embeddings obtained from pre-trained transformer models like BERT to measure the similarity between sentences at a more granular level.

To summarise, higher ROUGE-L scores suggest higher degrees of lexical overlap between the generated and reference texts; higher METEOR scores indicate greater semantic similarity between the generated and reference texts; higher CIDEr scores imply better alignment with human judgements and consensus among multiple reference texts; higher SPICE scores reflect greater precision and recall of generated linguistic structures; and higher BERTScores signify increased similarity between sentences at a contextual level. Although these metrics target different aspects of the generated text, experimental results reveal a consistent trend across various metrics: if a generation scores higher in one metric, it often scores higher in others as well (refer to subsection 6.5.2).

However, it is worth noting that none of these metrics directly measure the human interpretability of the generated explanations. N-gram metrics overlook semantic meanings, coherence, and relevance of the generated text, while BERTScore is also biased with the lengths of the generation greatly differing. More discussions on the limitation of the NLG metric are included in subsection 6.5.2 and section 6.7.

| MODEL | OK-VQA | A-OKVQA | | | | | VCR | |
| | *direct answer* | *multiple choice* | | | *direct answer* | | *multiple choice* | *BERTScore* |
	TEST	VAL (*ppl*)	VAL (*GloVe*)	TEST	VAL	TEST	VAL (*ppl*)	VAL
OFA*	40.40	24.54	56.19	47.40	48.09	39.77	33.55	64.55
OFA$_{Q->A}$	49.93	74.32	65.30	61.71	63.00	53.91	54.89	83.85
UMAE$_{ALL}$	**51.77**	**74.59**	**65.67**	**63.26**	**63.29**	**56.14**	**56.66**	**85.97**
PRIOR-BEST	54.41	–	60.30	53.70	48.60	40.70	(77.10)†	–

TABLE 6.1: Performance of models for answer generation. Better results are in bold. OFA* refers to the pre-trained OFA. Prior-best results for the three datasets are from Gui et al. (2022), D. Schwenk et al. (2022), Yanan Wang et al. (2023), respectively. † is from a discriminative model and thus not comparable (see Ye and Kovashka, 2021).

6.5 Main Results

In this section, we show the results for generated answers and explanations, compare models and analyse errors.

6.5.1 Answer Accuracy

Table 6.1 presents our observations for answer accuracy on `Q->A` task over the three datasets. We also evaluate VCR answers using BERTScore as the answers for VCR are usually sentences. We observe that UMAE$_{ALL}$ outperforms OFA$_{Q->A}$ on all datasets, improves the prior state-of-the-art on A-OKVQA by 10∼15%, and achieves competitive results on OK-VQA. For models that are fine-tuned on A-OKVQA, we also see a salient improvement (+9%) with the proposed mapping of options by perplexity in Multiple-Choice, instead of GloVe embeddings similarity.[4]

QUESTION	OBJECTS	IMAGES	ACCURACY
✓	✓	original	50.39
✓	✗	✗	39.16
✓	✗	random	33.48
✓	✓	✗	33.28

TABLE 6.2: Ablation on the modality dependency for answer accuracy of A-OKVQA.

---

[4]Preliminary experiments with NLG metrics (BERTScore and BLEU) for selecting the options given generation were sub-optimal.

DATASET	MODEL	e-ViL SCORES			N-GRAM SCORES					LEARNT SC.
		S_O	S_T	S_E	BLEU4	R-L	MET.	CIDEr	SPICE	BERTSCORE
A-OKVQA	OFA*	4.44	56.19	7.90	0.30	4.45	3.26	4.82	4.62	68.64
	OFA$_{Q->A}$+OFA$_{QA->E}$	35.82	74.32	48.29	22.18	48.51	23.56	86.76	22.46	85.96
	UMAE$_{A-OKVQA}$	37.10	73.97	50.15	**27.61**	52.23	24.06	**104.39**	22.88	87.86
	UMAE$_{ALL}$	**37.91**	**74.59**	**50.82**	27.35	**52.56**	**24.83**	101.09	**23.33**	**88.21**
VCR	e-UG	19.30	**69.80**	27.60	4.30	22.50	11.80	32.70	12.60	79.00
	UMAE$_{VCR}$	22.57	**56.68**	39.82	12.25	28.87	16.67	**48.14**	27.36	81.77
	UMAE$_{ALL}$	**22.82**	56.66	**40.27**	**13.44**	**29.53**	**17.54**	47.33	**26.45**	**81.91**
VQA-X	e-UG	36.50	80.50	45.40	23.20	45.70	22.10	74.10	20.10	87.00
	UMAE$_{ALL}$	31.58	77.65	40.67	14.63	35.12	20.29	50.35	19.13	85.40

TABLE 6.3: Explanation Scores. R-L, MET. stand for ROUGE-L, METEOR, respectively. OFA* is the pre-trained OFA, showing the transferability of OFA for generating explanations with natural language instructions. Results with e-UG are from Kayser et al. (2021). We show the best results of A-OKVQA and VCR in bold. The last row in blue shade shows *out-of-domain* performance.

We conducted several ablation studies to investigate the dependency of object features and images on the performance of our model UMAE$_{ALL}$ for answer accuracy of A-OKVQA, where we removed images, replaced them with random images, and removed extracted attributes and features. Results in Table 6.2 show that the visual encoder is crucial for performance and that visual objects alone are not sufficient for answer prediction. Using a random image would introduce noise and therefore performs worse than not including the image at all. We did not test removing the question because we believe the model needs the questions to be able to provide answers.

6.5.2 Explanation Evaluation

Table 6.3 shows e-ViL sores and individual NLG evaluation results for explanations using automatic NLG metrics.[5] Following the same setup as in Kayser et al. (2021), an explanation is evaluated only if the answer predicted by the system is correct.[6] We observe that pre-trained OFA with natural language prompts, e.g., *"what is the*

---

[5]Nucleus sampling shows the best results and is reported. Detailed scores with different decoding methods are shown in section C.1.

[6]A limitation of evaluating all explanations is that explanations of wrong answers may get high scores with n-gram metrics, even though they are justifying wrong answers and should be penalised.
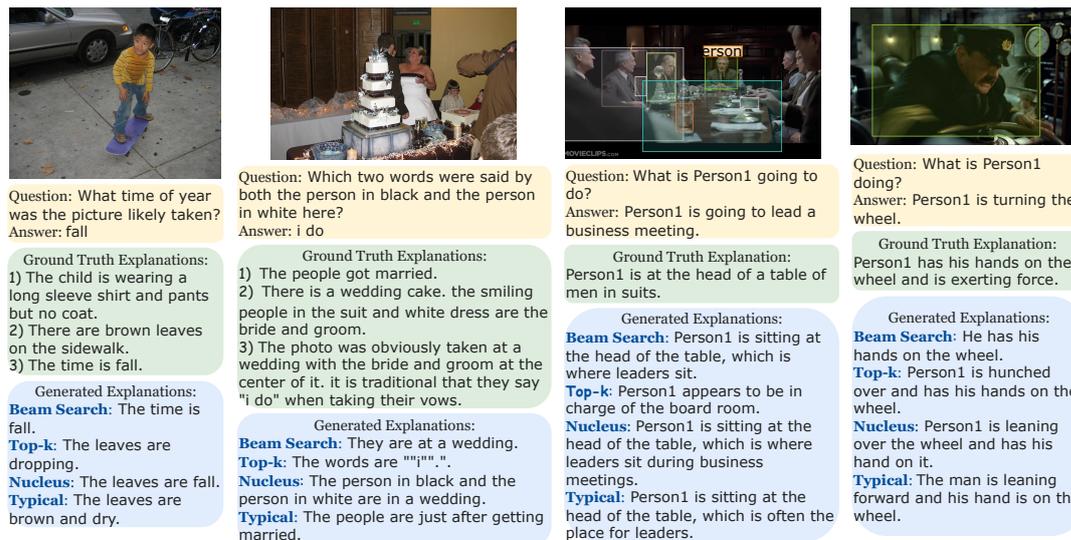
FIGURE 6.2: Examples of generated explanations from MIX model with different decoding strategies. Two examples on the left are from A-OKVQA and the other two on the right are from VCR.

explanation for the answer?" or *"this is because"* performs poorly, as most generated explanations are words (*"yes/no"*) or short-phrases.[7] We compare UMAE models (on all and individual datasets) with prior best results from e-UG (see section 6.2), and standard separated trained baselines (OFA$_{Q->A}$+OFA$_{QA->E}$). UMAE$_{ALL}$ achieves better results across all datasets, showing the advantage of mixing tasks and datasets in different domains. For out-of-domain evaluation on VQA-X, UMAE$_{ALL}$ also shows mostly competitive results. Examples of explanation generation are shown in Figure 6.2 as well as in section C.2 in Appendix C.

Since e-ViL only evaluates an explanation if a model generates the correct answer, the subset of explanations evaluated varies by model. To *fairly* compare explanations on the same subset, we propose only using the subset of samples where all models provide correct answers for explanation prediction. Table 6.4 shows the results on A-OKVQA with such a subset of 770 candidates, where UMAE$_{ALL}$ shows an even higher explanation score. This highlights that UMAE$_{ALL}$ generates explanations that overlap significantly better with gold explanations.

---

[7]BERTScore in not representative of the validity of outputs from OFA*. We refer the reader to an exposition of the problems associated with NLG metrics in Caglayan, Madhyastha, and Specia (2020).

MODEL	S_E	BLEU4	R-L	MET.	CIDEr	SPICE	BERTSCORE
OFA$_{Q->A}$+OFA$_{QA->E}$	42.4	20.0	44.2	19.3	66.7	19.1	85.1
UMAE$_{A-OKVQA}$	45.8	23.6	47.9	21.7	78.0	20.5	86.9
UMAE$_{ALL}$	**46.8**	**24.9**	**49.5**	**22.3**	**84.1**	**20.8**	**87.3**

TABLE 6.4: Explanation scores on the same subset of A-OKVQA.

6.5.3   Joint Answer and Explanation Generation

We further present the results of the proposed Q→AE task where answers and explanations are jointly generated. We parse the generated sequence to the answer and the explanation and use the same sets of metrics as the separate generation for evaluation. Results for answers in Table 6.5 and explanations in Table 6.6. For answers, since the perplexity metric does not directly compare the generation, we show the Multiple-Choice accuracy using the Glove metric for A-OKVQA and BERTScore for VCR answer sentences.

TASK	A-OKVQA MULTIPLE-CHOICE	VCR BERTSCORE	VQA-X DIRECT ANSWER
Q->A	65.67	81.91	77.65
Q->AE	65.67	82.30	69.60

TABLE 6.5: Evaluation of answers generated given questions (Q->A) and jointly generated with explanations (Q->AE). The last column with a blue shadow indicates out-of-domain performance. For simplicity, we use the Glove metric for selecting answers for Multiple-Choice questions in A-OKVQA.

DATASET	S_E		NGRAMSCORE		BERTSCORE	
	QA->E	Q->AE	QA->E	Q->AE	QA->E	Q->AE
A-OKVQA	50.82	47.01	35.69	32.15	88.21	87.39
VCR	40.27	37.02	26.70	24.02	81.91	80.68
VQA-X	40.67	39.67	26.69	25.85	85.40	85.21

TABLE 6.6: Scores of explanations generated given answers (QA->E) and jointly generated with answers (Q->AE). The last row with a blue shadow indicates out-of-domain performance.

In summary, our experiments demonstrate that the UMAE model leads to improved answer and explanation generation and allows for the flexibility to generate different types of outputs, including answers, explanations, or both. We observe that

UMAE exhibits promising results in jointly generating both the answer and explanation.

6.6 Analysis and Discussion

In this section, we analyse the errors in answer and explanation generation, showcase the issues existing in the datasets, and discuss the limitations in the explanation evaluation metrics.

6.6.1 Error Analysis

To better understand the generated answers and errors, we randomly sample 50 errors in OK-VQA and A-OKVQA. Our analysis reveals the following main error types, where the first three are related to model performance:

- *Knowledge*: the implicit knowledge learned by the model is insufficient for answering some of the knowledge-intensive questions, such as questions asking *when* a certain sport was invented.

- *Visual*: the model fails to identify the visual attributes correctly, such as questions about *recognising object shape or material*.

- *Semantic disassociation*: the model misinterprets questions or fails to match the intended semantic meaning. For example, it may answer what *an object is* instead of a more complex question such as *what is commonly packed in it* (e.g., answering "suitcase" instead of "clothes").

- *Metric*: the evaluation metric may penalise some of the plausible answers, especially when searching for exact match answers (mostly due to the difference of singular/plural or phrases with/without space in between).

- *Dataset*: errors due to issues in the datasets themselves.

We discuss prominent issues in dataset quality briefly in Appendix C and further present the distribution of error types in Figure 6.3.
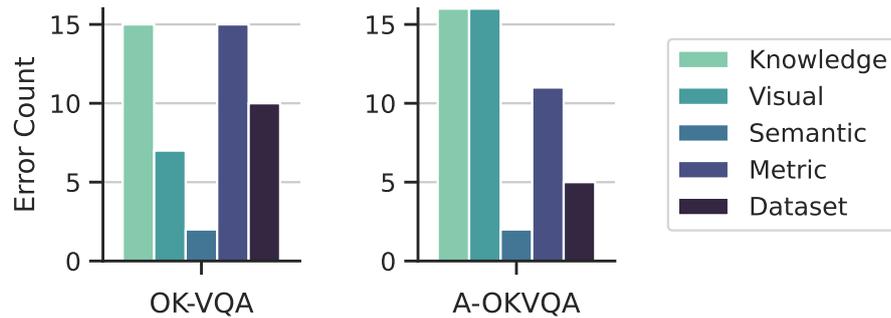
FIGURE 6.3: Distribution of error types in direct answers from 50 error samples in OK-VQA and A-OKVQA.

FIGURE 6.4: Example of misassociation between coloured highlights and entities. The model fails to associate *purple* box with `Person2` and *green* box with `Person3`.

We also observe misassociation between bounding boxes and entities in VCR. Many questions in VCR provide extra context assisting the association, for example, questions like *"Why are the eyes of Person2 closed?"* helps the model to identify `Person2` as "the person with closed eyes" in the image, but questions that specifically need the correct association without context can become problematic. Figure 6.4 shows an example revealing that the model struggles with the association. This may be related to the limitation of patch image processing used in OFA. The assumption that the model would learn the association via the coloured highlights in the images is not always valid.

6.6.2 Dataset Quality

We observe the following issues in the existing datasets: (i) wrong answers, (ii) subjective or unanswerable questions, (iii) typos or unclear expressions, (iv) not requiring images or knowledge to answer the question as designed.

Unlike OK-VQA and A-OKVQA where the ground truth answers are provided by multiple people, in VCR instead, answers and explanations for a question are obtained from the same person who authored the question. This makes the answers or explanations contain a severe amount of subjectivity. We find that a significant number of the examples expect an understanding of the *movie plot* from which the image is extracted, rather than requiring *commonsense* reasoning. While human annotators have an implicit understanding of the movies, the dataset itself does not contain relevant contextual information. We provide specific examples of the issues in section C.3 in Appendix C.

	OK-VQA	A-OKVQA	
	DIRECT ANSWER	MULTIPLE-CHOICE	DIRECT ANSWER
BEST	80.94	80.74	66.20
AVERAGE	54.98	71.53	57.29
WORST	16.37	59.35	41.46

TABLE 6.7: Human performance on OK-VQA and A-OKVQA measured from the ground truth answers. For simplicity, we use the Glove metric for selecting answers for Multiple-Choice questions in A-OKVQA.

We further measure the best, average and worst human performance on OK-VQA and A-OKVQA by selecting the most common answer, a random answer, and the least common answer, respectively, from the 10 ground truth answers for each question. We calculate the performance using the VQA metric for direct answers, and the GloVe metric for multiple choices for simplicity. Note that we also remove the answer selected from the ground truth answers when measuring human performance. From the results in Table 6.7 we can see that the average performance on both datasets is relatively poor, which indicates the noise in the datasets. The quality of the datasets needs to be more carefully inspected so that the model performance evaluated on these datasets can be more meaningful.

6.6.3   Explanation Evaluation

Current NLG metrics predominantly evaluate the n-gram scores between generation and reference. Although human evaluation may be the ultimate criterion (Kayser et al., 2021), this does not scale. Humans' judgements on generated explanations are contextual, especially the images that are predominantly taken into consideration. However, none of the current widely used metrics considers visual information. Multimodal evaluation metrics (Madhyastha, J. Wang, and Specia, 2019; M. Jiang et al., 2019; Hessel et al., 2021) are potentially better approaches to obtain visually grounded measures for evaluation.

6.7   Conclusion and Future Work

In summary, our proposed Unified Model for Answer and Explanation generation (UMAE) leverages a multitask learning approach within a multimodal encoder-decoder framework, incorporating artificial prompt tokens to distinguish distinct tasks while learning shared semantics.

Evaluation of our approach on various VQA tasks shows that UMAE outperforms prior best models and separately trained baselines in both answer and explanation scores, where we also demonstrate the benefit of using perplexity as the metric for mapping generated answers to Multiple-Choice options.

Additionally, UMAE offers flexibility in output and can generate explanations for datasets without explanations for training, e.g., OK-VQA, while also improving answer quality. In-depth case studies and error analyses undertaken in this chapter reveal valuable insights for future enhancements, underscoring the importance of continuous improvement in dataset quality.

This chapter highlights the benefit of grounded answer and explanation generation towards both, showcasing better utilisation of the parametric knowledge stored in the parameters of the multimodal language models.

6.8   Limitations

We address the limitations of our work in the following two aspects:

- **Model Specificity**: Firstly, the experiments conducted with our proposed framework and fine-tuning approach are primarily on the OFA model. While we posit that our approach applies to any multimodal generative model, broader insights could be gained by experimenting with a more diverse set of models.

- **Evaluation Methodology**: Secondly, in terms of evaluating our proposed joint framework, assessing the generated explanation quality, especially differentiating between explanations generated jointly with answers and those conditioned on the answers, necessitates human judgement. Relying on human evaluation becomes essential for a nuanced assessment compared to the use of automatic NLG metrics.

Chapter 7

Knowledge Distillation via LLM-powered Data Augmentation

In the final chapter, we explore the utilisation of knowledge from the latest trend of powerful large language models, in complex and challenging multilingual commonsense reasoning tasks.

The main content of this chapter is based on the paper "LLM-powered Data Augmentation for Enhanced Cross-lingual Performance" (Whitehouse, Choudhury, and Aji, 2023) published in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023.*

7.1   Background and Introduction

The success of NLP models greatly depends on the availability and quality of training data. This poses a significant challenge for multilingual NLP, as data for languages other than English is typically limited (Ponti et al., 2019; P. Joshi et al., 2020; Whitehouse, Christopoulou, and Iacobacci, 2022). An approach to address the data scarcity challenge is through zero-shot cross-lingual transfer or multitask training, in which a model is trained across data of diverse tasks and languages, exhibiting the capability to handle unseen tasks, particularly in larger models (Artetxe and H. Schwenk, 2019; Nooralahzadeh et al., 2020; K.-H. Huang et al., 2021). However, when aiming for task-specific objectives, a smaller, fine-tuned model dedicated to that particular task often outperforms larger general-purpose, zero-shot models. In

DATASET	Train		Validation		Test	
	English	Non-English	English	Non-English	English	Non-English
XCOPA	400	0	100	100	500	500
XWinograd	1858	0	233	0	233	424
XStoryCloze	300	300	60	60	1511	1511

TABLE 7.1: Number of examples available in XCOPA, XWinograd, and XStoryCloze per language. Since a validation split is not specified in XStoryCloze, we take 60 random examples from the train split for validation.

addition, a smaller task-specific model is more practical and cost-effective for training and deployment. Nevertheless, developing a powerful task-specific model becomes challenging in the absence of training data (Lauscher et al., 2020).

Conversely, recent powerful Large Language Models (LLMs) excel at handling general instructions and have shown promise in data generation tasks (Yizhong Wang et al., 2023). In this work, we leverage LLMs to generate synthetic data for various multilingual commonsense reasoning tasks, XCOPA (Ponti et al., 2020), XWinograd (Tikhonov and Ryabinin, 2021), and XStoryCloze (X. V. Lin et al., 2022), where the training data is limited even for English (see Table 7.1). To augment the training data, we provide LLMs with instructions and examples from the original training data, prompting them to generate new and diverse examples. We explore the generation of synthetic data in English using different LLMs, including open-source models like Dolly-v2[1] and StableVicuna[2], as well as ChatGPT and GPT-4. Although the weights and capabilities of the latter two models remain undisclosed, we explore them as they extend the capability of generating texts in languages beyond English.

We develop task-specific models by fine-tuning multilingual pre-trained language models, namely mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020a), using the generated data. We then compare their performance against models trained on a limited set of human-created data in the target language whenever available, and otherwise through zero-shot transfer learning from manually created English training data. Our experiments demonstrate that training the models with *relatively large* synthetically generated datasets yields better performance than training with *limited* manually-created datasets. This finding empirically confirms the

---

[1] https://github.com/databrickslabs/dolly
[2] https://github.com/Stability-AI/StableLM

utility of synthetic data generated by LLMs for improving downstream task-specific models.

We expand the multilingual data synthesis using ChatGPT and GPT-4 on XCOPA and find that generating multilingual datasets generally surpasses the effectiveness of the zero-shot cross-lingual transfer. We further assess the quality of the generated dataset in different languages by asking native speakers to evaluate the naturalness and logical soundness of the generated dataset compared to the human-written examples. The annotation results reveal that while ChatGPT and GPT-4 successfully generate natural text in most languages, they struggle with generating understandable text in certain languages such as Tamil. Moreover, a noticeable gap is observed in terms of commonsense coherence when comparing ChatGPT-generated data to human-constructed data. On the other hand, GPT-4 significantly narrows this difference.

To summarise, our work has the following key contributions:

- Leveraging and prompting four LLMs to augment three low-resource, multilingual commonsense reasoning datasets.

- Fine-tuning smaller models, mBERT and XLMR, using the synthesised data and showcasing the practical value of the LLM-generated data.

- Performing an extensive analysis of the effects of various target languages in data generation and scaling, as well as a human evaluation of the naturalness and logical coherence of the data generated in various languages.

- Releasing the synthesised datasets for public use and reproducibility.

7.2 Related Work

We review the related work of the chapter in the following two aspects.

7.2.1 Multilingual and Low-Resource NLP

Recently, there has been increased attention on expanding NLP beyond English, including the development of multilingual models (Devlin et al., 2019; Conneau et al.,

2020a; Xue et al., 2021; Scao et al., 2022) as well as the creation of benchmarks to address multilingual challenges (Conneau et al., 2018; Artetxe, Ruder, and Yogatama, 2020; Adelani et al., 2021; Winata et al., 2023). Among the prevailing challenges faced across various languages, a common theme is the scarcity of available data.

Consequently, when data is lacking, one approach is to employ zero-shot cross-lingual transfer. Studies conducted by Winata et al. (2023) have demonstrated the effectiveness of zero-shot cross-lingual transfer for related languages. Additionally, Muennighoff et al. (2023) show that models fine-tuned only with English instruction data are capable of understanding multilingual instructions. In this work, we are tackling a similar scenario where the availability of data is limited.

7.2.2   Multilingual Data Augmentation

Lauscher et al. (2020) show that few-shot can drastically increase the cross-lingual performance of small models, proving that multilingual data augmentation is an effective strategy. A series of works try to predict the cross-lingual accuracy of models through measurements and modelling (Xia et al., 2020), and study strategies for multilingual data augmentation, such as choosing the transfer languages (Y.-H. Lin et al., 2019), and predicting multilingual few-shot accuracy leading for optimal data augmentation approaches (Srinivasan et al., 2022).

Many works focus on synthetic data augmentation for code-mixing, including utilising linguistic theories (Grandee Lee, Yue, and Haizhou Li, 2019; Pratapa et al., 2018), machine translation models (Tarunesh, Syamantak Kumar, and Jyothi, 2021), parallel corpus and Wikipedia (Winata et al., 2019; Whitehouse, Christopoulou, and Iacobacci, 2022), and employing ChatGPT (H. Dai et al., 2023). To the best of our knowledge, this is the first work that utilises LLMs for multilingual data augmentation, comparing data generation in English and then translating and generating in target languages, with a special focus on the challenging multilingual commonsense reasoning tasks.

7.3 Dataset Augmentation

This section explains the datasets used in the experiments and the detailed instruction setup.

Our experiments use XCOPA, XWinograd, and XStoryCloze, which are selected due to (i) the limited availability of training data and (ii) commonsense reasoning datasets present greater challenges for data synthesis. Table 7.1 summarises the statistics of the three datasets.

XCOPA is a cross-lingual Choice of Plausible Alternatives dataset that translates and re-annotates the validation and test sets of English (EN) COPA (Roemmele, Bejan, and Gordon, 2011) into 11 target languages (ET: Estonian, HT: Haitian Creole, ID: Indonesian, IT: Italian, QU: Quechua, SW: Swahili, TA: Tamil, TH: Thai, TR: Turkish, VI: Vietnamese, and ZH: Chinese).[3] Each instance consists of a premise, a question (*cause/result*), and two alternatives. The task is to predict the more plausible alternative.

XWinograd is expanded from the original English Winograd Schema Challenge (WSC) (Levesque, Davis, and Morgenstern, 2012) to five other languages (FR: French, JA: Japanese, PT: Portuguese, RU: Russian, and ZH),[4] which consists of pronoun resolution problems aiming to evaluate the commonsense reasoning ability of a machine. Given a statement with two noun phrases and a pronoun, the challenge of WSC is to determine the referent of the pronoun, which can only be inferred from the context.

XStoryCloze is collected by X. V. Lin et al. (2022), where the validation split of the original English StoryCloze dataset (Mostafazadeh et al., 2016) is translated into 10 other typologically diverse languages (RU, ZH, ES: Spanish, AR: Arabic, HI: Hindi, ID, TE: Telugu, SW, EU: Basque, and MY: Burmese). Each example consists of a four-sentence commonsense story, a correct ending, as well as a wrong ending.

---

[3] https://huggingface.co/datasets/xcopa
[4] https://huggingface.co/datasets/Muennighoff/xwinograd

7.3.1   LLMs for Data Generation

Our preliminary experiments reveal that language models that are specifically fine-tuned on downstream NLP tasks, such as BLOOMZ (Scao et al., 2022) and Flan-T5 (Chung et al., 2022), struggle to follow the complex instructions. Conversely, more recent LLMs such as Dolly-v2, StableVicuna, ChatGPT, and GPT-4, which are designed to handle more intricate and general-purpose instructions, have demonstrated success in following our instructions for data generation. ChatGPT and GPT-4 also stand out with the capability of generating examples in non-English languages.

We explore synthetic data generation with the four aforementioned LLMs, balancing between open-access models and closed models (see subsection 7.5.1). Specifically, we use `dolly-v2-12b`, which is derived from EleutherAI's Pythia-12b (Biderman et al., 2023a) and fine-tuned on approximately 15K instructions generated by Databricks employees; and `StableVicuna-13B`, an RLHF (reinforcement learning from human feedback) fine-tuned Vicuna model on various conversational and instructional datasets - Vicuna is an open-source LLaMA model (Touvron et al., 2023a) fine-tuned on user-shared conversations collected from ShareGPT.[5]

7.3.2   Instructions and Responses

We utilise LLMs to generate synthetic examples for all datasets by prompting them. We construct instructions using the descriptions from the dataset papers as a reference and provide LLMs with some examples, randomly sampled from the *train (+validation)* split of the original dataset, then ask LLMs to generate similar data points. We experiment with various instructions and evaluate the synthesised data on a smaller scale, update the instructions based on the errors, and then choose the best instruction to generate the final datasets.

The final instructions and responses are in Table 7.2. Our data generation process comprises the following key steps: (i) We establish the desired total number of examples to generate. This quantity can be determined by various factors such as budget constraints, a fixed ratio concerning the original dataset, etc. (ii) We proceed

---

[5]https://github.com/lm-sys/FastChat

XCOPA	XWINOGRAD	XSTORYCLOZE
👤 We are collecting more examples for the COPA dataset which will be used to test a system's ability of Commonsense Causal Judgments. The format of the data: A premise: a statement of something that happened, and two choices that could plausibly {*occur as the result / be the cause*} of the premise. The correct choice is the alternative that is more plausible than the wrong choice. Here are *n* examples in {`language`}: Example 1: Premise: The man wanted to save money. What happened as a result? Correct choice: He cut back on making frivolous purchases. Wrong choice: He withdrew money from his savings account. … Example *n*: … Based on the examples above, generate *m* new examples in {`language`}.	👤 We are collecting more examples for the Winograd Schema Challenge. Each example has a short sentence that contains two noun phrases and one pronoun replaced by "_", and the challenge is to determine the referent of the pronoun, which can only be inferred from the context. Here are *n* examples of the data: Example 1: Sentence: Harley hides from Dyna because _ is scary. Who/What is scary? Correct answer: Dyna. Wrong answer: Harley. … Example *n*: … Based on the examples above, generate *m* new examples. Both noun phrases in each example can be males, females, inanimate objects, or groups of people or objects. There should only be one "_" in the sentence. The correct and wrong answer should be one of the noun phrases mentioned in the sentence.	👤 We are collecting more examples for the Story Cloze dataset. Each example consists of a 4-sentence story, one correct ending sentence which is a plausible continuation of the story, and one wrong ending sentence which is logically inconsistent with the context. Here are *n* examples of the data: Example 1: Sent-1: Tina is very tired every single morning. Sent-2: She does not get enough sleep because of her two jobs. Sent-3: Tina decides to quit one of the jobs. Sent-4: She now gets enough sleep to function everyday. Correct ending: Tina is well rested. Wrong ending: Tina is more tired than ever before. … Example *n*: … Based on the examples above, provide *m* new similar examples. Requirements: 1) the story should read like a coherent story, with a specific beginning and ending, where something happens in between 2) both ending sentences should be entirely reasonable, realistic and sensible when read in isolation, and 3) both ending sentences should follow up the story by sharing at least one of the characters of the story.
💬 Premise: The politician made a controversial statement. What happened as a result? Correct choice: The politician faced criticism from the media. Wrong choice: The politician's approval ratings increased. 💬 Premise: 我裤子口袋里的钥匙不见了。 What was the cause? Correct choice: 这个口袋上有一个洞。 Wrong choice: 裤子是新的。	💬 Sentence: Sam gave Andrew the book because _ had already read it. Who/What had already read the book? Correct answer: Sam. Wrong answer: Andrew. 💬 Sentence: The dog chased the cat , but _ was too fast. Who/What was too fast? Correct answer: the cat. Wrong answer: The dog.	💬 Sent-1: Jordan was a high school student who wanted to become a doctor. Sent-2: He spent all his free time studying biology and chemistry. Sent-3: One day, his school hosted a science fair competition. Sent-4: Jordan's project won first place. Correct ending: Jordan went on to study medicine in college. Wrong ending: Jordan gave up his dream of becoming a doctor.

TABLE 7.2: Examples of instructions and LLM-responses for XCOPA, XWinograd, and XStoryCloze. We use ChatGPT for demonstration.

to generate examples through the following iterative process: (a) To ensure diversity, we randomly sample a set of *n* examples from the training datasets. (b) We append

Model	XCOPA	XWinograd	XStoryCloze
DOLLY-V2	41.6%	22.4%	41.2%
STABLEVICUNA	36.1%	33.8%	36.1%
CHATGPT	86.4%	43.8%	77.6%
GPT-4	89.7%	85.0%	89.3%

TABLE 7.3: Generation Success Rate in English (valid examples obtained / total examples requested) with different LLMs on the three datasets.

these sampled examples to the instructions and prompt the model to generate an additional set of *m* new examples. (c) Afterwards, we perform post-processing and only add valid and unique examples to the generated set. Typically, the values of *n* and *m* are set to 5 to 10.

We focus on a fixed-budget scenario and first generate a total of 3-4K data points for each dataset with LLMs. LLMs tend to generate fewer samples than requested or inconsistent output in invalid formats. We report the success rate for different LLMs on the three datasets in Table 7.3, which indicates that GPT-4 has the most robustness.

Among the datasets, LLMs have the lowest generation success rate for XWinograd, which is more challenging. XWinograd requires both answers to be from the generated sentence, with only one pronoun being replaced. One failed example in generated XWinograd: Sentence: "The computer crashed and _ lost all of their files". Correct answer: the user. Wrong answer: the computer. In addition, we observed pronoun inconsistency in the generated XWinograd data. Despite the requirement for interchangeable pronouns in the options, models frequently fail to comply. For example, "The dog bit the mailman because _ entered the yard." is generated by ChatGPT with the options "The dog" or "the mailman", however, "_" in the sentence cannot be replaced by the same pronoun for the given two options, hence it may make the task easier and the example is considered sub-optimal.

Despite multiple instructions emphasising the necessity of maintaining consistency between the two phrases to be replaced, all Language Model Models (LLMs) experimented with, failed to consistently follow this requirement. The best-performing instruction, as illustrated in Table 7.2, still fell short of achieving perfect consistency.

FIGURE 7.1: Comparison between the 30 most frequent events in the original and the ChatGPT-generated English StoryCloze dataset.

We retain these instances within the dataset and further include a human evaluation in subsection 7.6.1. Specifically, we found that among the LLMs studied, GPT-4 demonstrated the highest level of consistency, followed by Chat-GPT, with 76.6% and 48.9% of the annotated examples exhibiting adherence to the rule, respectively, as confirmed by annotations from native speakers.

7.3.3 Topic Diversity

As the StoryCloze dataset contains more sentences and has richer content, we analyse the diversity of the generation as well as topic coverage and the most frequent events, following Specifically, an event is counted as any hyponym of "event" or "process" in WordNet.

This helps us to determine whether LLM-generated data can capture the corpus distribution by randomly sampling n examples from the dataset in the instructions. ChatGPT-generated data is used for demonstration.

In Figure 7.1, we present the results of comparing the generated data points with the original 300 train set used as few-shot examples in the generation instructions. We can see that 23 of the 30 most frequent events in the original dataset can also be found in the 30 most frequent events of the ChatGPT-generated data.

7.4   Experimental Setup

We first generate synthetic English examples for XCOPA, XWinograd, and XStoryCloze, with Dolly-v2, StableVicuna, ChatGPT, and GPT-4. The size of the final filtered synthesised data for the three datasets is 3.7K, 2K, and 1.7K, respectively. We then fine-tune mBERT, XLMR-base, and XLMR-large with the synthesised data and compare the zero-shot cross-lingual transfer performance across different languages, where we use the original validation set in target languages.

For XCOPA, we additionally experiment with generating data points directly in non-English languages, by providing examples in the target language and specifying the language desired for the generated data (see Table 7.2). However, since no examples for *cause* are included in TH and TR train/validation data (they do appear in the test split), we do not generate XCOPA for the two languages. We use ChatGPT and GPT-4 for multilingual synthetic data generation, as both Dolly-v2 and Stable-Vicuna exhibit limitations in effectively generating multilingual text. The size of the multilingual synthesised data is ~3.6K in each language.

We fine-tune models on all datasets as multiple-choice tasks[6] by searching the best learning rate from $\{5e^{-6}, 10e^{-6}\}$, and batch size from $\{8, 16, 32\}$. All the fine-tuning experiments are conducted on a single 40G A100. For generating data with Dolly-v2 and StableVicuna, we use 2×40G A100. Additionally, for XLMR-large trained on the original XWinograd dataset, we found that a lower learning rate was necessary. Therefore, we performed an additional tuning process with a learning rate of $10e^{-7}$.

7.5   Results and Discussion

We now present the main results of fine-tuned models on the three datasets and compare performance with generated data in different LLMs, languages, and scales.

---

[6]In our preliminary experiments, we find that formulating XWinograd as a binary text classification results poorly, in line with the observation from Haokun Liu et al. (2020) that the task formulation is essential to the performance of Winograd.

Finetuned Model	LLM for Generation	XCOPA			XWINOGRAD			XSTORYCLOZE		
		ORI_{400}	$GEN_{3.7k}$	$O+G_{4.1k}$	$ORI_{1.8k}$	GEN_{2k}	$O+G_{3.8k}$	ORI_{300}	$GEN_{1.7k}$	$O+G_{2k}$
mBERT	Dolly-v2	47.9	53.3 ↑5.4	54.0 ↑6.1	52.9	**59.6** ↑6.7	**59.3** ↑6.4	65.0	**68.7** ↑3.7	68.1 ↑3.1
	StableVicuna	47.9	52.9 ↑5.0	54.7 ↑6.8	52.9	53.7 ↑0.8	58.5 ↑5.6	65.0	64.6 ↓0.4	67.3 ↑2.3
	ChatGPT	47.9	55.0 ↑7.1	54.1 ↑6.2	52.9	56.0 ↑3.1	58.3 ↑5.4	65.0	64.3 ↓0.7	68.3 ↑3.3
	GPT-4	47.9	**56.4** ↑8.5	**57.2** ↑9.3	52.9	54.9 ↑2.0	57.5 ↑4.6	65.0	68.0 ↑3.0	**69.8** ↑4.8
XLMR-Base	Dolly-v2	54.8	58.1 ↑3.3	58.1 ↑3.3	53.5	56.5 ↑3.0	66.3 ↑12.8	73.0	75.8 ↑2.8	76.5 ↑3.5
	StableVicuna	54.8	57.6 ↑2.8	59.3 ↑4.5	53.5	59.0 ↑5.5	66.0 ↑12.5	73.0	69.6 ↓3.4	74.2 ↑1.2
	ChatGPT	54.8	58.2 ↑3.4	59.4 ↑4.6	53.5	62.7 ↑9.2	65.9 ↑12.4	73.0	67.4 ↓5.6	74.5 ↑1.5
	GPT-4	54.8	**62.7** ↑7.9	**63.0** ↑8.2	53.5	**63.3** ↑9.8	**66.9** ↑13.4	73.0	**74.6** ↑1.6	**79.3** ↑6.3
XLMR-Large	Dolly-v2	63.0	58.6 ↓4.4	65.0 ↑2.0	80.1	**76.9** ↓3.2	83.1 ↑3.0	85.0	84.8 ↓0.2	86.4 ↑1.4
	StableVicuna	63.0	64.4 ↑1.4	68.7 ↑5.7	80.1	68.2 ↓11.9	82.0 ↑1.9	85.0	74.6 ↓10.4	84.8 ↓0.2
	ChatGPT	63.0	64.6 ↑1.6	68.1 ↑5.1	80.1	73.2 ↓6.9	83.2 ↑3.1	85.0	77.3 ↓7.7	85.8 ↑0.8
	GPT-4	63.0	**72.1** ↑9.1	**72.2** ↑9.2	80.1	76.4 ↓3.7	**83.5** ↑3.4	85.0	**86.0** ↑1.0	**88.4** ↑3.4

TABLE 7.4: Comparison of Average Accuracy across all languages for mBERT, XLMR-Base, and XLMR-Large on XCOPA, XStoryCloze, and XWinograd. Training datasets include *ORI* (original EN data), *GEN* (LLM-generated EN data), and *O+G* (both), with the number of examples used for training indicated by the subscripts. The best results obtained with the same amount of training data are highlighted in bold. Green and red subscripts denote improvement and decline in performance compared to the baseline (*ORI*). See per language results in Appendix D.

7.5.1 General Result

Table 7.4 presents the average accuracy of fine-tuned mBERT, XLMR-Base, and XLMR-Large models across all languages on the three datasets. The models are trained using original data (*ORI*), different LLM-generated data (*GEN*), as well as a combination of both sources (*O+G*) in English.

Across different datasets, LLMs, and fine-tuned models, consistent improvements are observed when using both original and LLM-generated data. Among the models, Dolly-v2 performs the best on XWinograd when fine-tuned on mBERT, while GPT-4 achieves the highest accuracy in other settings. The most significant improvement is shown in XWinograd with XLMR-Base, where the addition of an extra 2k data points leads to an average accuracy enhancement of 12.8 compared to the baseline, across all four LLMs.

We observe the following from the results: (i) In instances where the baseline performance is below 60, exemplified by XCOPA and XWinograd tasks using small models like mBERT and XLMR-Base, leveraging LLM-generated data either solely or in conjunction with original data consistently enhances performance across the

board; (ii) As the baseline performance increases, notably exceeding 80 accuracy as seen in the case of XStoryCloze, relying solely on LLM-generated data might adversely impact performance. However, even in such cases, utilising data generated solely by GPT-4 still yields improvements across all fine-tuned models; (iii) Combining original data with LLM-generated data consistently demonstrates enhancements over using original data alone, irrespective of the baseline performance or the specific datasets and fine-tuned models employed. This trend holds true across various scenarios, with the exception of XStoryCloze with XLMR-Large, where there's only a marginal 0.2 score difference.

Overall, we can see that LLM-based data augmentation particularly benefits scenarios where baseline performance is moderate to low, while stronger LLMs show more promises in generating high-quality data, as exemplified by GPT-4.

For smaller models with When using only LLM-generated data, smaller models like mBERT and XLMR-Base generally outperform the baseline. However, with XLMR-Large, which achieves stronger baselines, e.g., >80 in XWinograd and XStoryCloze, the accuracy remains similar or even worse compared to using the original data. GPT-4-generated data demonstrates the best robustness but still experiences a decline in performance in XWinograd when the generated data size is similar to the original data. This highlights the challenges of generating data at a human-level quality.

7.5.2   Multilingual Data Generation

We investigate whether the synthetically generated multilingual dataset outperforms training solely in English. We choose the XCOPA dataset and explore two settings: synthetic multilingual data by asking LLMs to generate responses in the target languages directly and translating the English-generated data to target languages with Google Translate API. We exclude Dolly-v2 and StableVicuna due to their limited effectiveness in generating non-English text. Although GPT-4 exhibits the most promising performance, it is significantly costlier compared to ChatGPT. Therefore, we also consider ChatGPT as a contrasting experiment under resource-constrained conditions.

Finetuned	LLM	Training data	AVG	EN	ET	HT	ID	IT	SW	TA	VI	ZH
mBERT	Baseline	*ORI*	47.2	53.8	44.2	48.6	47.2	46.2	45.4	48.4	43.6	47.4
	ChatGPT	$GEN_{EN} + ORI$	54.6	59.6	56.4	53.6	53.8	51.4	51.6	50.4	55.0	59.2
		$GEN_{XX} + ORI$	56.8	59.6	58.8	54.6	56.2	61.2	54.6	53.6	52.0	60.2
		$GEN_{EN}^{Trans} + ORI$	58.7	59.6	59.8	58.2	62.8	61.0	52.6	56.8	58.2	59.4
	GPT-4	$GEN_{EN} + ORI$	59.3	72.6	58.8	53.0	62.0	61.0	50.0	54.0	57.6	64.6
		$GEN_{XX} + ORI$	61.8	72.6	61.2	58.2	62.2	66.4	57.4	53.4	63.0	61.8
		$GEN_{EN}^{Trans} + ORI$	62.6	72.6	58.6	55.2	65.6	65.4	53.8	62.6	64.6	65.4
XLMR-Base	Baseline	*ORI*	55.6	57.6	54.6	50.6	59.6	54.8	55.0	53.4	54.8	59.6
	ChatGPT	$GEN_{EN} + ORI$	59.8	63.8	61.6	51.6	62.6	59.8	51.6	60.4	64.8	62.0
		$GEN_{XX} + ORI$	59.9	63.8	60.6	55.0	64.6	59.6	54.6	56.4	59.6	64.8
		$GEN_{EN}^{Trans} + ORI$	61.1	63.8	60.0	58.0	65.0	60.8	53.8	60.2	62.6	66.0
	GPT-4	$GEN_{EN} + ORI$	63.6	69.6	63.8	51.2	67.2	62.4	58.4	63.8	66.8	69.4
		$GEN_{XX} + ORI$	64.0	69.6	62.2	56.2	68.6	63.8	57.8	61.2	66.8	70.0
		$GEN_{EN}^{Trans} + ORI$	63.9	69.6	61.6	56.6	68.4	65.2	58.2	60.2	66.0	69.6
XLMR-Large	Baseline	*ORI*	64.4	71.4	62.8	51.4	69.0	65.8	60.6	62.0	69.4	66.8
	ChatGPT	$GEN_{EN} + ORI$	69.5	76.4	69.8	48.2	76.0	72.8	63.4	67.8	73.4	77.8
		$GEN_{XX} + ORI$	65.2	76.4	62.4	55.2	75.0	62.2	58.2	55.4	66.2	76.2
		$GEN_{EN}^{Trans} + ORI$	67.0	76.4	60.0	59.6	66.2	66.6	59.0	64.8	74.8	75.6
	GPT-4	$GEN_{EN} + ORI$	73.7	84.6	70.4	50.0	80.8	80.2	65.8	72.8	78.4	80.4
		$GEN_{XX} + ORI$	74.6	84.6	77.0	56.0	82.2	77.0	65.0	73.8	76.2	80.0
		$GEN_{EN}^{Trans} + ORI$	74.1	84.6	74.2	57.2	82.0	77.4	62.2	75.0	74.4	79.6

TABLE 7.5: Accuracy on XCOPA. *ORI* corresponds to the original data, GEN_{EN} and GEN_{XX} represents data generated in English and target languages. *Trans* denotes translations of the English-generated data. We show languages that are available in all settings. Improvement and decline in performance are represented with green and red shadows.

Table 7.5 shows the results for the languages that are available for all settings, excluding TR and TH (unavailable for LLM-generation, refer to section 7.4), and QU (not supported by the Google Translate API). We can see the impact of the generated data varies across different fine-tuned models and languages, aligning with the findings of Shanu Kumar, Dandapat, and Choudhury (2022). Training on GPT-4 synthesised data displays consistent improvement across all scenarios and languages, except the zero-shot cross-lingual result on HT with XLMR-Large.

More fluctuating results can be observed with ChatGPT-generated data. A comparison between $GEN_{EN} + ORI$ and $GEN_{XX} + ORI$ indicates that utilising data generated in target languages generally leads to improved performance with GPT-4 generated data, as well as in base models with ChatGPT-generated data. However, for XLMR-Large, employing ChatGPT-generated data in target languages mostly yields negative outcomes. In languages such as TA and VI, training on generated data in

Model	$GEN_{EN} + ORI_{EN}$		$GEN_{EN}^{Trans} + ORI_{EN}$	
	3.7K	28.6K	3.7K	28.6K
mBERT	54.3	56.0	58.0	**60.1**
XLMR-Base	60.1	**61.8**	61.2	61.7
XLMR-Large	69.7	**72.4**	67.2	71.4

TABLE 7.6: Accuracy on XCOPA when scaling up the generated data to over 28K with ChatGPT. We report average results on all XCOPA languages excl. QU, since it is not available with the Google Translate API.

the target languages results in more performance degradation compared to zero-shot cross-lingual transfer. This suggests that ChatGPT performs worse in those languages than XLMR-Large (Ahuja et al., 2023).

Translating the English dataset generally shows overall better results than training on the data generated directly in the target languages, except for XLMR-Large with GPT-4. For SW, both XLMR-Base and XLMR-Large models fined-tuned with ChatGPT-generated data exhibit performance decline in most cases, even when the English-generated data benefits all other languages. This observation suggests that XLMR struggles with SW. In subsection 7.6.1, we select TA, SW, and the two best languages, ID and ZH, along with EN, for human evaluation.

Furthermore, we explore the effects of incorporating Target Languages in Validation (TLV). This approach involves training on English examples but evaluating and testing on the target language. The detailed results are presented in Table D.4 in Section D. Notably, for smaller models trained on limited data (e.g., 4.1K examples), integrating target languages during validation led to significant performance boosts of 3.0 and 0.9 for mBERT and XLMR-Base, respectively. However, when considering larger models like XLMR-large and smaller models trained with more extensive datasets (e.g., 29K examples), the impact of including target languages during validation was less pronounced. In these cases, we observe only minor variations in performance. These findings align with those of Ponti et al. (2020), suggesting that the effectiveness of TLV may vary depending on factors such as model size and training data availability.

Model	Ratio	XCOPA	XWingrad	XStoryCloze
mBERT	1×	64.0	50.2	74.6
	2×	64.8	51.9	76.8
	5×	68.0	57.1	80.6
	10×	69.8	65.7	80.3
XLMR-Base	1×	58.0	45.9	70.7
	2×	59.0	53.7	79.7
	5×	63.0	67.8	81.9
	10×	65.8	71.2	84.1
XLMR-Large	1×	56.0	78.1	81.1
	2×	61.2	79.8	90.9
	5×	81.4	82.0	89.9
	10×	85.2	82.8	91.9

TABLE 7.7: Performance on English test examples training on GPT-4-generated English data and the original data. Original data points selected from the three datasets are set to 200. 1× corresponds to using only the original data, 2× means using 200 original data and 200 generated data.

7.5.3 Dataset Scaling Up

We now investigate the impact of training on a larger scale of generated data on model performance. We focus on the XCOPA dataset and expand the generated data with ChatGPT (more budget-efficient) to 28.6k examples in English. We also compare the results of zero-shot cross-lingual transfer with translating the English-generated data to target languages.

The results in Table 7.6 demonstrate the positive impact of scaling up the generated data on model performance. Particularly, XLMR-Large exhibits the most significant improvement.

7.5.4 Fixed Ratio Data Augmentation

We experiment with generating data with a fixed ratio of the original datasets. Specifically, we compare training with the original English data (200 randomly selected examples) and augment it with different quantities of English examples generated by GPT-4, where we include original training instances in all cases.

The results in Table 7.7 showcase the performance on English test examples when fine-tuning mBERT and XLMR models with training data sizes that are 1×, 2×, 5×, and 10× the size of the original dataset. We can see that performance consistently

improves as we increase the amount of generated data except XStoryCloze, which has the highest baselines, echoing the previous findings. The relative performance gain is generally more pronounced when increasing the data from $2\times$ to $5\times$ for the other two datasets.

7.6 Human Evaluation

To better evaluate the quality of the generated datasets and compare them with the human-created data, we ask native speakers to annotate the multilingual data generated by ChatGPT and GPT-4.

For each dataset, we first select 50 generated examples in English, and then request two annotators to evaluate the examples in two categories: (i) **Text Naturalness**. The annotators are asked to choose one of the following options for each example: "the text sounds natural", "the text sounds awkward but understandable", or "the text is not understandable", and (ii) **Logic Soundness**. This category focuses on the commonsense aspect of the examples. The annotators are required to select the most appropriate description from: "the correct option is (clearly) more plausible", "both options are equally plausible", "both options are implausible", or "the wrong option is more plausible". We only ask the annotators to evaluate the logic if the text is at least understandable.

For XWinograd, we introduce an additional evaluation criterion. Annotators are asked to determine whether the two noun phrases in the examples can be replaced by the same pronoun (refer to subsection 7.3.2). For XCOPA, we extend the annotations to non-English languages, where we choose the two languages that demonstrate the most notable improvement, namely ZH and ID, as well as the two languages that exhibit the least improvement or regression in performance with ChatGPT-generated data, namely TA and SW (see Table 7.5). In addition to the original examples and the generated examples in the target languages, we include 50 examples that are translated from the same English-generated examples (that were selected for annotation).

To ensure impartiality, all the examples are shuffled, and the annotators are not provided with information regarding the source of the examples (human-created,
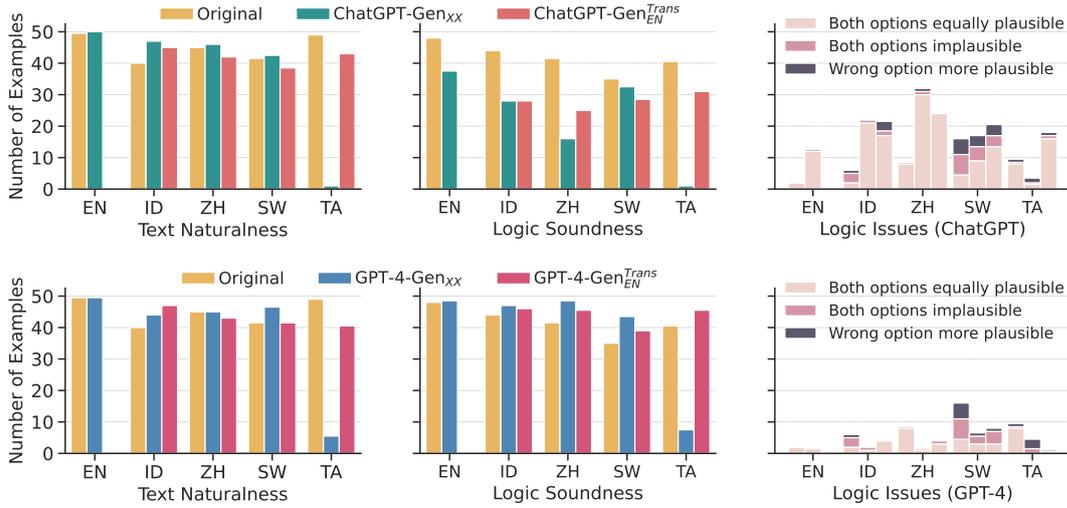
FIGURE 7.2: Human evaluation of 50 random examples from the original XCOPA, ChatGPT (top) and GPT-4 (bottom) generated data in target languages, and translation of English generated data. Examples are annotated by two native speakers in each language. The subplots in the last column show the logic issues of the XCOPA data, where the three bars for each language represent *Original*, Gen_{XX}, and Gen_{EN}^{Trans} (from left to right).

LLM-generated, or translated).

7.6.1 Text Naturalness

Figure 7.2 presents the annotation results for XCOPA, averaged from two annotators for each language. For Text Naturalness, we can see that in EN, ID, ZH, and SW, both ChatGPT and GPT-4 achieved higher naturalness than the original dataset. This is particularly prominent in ID, revealing the fluency issue in the original ID data in XCOPA, which is also confirmed by a native speaker.

Issues with Tamil

In contrast, the performance of the TA dataset is surprisingly low, with a majority of examples classified as "not understandable." Upon consulting language experts, we have identified several main issues in Tamil, including (i) the insertion of redundant words with the same meaning, such as "I will retry to try it again" (ii) verb agreement errors, and (iii) the presence of uncommon and out-of-context words.

It is worth noting that generating Tamil using GPT-4 is both slow and costly. We suspect that the tokeniser for Tamil, as well as similar languages like Telugu and

Kannada, are poorly trained, resulting in unusable generation in those languages. While the low quality of the generated data could explain the significant decline in the performance of the XLMR-Large model when trained on ChatGPT-generated data in Tamil, intriguingly, models trained on Tamil data generated by GPT-4 show improvement over the baselines.

To further investigate this issue, we conduct an experiment where we fine-tune the models using only five examples from the TA examples generated by GPT-4 that are identified as natural and sound by the annotators. The improvement on mBERT under this setting is 50% of the total improvement seen with the entire 3.6K TA examples. For XLMR-base and XLMR-large, 15% and 3% of the total improvement can be observed, respectively. Considering that the estimated number of correct samples in the 3.6k dataset is around 360, it is plausible that training solely on those examples could raise the accuracy level, or even surpass, what we observe for the entire dataset.[7] An intriguing question that remains to be investigated in future research is why the remaining 3.2k incorrect or unnatural examples do not negatively impact the model's performance.

The translated text is typically less natural than the original and generated data (apart from ID due to issues in the original data). This result affirms that LLMs generally excel in generating fluent text for the languages it supports.

7.6.2   Logic Soundness

In terms of logic soundness, ChatGPT falls short compared to the original dataset. We further illustrate the categorised issues in the last column of the plots in Figure 7.2. We can see that for ChatGPT, the majority of the examples are labelled as "both options are equally plausible", only SW has more problematic examples with "the wrong option is more plausible". We suspect that this issue arises from the instruction provided (taken from the description of the original COPA dataset), which states that "both options could be plausible, but one is more plausible." In some cases, ChatGPT generates two choices that are excessively similar in terms of plausibility. On the other hand, GPT-4 tends to generate options with more clearcut differences in plausibility, mirroring the original data. We note that despite the

---

[7]We could not conduct this experiment as the entire dataset was not manually labelled.

description and instruction that both alternatives could happen, both the original dataset and the data synthesised by GPT-4 tend to present one plausible and one *implausible* option.

For English XWinograd and XStoryCloze, the majority of the examples in both original and generated examples are evaluated as natural and logically sound. For XWinograd, although more than 47 examples are evaluated to exhibit high text quality and follow commonsense logic, as mentioned in subsection 7.3.2, only 23 ChatGPT-generated examples fulfil the requirement that both noun phrases should be interchangeable with the same pronoun. GPT-4 examples demonstrate better consistency, with 36 following this rule, whereas all original examples are found satisfactory.

7.7 Conclusion and Future Work

This chapter explores the effectiveness of utilising LLMs for data augmentation in cross-lingual datasets with limited training data. We specifically focus on commonsense reasoning tasks that are challenging for data synthesis. Our experiments including four LLMs for data generation on three datasets, showcase enhanced cross-lingual zero-shot transfer on smaller fine-tuned task-specific language models. However, the impact varies across different datasets and languages. Notably, larger models such as XLMR-Large, which have higher baselines, demonstrate more difficulty in achieving performance improvements with LLM-generated data. Among the four LLMs, GPT-4-generated data exhibits mostly consistent superior performance.

Expanding data generation directly in target languages also shows general improvements compared to cross-lingual zero-shot with the English-generated data. Human evaluation of the synthesised multilingual dataset shows that the ChatGPT and GPT-4 generated data demonstrate high naturalness in most languages, even surpassing the original data. However, in certain languages like TA, both models fail to generate natural text. Additionally, when assessing the logical soundness of the dataset, examples synthesised by ChatGPT reveal notable inconsistencies regarding more plausible options compared to the original human-created data. In contrast, GPT-4 exhibits a performance on par with human-written data.

In conclusion, leveraging LLMs for data augmentation shows promise. However, the choice of LLM used for data generation significantly influences the quality of the resulting data, as well as its applicability to the language under consideration. In circumstances where a more advanced model such as GPT-4 cannot be accessed, other models can be utilised, though this might result in performance difficulties in certain non-English languages, a challenge that also exists for GPT-4, and concerns regarding logical coherence. A compelling direction for future research could involve exploring the efficacy of more recent instruction-tuned or aligned open-source LLMs, such as LLaMA 2 (Touvron et al., 2023b) or TÜLU (Zeqiu Wu et al., 2023), in enhancing data augmentation.

7.8   Limitations

The identified limitations in this chapter are as follows:

- **Language Resource Challenges**: While LLMs, particularly GPT-4, showcase promising results in the realm of multilingual commonsense data augmentation, challenges may arise when applied to extremely low-resource languages. The effectiveness of these models could be constrained by the availability of language-specific resources.

- **Dependency on Few-Shot Examples**: To achieve optimal performance, the incorporation of few-shot examples in the target language remains necessary for generating new examples. However, obtaining such examples may pose challenges, particularly for languages with limited available resources.

- **Closed Model Accessibility**: The usage of closed models like GPT-4 is restricted by licensing limitations, and the reproducibility of results obtained from these models may be compromised. Despite these constraints, the conducted experiments in this chapter illustrate the potential advantages of leveraging LLMs for multilingual dataset augmentation.

- **Black-box LLMs**: In addition to the limitation of the closed models, currently, only a handful of models are *truly* open with training code, model checkpoints,

and training data (i.e., BLOOM (Scao et al., 2022), the Pythia suite (Biderman et al., 2023b), and OLMO (Groeneveld et al., 2024)). All LLMs studied in this chapter are not *truly* open or with detailed pre-training data released. This lack of transparency raises concerns regarding the potential biases and shortcomings embedded within these "black-box" LLMs. Without full disclosure of their training data sources and methodologies, there exists the possibility that such models were pre-trained on datasets that overlap with the test sets of public benchmarks, including those used in our experiments. This scenario could introduce a confounding factor, rendering performance evaluations of LLMs on downstream tasks and data augmentation unreliable and potentially skewed.

Chapter 8

Conclusion

This thesis presents five focused studies that explore knowledge-grounded natural language understanding and generation, covering knowledge-enhanced fake news detection, multilingual knowledge-enhanced cross-lingual transfer, faithful and robust knowledge extraction from the web, grounded answer and explanation generation for knowledge-intensive VQA, and the employment of LLMs for data augmentation in tasks requiring multilingual commonsense knowledge.

Throughout this thesis, our focus centres on studying the utilisation and extraction of knowledge, whether in the form of structured knowledge, multilingual knowledge, parametric knowledge, or knowledge represented as augmented data distilled from powerful LLMs.

Addressing the research questions outlined in Chapter 1, we hereby present the conclusions derived from this thesis.

Application of Structured Knowledge

This thesis investigates the broader applicability of structured knowledge beyond entity-centric tasks. Through our experiments involving four distinct techniques to incorporate knowledge related to entities across two diverse fake news detection datasets, we find that leveraging structured knowledge improves applications such as fake news detection, if the applied knowledge is relevant and current.

Multilingual Entity Knowledge

This thesis broadens the scope of structured knowledge to a multilingual setup. Specifically, we find that utilising multilingual entity knowledge via (i) the creation

of an entity-centric code-switched corpus using data from Wikipedia and Wikidata, (ii) the intermediate training of a pre-trained multilingual language model, incorporating masked language modelling and entity prediction objectives, and (iii) fine-tuning the intermediate-trained model on downstream tasks, can consistently and significantly enhance the cross-lingual transferability.

Knowledge Extraction from Web Text

This thesis explores approaches that facilitate the effective and accurate extraction of information of knowledge from the vast and often noisy web text. Upon collecting a novel mWEBIE dataset and benchmarking against mWEBIE, we conclude that (i) the inclusion of negative examples within the dataset, and (ii) the integration of entity-centric auxiliary tasks, are beneficial to the successful extraction of structured knowledge from the inherently noisy web text.

Grounded Answer and Explanation Learning

This thesis highlights the benefit of multi-task grounded answer and explanation generation for knowledge-intensive VQA. Notably, we find that even without explicitly introducing external knowledge, which is commonly deemed a prerequisite in the datasets experimented, the adoption of this multi-task learning approach contributes to a notable enhancement in both answer accuracy and explanation quality, showcasing the advantage of better utilisation of the parametric knowledge stored within the model's parameters.

LLM-Powered Data Augmentation

This thesis leverages the inherent knowledge of recent powerful LLMs to enhance the performance of smaller, task-specific models. We find that data augmentation by prompting LLMs significantly improves complex and scarce multilingual common-sense reasoning tasks for smaller models, underscoring promising directions in data augmentation with LLMs and broader knowledge distillation methods.

8.1 Future Work

Having summarised key insights aimed at optimising the utilisation of knowledge in NLP applications throughout this thesis, we conclude with a vision for promising future research directions as follows.

Dynamic Knowledge Integration with Emphasised Grounding

As the capabilities of LLMs continue to advance, the incorporation of knowledge emerges as a crucial supplement to maintain the generality of LLMs while ensuring they remain up-to-date and finely adapted to domain-specific tasks.

Current approaches, including the studies in this thesis, often involve single or pre-defined knowledge sources based on the targeted downstream tasks. For instance, in Chapters 3-5, we employed *static* knowledge bases for both pre-training and downstream tasks. Although these strategies demonstrate effectiveness in benchmark tasks, they risk obsolescence without re-training. Hence, there is a growing need for more adaptive and dynamic knowledge integration methods in real-world applications.

A promising approach introduced in Chapter 2 is retrieval-augmented language models for text generation, demonstrating the potential of leveraging dynamic unstructured knowledge (P. Lewis et al., 2020; Guu et al., 2020; Izacard et al., 2023). While extensive research focuses on effective *retrieval* of documents (Borgeaud et al., 2022; X. Ma et al., 2023; Soares et al., 2023) and *ranking* of these documents (Zerveas, Rekabsaz, and Eickhoff, 2023), limited attention is given to how these models are *grounded* with the provided context. Questions arise, such as whether the model can learn to disregard irrelevant retrieved context (Yoran et al., 2024), and how the generation is impacted by the context retrieved from end-to-end systems versus post hoc retrieval systems. We posit that a more thorough investigation into the grounding of retrieval-augmented systems is needed.

Knowledge Integration with Mixture-of-Experts

In addition to dynamic knowledge integration, aligned with the concept of *general-purpose* LLMs, there is a compelling need for a *general-purpose* knowledge augmentation approach, one that does not necessitate the explicit specification of domain knowledge.

Revisiting the investigation on fake news detection in Chapter 3, we noted the importance of knowledge base relevance to the target task. Specifically, the integration significantly enhances the LIAR datasets but only marginally improves COVID-19. Consequently, it may become desirable to alleviate the need for predetermined knowledge bases for model training.

A promising direction for such exploration involves extending the adapter approach (R. Wang et al., 2021) as introduced in Chapter 2. This entails leveraging different knowledge sources to train distinct adapters, thereby enhancing the adaptability of LLMs to various domains. Beyond adapters, broader parameter-efficient fine-tuning methods, such as Low-Rank Adaptation (LoRA) (E. J. Hu et al., 2022), are also promising. By retaining the pre-trained weights of powerful LLMs, it becomes possible to train different versions of knowledge adapters or low-rank matrices. Subsequently, a carefully designed routing mechanism becomes crucial to determine the appropriate knowledge component for a given context.

While related work has demonstrated the advantages of the mixture of experts (Du et al., 2022; Shen et al., 2024), the mixture of adapters (Yaqing Wang et al., 2022), and the mixture of few-shot LoRA learning (C. Huang et al., 2023) for cross-task generalisation, we believe there exists unexplored potential for knowledge augmentation combined with the mixture of experts (Diao et al., 2023).

Factuality in Text Generation

Recent advanced LLMs excel at text fluency, typically empowered by large-scale pre-training and methods that leverage rankings over responses, such as reinforcement learning from human feedback (RLHF) (Ziegler et al., 2019; L. Ouyang et al., 2022). However, language models are susceptible to producing convincing yet factually inaccurate claims, commonly referred to as "hallucinations" (Tian et al., 2023).

This underscores the critical need for advancements in factuality and faithfulness in text generation (Augenstein et al., 2023). An expanded exploration of this thesis on knowledge grounding, in this context, plays a crucial role in determining what is *factual* or *faithful*, and in controlling the generation (P. Xu et al., 2020; Rashkin et al., 2021; Brahman et al., 2022).

Researchers have made significant efforts in enhancing actuality in text generation, such as fine-tuning with automatically generated factuality preference rankings (Tian et al., 2023), employing factual-nucleus sampling (N. Lee et al., 2022), training models to self-evaluate for faithfulness (Kadavath et al., 2022), among others. Despite these advancements, there remains a considerable amount of unexplored directions, such as the development of better evaluation metrics. Current metrics, whether n-gram-based or relying on trained neural models like BERTScore (T. Zhang et al., 2020), often fall short in capturing factuality (E. Clark et al., 2023; Aharoni et al., 2023). Moreover, commonly used entailment-based Natural Language Inference scores for faithfulness evaluation (Maynez et al., 2020) offer limited interpretability and may be biased by the underlying pre-trained models.

Therefore, we anticipate that training models for more faithful and grounded generation, as well as developing metrics to evaluate factuality performance, will be highly relevant research directions.

8.2   Summary

In this chapter, we highlight key findings and conclusions of the thesis, along with exciting and promising directions for future work including dynamic knowledge integration with emphasised knowledge grounding, knowledge integration with mixture of experts, and knowledge-grounded text generation for enhanced factuality. Quoting Prof Christopher D. Manning,[1] we echo the sentiment of NLP research "Nothing but blue skies!" as we look forward to the boundless possibilities that lie ahead in the ever-evolving language technology.

---

[1]https://2023.emnlp.org/program/keynotes

Appendix A

Language-Specific Results for ENTITYCS Experiments

We report the per-language results of the experiments on ENTITYCS in the following tables. WikiAnn results can be found in Table A.1. X-FACTR results in Table A.2, MultiATIS++ Slot Filling-only training in Table A.3, and XL-WiC in Table A.4.

MODEL	AR	HE	VI	ID	JV	MS	TL	EU	ML	TA	TE	AF	NL	EN
XLM-R$^{\text{OURS}}$	44.6	51.9	68.3	48.6	59.6	63.3	72.5	61.2	63.2	54.3	49.3	76.3	80.7	83.4
MLM	50.7	53.7	72.7	56.4	59.2	68.4	75.1	58.4	65.1	58.1	53.0	76.3	80.9	84.2
WEP	49.9	52.4	69.8	57.4	60.1	66.7	74.0	60.1	60.8	56.1	48.2	76.5	80.3	83.8
PEP$_{\text{MRS}}$	47.1	52.6	69.8	56.0	60.1	62.4	74.8	56.1	61.6	56.1	50.9	77.9	81.4	83.8
PEP$_{\text{M}}$	47.7	52.9	68.9	59.1	63.1	65.5	76.3	60.0	64.0	57.5	51.6	76.8	80.9	83.9
WEP+MLM	50.3	53.2	69.8	60.8	60.7	69.8	74.5	59.2	64.8	57.2	51.7	76.4	80.9	84.1
PEP$_{\text{MRS}}$+MLM	46.7	53.6	69.6	64.0	60.2	69.2	74.3	57.5	65.6	55.8	52.3	77.5	81.3	84.1
PEP$_{\text{M}}$+MLM	52.4	53.5	70.4	60.3	59.7	69.2	75.5	58.4	66.6	58.1	54.5	77.7	81.2	84.1

	DE	EL	BN	HI	MR	UR	FA	FR	IT	PT	ES	BG	RU	JA
XLM-R$^{\text{OURS}}$	75.4	74.2	67.9	68.3	61.8	55.8	47.6	78.0	78.2	78.9	76.2	77.3	63.9	22.9
MLM	75.2	76.3	73.9	69.9	64.5	67.0	51.6	79.0	78.6	79.5	77.6	78.6	67.2	22.7
WEP	74.7	74.5	70.8	67.5	61.1	60.7	50.9	77.6	77.7	77.3	74.1	78.7	66.3	20.7
PEP$_{\text{MRS}}$	75.4	74.8	69.6	68.3	64.1	48.7	53.0	78.9	78.6	78.7	77.1	78.6	67.3	21.9
PEP$_{\text{M}}$	75.1	76.5	73.0	69.6	65.8	63.3	55.3	78.5	78.4	78.6	74.3	78.2	67.2	21.0
WEP+MLM	75.4	75.2	72.1	68.9	63.9	58.6	53.4	78.1	78.3	79.0	74.9	78.3	66.6	23.0
PEP$_{\text{MRS}}$+MLM	75.5	76.2	72.2	68.4	64.8	60.1	54.3	79.2	79.1	80.0	76.4	79.1	67.6	23.6
PEP$_{\text{M}}$+MLM	75.3	76.2	74.2	70.0	67.1	64.5	50.0	79.9	78.9	79.7	78.4	79.3	68.2	22.7

	KA	KO	TH	SW	YO	MY	ZH	KK	TR	ET	FI	HU		
XLM-R$^{\text{OURS}}$	66.4	48.8	4.3	68.3	45.4	52.7	27.7	44.2	76.9	72.4	75.6	76.9		
MLM	66.1	50.8	2.5	65.1	42.9	55.7	29.7	50.7	77.8	71.4	76.2	78.1		
WEP	64.8	52.0	2.5	65.8	50.4	52.6	26.1	52.1	75.5	71.9	75.8	76.6		
PEP$_{\text{MRS}}$	63.6	51.4	3.7	66.2	45.9	54.6	26.6	49.1	78.0	72.8	77.2	77.7		
PEP$_{\text{M}}$	66.7	50.0	5.0	66.8	52.3	56.9	26.7	48.4	77.6	73.2	76.6	77.3		
WEP+MLM	65.8	50.4	2.1	63.9	44.9	56.6	29.1	51.0	75.2	71.8	76.9	77.2		
PEP$_{\text{MRS}}$+MLM	66.1	50.9	2.4	66.6	41.4	54.5	31.1	51.8	78.4	71.4	77.1	78.7		
PEP$_{\text{M}}$+MLM	67.7	51.1	3.3	64.5	43.7	56.6	29.3	51.7	78.2	72.0	76.8	78.6		

TABLE A.1: F1-score per language on the WikiAnn test set. Results are averaged across five seeds.

MODEL			AVG	EN	FR	NL	ES	RU	ZH	HE	TR	KO	VI	EL	MR	JA	HU	BN	CEB	WAR	TL	SW	PA	MG	ILO
XLM-R^OURS	IND	A	3.5	8.2	4.7	4.4	6.5	5.3	4.6	2.5	3.1	5.1	8.5	6.3	2.7	2.3	0.9	0.1	1.4	1.2	2.8	3.7	0.2	1.9	0.1
		S	9.4	15.2	11.3	11.0	13.4	14.4	11.9	12.3	4.0	16.7	14.2	27.3	19.5	9.2	2.2	0.0	1.7	1.3	5.1	5.6	5.8	3.7	0.4
		M	2.1	3.3	2.3	2.6	3.3	3.8	4.5	2.2	2.5	2.6	5.1	2.9	1.1	2.1	0.2	0.1	1.0	1.1	1.4	1.9	0.0	1.6	0.0
	CONF	A	3.3	4.4	2.9	2.7	4.3	5.5	5.3	3.0	3.0	5.6	9.5	7.3	3.4	4.4	0.9	0.1	1.2	1.1	2.3	2.9	0.6	1.8	0.5
		S	7.5	5.2	4.4	3.6	4.9	14.2	11.8	11.4	3.9	15.9	12.6	25.6	18.9	8.8	2.0	0.0	1.4	1.4	4.4	4.3	5.8	3.5	0.5
		M	2.6	3.9	2.3	2.7	4.2	4.1	5.2	2.7	2.4	3.4	7.0	4.3	2.07	4.2	0.3	0.1	1.0	1.1	1.3	1.9	0.4	1.5	0.5
MLM 39	IND	A	2.3	2.1	3.7	2.9	3.9	2.9	1.9	3.4	1.2	5.0	4.6	4.2	3.6	0.3	0.7	0.0	2.1	1.0	1.4	5.2	0.0	0.0	0.1
		S	6.4	5.1	8.7	6.4	9.4	6.0	8.3	8.7	3.1	16.6	9.1	19.3	17.9	2.5	1.8	0.6	2.9	1.1	4.4	8.3	0.3	0.5	0.1
		M	1.3	0.9	2.0	2.0	1.9	2.0	1.8	1.9	0.6	2.3	2.4	2.8	2.1	0.2	0.4	0.0	1.8	1.0	0.5	2.2	0.0	0.0	0.1
	CONF	A	2.5	2.5	3.6	2.9	4.3	2.6	2.0	4.8	1.1	5.7	6.3	5.2	4.2	0.4	0.6	0.1	2.0	1.0	1.2	5.2	0.0	0.0	0.1
		S	5.9	4.9	7.6	5.9	9.0	4.4	7.6	7.4	2.5	16.1	8.5	17.2	16.7	2.5	1.6	0.6	2.8	1.1	3.9	7.8	0.3	0.5	0.0
		M	1.7	1.8	2.3	2.2	2.6	2.3	1.9	3.4	0.5	3.4	4.6	4.2	2.9	0.3	0.4	0.0	1.7	1.0	0.4	2.4	0.0	0.0	0.1
EN	IND	A	3.3	18.2	6.1	6.0	5.8	1.1	0.4	0.4	1.1	0.5	8.0	3.5	0.4	0.6	3.7	0.0	3.5	0.6	5.0	4.2	0.1	1.7	1.6
		S	8.5	38.3	16.4	18.7	14.9	4.4	3.4	1.4	5.6	2.7	16.8	7.3	4.1	2.5	8.5	0.0	6.9	2.6	9.7	10.3	0.0	6.9	5.4
		M	1.6	9.4	2.7	2.9	2.9	0.6	0.3	0.4	0.3	0.3	3.5	1.2	0.1	0.5	2.6	0.0	1.5	0.3	1.5	1.9	0.1	1.1	0.5
	CONF	A	3.1	16.2	6.4	5.6	5.4	1.1	0.3	0.4	1.1	0.5	7.6	3.4	0.4	0.6	3.6	0.0	3.4	0.5	4.5	4.1	0.1	1.3	1.7
		S	7.9	35.8	15.9	17.2	13.3	4.5	2.7	1.6	5.4	2.4	15.8	7.3	4.1	2.5	8.2	0.0	6.6	2.5	8.6	7.8	0.0	6.4	5.4
		M	1.5	7.5	3.3	2.9	2.9	0.6	0.3	0.4	0.2	0.3	3.6	1.1	0.1	0.5	2.6	0.0	1.5	0.2	1.5	2.0	0.1	1.0	0.6
WEP 39	IND	A	6.1	15.6	9.1	11.5	10.5	2.8	6.7	3.7	3.2	6.7	13.2	7.9	4.0	4.6	6.7	0.9	4.3	2.1	7.4	7.2	0.0	2.3	3.3
		S	19.4	36.4	24.1	30.3	25.6	14.3	18.5	34.7	12.2	31.5	23.4	36.0	29.8	17.8	18.5	6.1	8.5	5.0	16.9	21.3	0.0	5.4	9.3
		M	3.0	7.2	3.9	4.9	4.6	1.5	6.3	2.6	1.0	2.9	7.3	3.9	1.5	4.1	2.5	0.0	2.2	1.4	1.8	3.3	0.0	1.4	0.6
	CONF	A	4.9	12.1	8.2	9.6	8.8	2.4	3.1	3.3	2.9	5.9	9.3	7.4	3.5	1.9	5.6	0.8	4.1	1.7	6.8	5.7	0.0	1.8	3.3
		S	17.4	32.6	22.9	26.5	23.4	12.2	16.7	32.4	11.2	28.3	19.3	34.3	27.1	15.9	16.0	5.6	8.2	4.7	14.9	17.2	0.0	5.1	9.2
		M	2.1	4.6	3.3	3.6	3.0	1.2	2.6	2.3	0.8	2.7	3.9	3.7	1.4	1.5	1.8	0.0	2.1	1.0	1.9	2.0	0.0	1.0	0.7
93	IND	A	5.8	13.9	7.6	10.1	11.2	2.8	7.2	2.9	2.9	5.8	13.6	8.1	4.4	3.2	7.2	0.6	3.1	2.4	6.8	6.6	1.0	2.5	3.2
		S	18.5	34.5	20.0	28.9	25.3	14.0	20.1	26.0	13.0	28.6	25.4	35.0	25.6	17.3	18.2	4.9	6.7	7.2	13.6	17.6	11.6	5.7	8.3
		M	2.7	6.6	3.0	4.7	5.2	1.3	6.8	2.3	0.9	2.5	7.6	4.3	2.1	2.7	3.1	0.0	1.8	0.9	1.3	1.4	0.2	1.3	0.4
	CONF	A	4.6	11.3	6.4	8.6	9.1	2.2	2.7	2.5	2.7	4.9	10.5	7.2	3.5	1.8	6.1	0.6	2.8	2.0	6.2	5.1	0.8	1.4	2.5
		S	16.3	31.5	18.4	26.3	22.3	11.8	18.0	24.3	12.2	25.5	22.3	31.3	20.8	14.5	16.4	4.4	6.1	6.6	11.6	15.3	8.9	2.4	7.6
		M	1.8	4.7	2.1	3.5	3.4	0.9	2.3	2.0	0.8	2.1	4.9	3.5	1.5	1.4	2.2	0.0	1.6	0.7	1.1	0.9	0.1	0.6	0.4
PEP_MS 39	IND	A	4.7	15.1	6.9	11.0	9.6	5.0	3.8	3.2	2.0	7.3	9.0	5.5	3.0	3.3	1.9	0.2	3.3	1.5	5.9	5.5	0.0	0.7	0.6
		S	15.0	35.2	18.6	29.4	22.0	16.7	15.7	19.4	8.7	29.3	19.2	30.2	24.5	19.9	4.6	1.7	6.4	2.5	10.3	12.8	0.0	1.1	1.2
		M	2.4	7.1	2.4	4.5	4.2	2.1	3.5	2.6	0.7	3.3	4.4	3.5	0.5	2.7	1.0	0.0	2.0	1.2	2.4	2.8	0.0	0.5	0.4
	CONF	A	6.0	15.7	8.1	12.5	11.7	5.7	6.9	5.2	2.9	9.2	14.0	6.3	5.1	6.7	3.4	0.4	3.4	1.5	6.4	5.6	0.0	0.5	0.5
		S	13.1	31.9	17.1	27.1	19.6	12.1	13.6	17.7	7.6	26.1	16.0	27.1	21.4	16.9	3.9	1.6	5.3	2.2	9.4	9.0	0.0	0.8	1.1
		M	4.3	10.0	4.5	7.1	8.6	4.0	6.7	4.7	2.0	6.2	11.8	4.9	3.3	6.4	2.7	0.2	2.4	1.3	3.9	3.8	0.0	0.2	0.3
EN	IND	A	2.6	16.8	5.0	5.2	4.9	1.5	0.2	0.6	0.2	0.6	6.3	3.0	0.4	0.6	1.0	0.0	1.2	0.4	4.5	2.3	0.6	1.8	0.5
		S	6.5	35.5	13.4	15.5	12.3	6.2	2.0	4.7	0.5	3.3	13.0	8.7	3.1	3.0	1.9	0.0	2.5	0.5	7.0	5.3	0.7	4.5	0.4
		M	1.2	6.6	2.2	2.7	2.3	1.2	0.2	0.5	0.1	0.3	2.8	0.7	0.1	0.5	0.3	0.0	0.6	0.4	1.4	1.0	0.5	1.3	0.5
	CONF	A	2.7	18.0	5.3	5.1	4.6	1.5	0.6	1.0	0.2	0.9	6.3	3.0	0.6	0.9	0.9	0.0	1.5	0.3	4.6	2.1	0.6	1.6	0.4
		S	5.7	33.1	12.0	13.3	10.2	5.8	2.0	4.4	0.5	2.7	11.1	7.3	3.1	3.0	1.7	0.0	1.7	0.3	6.7	4.6	0.6	1.0	0.3
		M	1.6	10.4	3.0	3.2	2.6	1.2	0.5	0.9	0.1	0.8	3.8	1.2	0.3	0.8	0.3	0.0	1.1	0.3	1.8	1.1	0.5	1.2	0.5
PEP_MS+MLM 39	IND	A	4.9	14.9	9.7	10.5	10.5	7.3	5.5	4.4	1.3	7.0	9.4	5.8	2.0	2.6	1.6	0.0	2.8	0.9	4.5	6.5	0.0	0.3	0.4
		S	13.9	34.8	24.2	27.7	23.1	20.2	16.0	17.5	5.8	28.6	19.3	25.2	13.5	15.3	4.0	2.4	5.5	1.7	8.3	11.2	0.0	0.3	0.7
		M	2.4	6.3	3.4	4.6	4.2	3.8	5.2	2.7	0.5	3.0	4.6	4.1	0.6	2.1	0.7	0.0	1.7	0.8	1.9	3.1	0.0	0.3	0.2
	CONF	A	5.7	16.3	10.5	11.0	11.3	7.9	7.0	6.2	1.3	9.2	14.2	6.6	2.8	4.2	1.5	0.3	3.4	1.3	4.1	6.6	0.0	0.3	0.3
		S	12.0	30.6	21.6	24.8	20.1	16.6	15.7	16.0	3.0	27.1	15.8	20.8	12.3	12.8	3.4	0.0	4.9	2.0	7.0	9.5	0.0	0.2	0.4
		M	3.9	9.4	5.4	6.2	6.7	5.0	6.7	4.5	0.8	6.0	11.8	5.6	1.5	3.8	0.7	0.3	2.8	1.3	2.6	4.5	0.0	0.3	0.2
93	IND	A	4.5	14.1	8.6	9.1	8.8	5.8	5.0	3.1	0.8	6.3	9.4	5.7	1.8	1.5	1.5	0.1	2.9	1.7	4.0	5.2	0.7	2.3	0.4
		S	13.2	32.7	22.0	24.4	20.3	18.6	17.0	12.8	3.8	26.9	19.4	25.9	12.2	12.0	3.4	0.9	5.0	2.1	9.7	9.5	8.1	3.1	0.9
		M	2.3	5.9	3.3	4.4	3.8	3.1	4.7	2.6	0.4	2.8	4.4	3.8	0.4	1.1	0.7	0.0	2.2	1.5	1.2	1.3	0.0	1.9	0.2
	CONF	A	5.5	15.7	9.5	9.8	9.3	6.2	6.9	5.0	0.8	8.1	13.5	7.3	2.8	3.0	1.4	0.1	2.6	1.8	4.4	6.8	2.0	2.6	0.4
		S	11.9	31.3	19.2	22.0	16.8	15.7	16.3	12.5	3.5	25.0	16.6	23.9	11.3	8.6	2.8	0.6	4.3	2.0	9.0	9.4	7.5	2.7	0.8
		M	3.8	8.7	5.2	6.0	5.5	4.2	6.6	4.5	0.5	5.4	10.9	5.9	1.7	2.7	0.7	0.1	2.3	1.8	2.3	3.7	1.5	2.6	0.2

TABLE A.2: X-FACTR results.

MODEL	EN	ES	DE	HI	FR	PT	ZH	JA	TR
XLM-R$^{\text{OURS}}$	95.6 $_{0.15}$	81.5 $_{0.71}$	79.8 $_{2.04}$	50.6 $_{5.35}$	74.8 $_{1.90}$	76.5 $_{1.14}$	77.2 $_{2.06}$	56.8 $_{4.99}$	43.0 $_{2.72}$
MLM	95.6 $_{0.16}$	78.8 $_{2.88}$	78.0 $_{2.56}$	61.5 $_{7.26}$	74.4 $_{3.18}$	74.6 $_{1.39}$	76.4 $_{1.81}$	70.3 $_{2.00}$	39.7 $_{4.09}$
WEP	95.7 $_{0.15}$	79.9 $_{1.34}$	80.3 $_{0.58}$	52.7 $_{4.15}$	75.6 $_{0.87}$	76.3 $_{0.63}$	78.1 $_{1.43}$	60.7 $_{7.07}$	40.4 $_{4.40}$
PEP$_{\text{MS}}$	95.3 $_{0.06}$	79.3 $_{2.60}$	79.7 $_{2.28}$	62.9 $_{2.30}$	75.3 $_{2.10}$	76.2 $_{1.60}$	77.8 $_{1.30}$	69.0 $_{4.90}$	45.3 $_{2.50}$
PEP$_{\text{MS}}$+MLM	95.6 $_{0.10}$	81.3 $_{1.90}$	81.4 $_{0.90}$	65.8 $_{2.20}$	78.2 $_{0.30}$	76.1 $_{1.00}$	78.8 $_{0.60}$	68.8 $_{3.30}$	42.1 $_{3.30}$

TABLE A.3: F1-score (average across five seeds) on MultiATIS++ SF-only training.

MODEL	BG	DA	ET	FA	HR	JA
XLM-R$^{\text{ours}}$	57.5 $_{1.03}$	60.6 $_{2.06}$	61.7 $_{3.23}$	62.4 $_{1.05}$	61.7 $_{2.93}$	54.0 $_{1.56}$
MLM	59.3 $_{0.85}$	59.0 $_{0.72}$	60.6 $_{1.15}$	63.5 $_{1.41}$	62.3 $_{1.87}$	52.6 $_{1.26}$
WEP	59.0 $_{1.84}$	61.3 $_{1.13}$	62.2 $_{0.74}$	64.9 $_{1.06}$	63.7 $_{2.40}$	54.7 $_{2.69}$
PEP$_{\text{MS}}$	59.4 $_{0.93}$	60.7 $_{1.03}$	64.4 $_{1.72}$	63.5 $_{1.51}$	64.2 $_{1.85}$	53.6 $_{2.60}$
PEP$_{\text{MS}}$+MLM	59.7 $_{1.04}$	60.9 $_{1.20}$	63.9 $_{1.02}$	63.1 $_{1.63}$	63.8 $_{1.89}$	53.2 $_{2.18}$
MODEL	KO	NL	ZH	DE	FR	IT
XLM-R$^{\text{ours}}$	62.4 $_{1.99}$	61.5 $_{1.94}$	56.4 $_{3.83}$	57.7 $_{1.58}$	56.4 $_{1.40}$	57.1 $_{1.35}$
MLM	63.1 $_{1.38}$	62.3 $_{0.76}$	52.4 $_{1.03}$	57.2 $_{0.54}$	56.6 $_{0.33}$	58.0 $_{1.09}$
WEP	64.6 $_{0.74}$	63.8 $_{0.77}$	55.2 $_{3.30}$	59.6 $_{1.04}$	57.0 $_{0.98}$	59.1 $_{1.03}$
PEP$_{\text{MS}}$	64.6 $_{2.88}$	63.9 $_{0.79}$	52.8 $_{3.01}$	59.5 $_{1.79}$	57.4 $_{0.64}$	58.8 $_{1.25}$
PEP$_{\text{MS}}$+MLM	62.1 $_{3.10}$	63.0 $_{1.01}$	53.2 $_{2.24}$	59.0 $_{0.90}$	57.3 $_{0.58}$	58.3 $_{1.00}$

TABLE A.4: XL-WiC test set accuracy (average across five seeds) across languages.

Appendix B

Additional Results and Analysis for WEBIE

B.1 Additional Results

We show in Table B.1 the results for non-English languages for mWEBIE when specifying the source language and using the default (English) for the mBART tokenizer. These results are from beam search without constraint Trie. We can see that specifying the source language mostly harms the performance (except French), especially for Portuguese. We hypothesise that due to the model being trained solely on English as the source token, mBART may have difficulty handling other languages.

LANGUAGE	EN as Source Language in mBART Tokenizer					XX as Source Language in mBART Tokenizer				
	P	R	F1	Empty-Pos%	Acc-N	P	R	F1	Empty-Pos%	Acc-N
FRENCH	43.27	36.13	39.38	11.89	96.19	41.29	37.73	39.43	8.56	94.87
SPANISH	41.93	34.63	37.93	12.34	96.74	40.47	36.57	38.42	8.56	95.82
PORTUGUESE	41.17	32.37	36.24	14.07	96.91	13.81	1.77	3.14	86.33	98.21
HINDI	4.28	1.62	2.35	67.38	98.64	3.69	1.69	2.31	60.62	98.43

TABLE B.1: Comparison of the zero-shot performance on mWEBIE with mBART when specifying the source language (XX) and keeping the default setting as the source language (EN). Results are with standard beam search (without the constraint Trie).

B.2 Examples of ReFinED Output

We show examples of the sentences processed by ReFinED in Table B.2. For each input sentence, ReFinED identifies the set of entities in that sentence, and outputs

Example Id	Sentence	ReFinED Output
21464177	On Thursday, British campaigning group the Environmental Investigation Agency accused Italy of trying to sabotage efforts to reform the EU ETS.	[["Thursday", None, "DATE"], ["British", Entity(wikidata_entity_id=Q145, wikipedia_entity_title=United Kingdom), "GPE"], ["Environmental Investigation Agency", Entity(wikidata_entity_id=Q1345905, wikipedia_entity_title=Environmental Investigation Agency), "ORG"], ["Italy", Entity(wikidata_entity_id=Q38, wikipedia_entity_title=Italy), "ORG"], ["EU", Entity(wikidata_entity_id=Q458, wikipedia_entity_title=European Union), "ORG"], ["ETS", Entity(wikidata_entity_id=Q899383, wikipedia_entity_title=ETSI), "ORG"]]
1274217	It culminates in the decade-long debate ending in 1913 to turn the Hetch Hetchy valley in Yosemite National Park into a reservoir for San Francisco.	[["decade-long22", None, "DATE"], ["1913", Entity(parsed_string=[timepoint: ["1913"]]), "DATE"], ["Hetch Hetchy", Entity(wikidata_entity_id=Q1616130, wikipedia_entity_title=Hetch Hetchy), "GPE"], ["Yosemite National Park", Entity(wikidata_entity_id=Q180402, wikipedia_entity_title=Yosemite National Park), "FAC"], ["San Francisco", Entity(wikidata_entity_id=Q62, wikipedia_entity_title=San Francisco), "GPE"]]

TABLE B.2: ReFinED outputs on WEBIE validation examples.

mention span, Wikidata id, and Wikipedia title for each entity. For our experiments, we use the `wikipedia_model_with_numbers` model with `wikipedia` entity set.

B.3 MTurk Annotation Details

In this section, we describe the detailed settings for annotating (m)WEBIEwith MTurk.

WEBIE

The first annotation task (HIT) is to verify the correctness of the triples automatically created from the DS approach and filtered by the NLI model. The guidance and the interface are shown in Figure B.1 and Figure B.2, respectively.

In each HIT, we provide a sentence with its entities highlighted (head entity in blue and tail entity in green) and the URL of the web page which the sentence is extracted from. For the first EL annotation job, we provide both the links to the Wikipedia and Wikidata pages. Annotators are asked to choose if the highlighted spans are linked correctly to the KB. Next, the annotators are asked to verify if a

FIGURE B.1: MTurk HIT guidance entity and relation labelling.

FIGURE B.2: MTurk HIT user interface for entity and relation labelling.

relation (highlighted in orange) can be inferred from the sentence. We provide the description of the relation and an example use case to facilitate the annotation.

Each triple is annotated by three workers, and we pay $0.2 per HIT. We hire

MTurk workers with Masters Qualification and set additional requirements including (i) having done 2,000 HITs and (ii) having a job approval rate $\geq$99%.

mWEBIE

Figure B.3 and Figure B.4 illustrates the interface for correcting machine-translated sentence and identifying corresponding entities in them. As it is challenging to find qualified crowd workers for the translation task[1], we set the geographical regions for each language to the countries where the language is one of the official languages. We find that only India and countries in America have an adequate number of MTurk workers, which highly restricts the options for our target languages. In the end, the countries we set for the target languages are as follows: Portuguese: AO, BR, CV, ST, GW, GQ, MZ; Spanish: ES, MX, CO, PE, CL; CA for French, and IN for Hindi[2]. It was also necessary to remove the Masters Qualification requirement for MTurk workers (except Hindi) to find adequate annotators. We then conduct pilot annotations, where we deliberately introduce errors in the reference machine translation to verify if the workers under our requirement settings are able to correct them.

We provide the English sentence paired with the original machine-translated sentence for the actual HIT. The English sentence is highlighted with its entity spans, and we instruct the workers to correct the translation while ensuring that the entities are correctly translated. After confirming the translation, workers are then asked to highlight the corresponding entities in the target language (in green). For negative sentences without entity spans, the longest noun phrases were highlighted instead to prevent workers from simply copying the reference translations. We pay $0.35 per HIT for positive sentences and $0.25 for negative sentences (since most sentences in negative examples have only one highlighted entity/noun phrase and it is considered an easier task).

Two MTurk workers were asked for the translation task, and an additional worker was asked to select the better translation, for which $0.10 per HIT was paid.

---

[1]Preliminary results where we include the USA for the mWEBIE annotation task indicate that MTurk workers with limited or no knowledge of the target language (or English) still accept the job, despite our specific requirement for proficiency in both English and the target language.

[2]For the mapping between country codes and countries, please refer to https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_LocaleDataStructureArticle.html.

Multilingual Translation and Mention Linking

Part 1: Translation

Sentence (English):	The 1954 Major League Baseball All-Star Game was the 21st playing of the midsummer classic between the all-stars of the American League (AL) and National League (NL), the two leagues comprising Major League Baseball.
Reference from Machine Translation:	O All-Star Game da Major League Baseball de 1954 foi o 21o jogo do clássico de meio de verão entre os All-Stars da American League (AL) e da National League (NL), as duas ligas que compõem a Major League Baseball.
Corrected Portuguese Sentence:	O All-Star Game da Major League Baseball de 1954 foi o 21o jogo do clássico de meio de verão entre os All-Stars da American League (AL) e da National League (NL), as duas ligas que compõem a Major League Baseball.

Confirm Translation

FIGURE B.3: MTurk HIT user interface for correcting the machine-translated text.

Multilingual Translation and Mention Linking

Part 1: Translation

Sentence (English):	The 1954 Major League Baseball All-Star Game was the 21st playing of the midsummer classic between the all-stars of the American League (AL) and National League (NL), the two leagues comprising Major League Baseball.
Reference from Machine Translation:	O All-Star Game da Major League Baseball de 1954 foi o 21o jogo do clássico de meio de verão entre os All-Stars da American League (AL) e da National League (NL), as duas ligas que compõem a Major League Baseball.
Corrected Portuguese Sentence:	O All-Star Game da Major League Baseball de 1954 foi o 21o jogo do clássico de meio de verão entre os All-Stars da American League (AL) e da National League (NL), as duas ligas que compõem a Major League Baseball.

Edit Translation

Part 2: Entity List

Entity (English)	Selected Entity (Portuguese)	Wikidata Link	Select
Major League Baseball	O All-Star Game da Major League Baseball	Q1163715	Select
National League		Q858082	Select
Major League Baseball All-Star Game		Q1069698	Select
American League		Q465469	Select
NL		Q858082	Select
AL		Q465469	Select
Comments & Feedback (Optional):			

FIGURE B.4: MTurk HIT user interface for entity labelling in the target language.

Appendix C

Additional Results and Analysis for UMAE

C.1 Detailed Explanation Scores

For explanation generation on the VQA tasks, we evaluate the performance of beam search with the size of 5, top-k sampling with k from $\{50, 100, 200, ..., 1000\}$, and Nucleus and Typical (Meister et al., 2023) sampling, both with p from $\{0.1, 0.2, ..., 0.9\}$. We show the details of the NLG scores using different decoding strategies for explanations generated from QA→E in Table C.1, and Q→AE in Table C.2.

DATASET	DECODING	e-ViL S_E	B1	B2	B3	B4	R-L	MET.	CIDEr	SPICE	LEARNT BERTSC.
A-OKVQA	BEAM SEARCH	44.71	52.01	36.69	26.72	19.88	40.39	22.06	68.48	20.94	86.05
	TOP-K ($k = 100$)	44.34	52.56	37.06	27.06	19.72	44.45	21.58	73.44	19.38	86.27
	NUCLEUS ($p = 0.4$)	**50.82**	58.92	44.66	35.06	27.35	52.56	24.83	101.09	23.33	88.21
	TYPICAL ($p = 0.6$)	47.27	54.18	39.39	29.82	22.18	47.78	22.79	84.43	21.47	86.95
VCR	BEAM SEARCH	40.23	26.41	20.15	15.95	12.47	29.13	16.82	49.72	27.70	81.84
	TOP-K ($k = 50$)	33.19	20.98	14.89	11.18	8.33	23.65	13.72	32.73	21.99	80.31
	NUCLEUS ($p = 0.1$)	**40.27**	31.42	22.95	17.62	13.44	29.53	17.54	47.33	26.45	81.91
	TYPICAL ($p = 0.4$)	35.12	23.42	16.88	12.83	9.64	25.36	14.70	35.85	23.32	80.70
VQA-X	BEAM SEARCH	35.88	37.84	24.91	16.67	10.97	31.32	17.90	38.23	16.23	84.39
	TOP-K ($k = 50$)	33.28	38.35	23.11	14.21	8.45	29.15	17.05	32.89	15.26	83.41
	NUCLEUS ($p = 0.1$)	**40.67**	47.56	31.44	21.47	14.63	35.12	20.29	50.35	19.13	85.40
	TYPICAL ($p = 0.5$)	36.31	40.85	25.57	16.82	11.14	31.08	18.15	39.71	16.62	83.93

TABLE C.1: Explanation scores with automatic NLG for generated explanations (QA→E) from UMAE_ALL model with different decoding strategies. B1 B4 correspond to BLEU1 BLEU4, R-L means ROUGE-L and MET. means METEOR. The last two rows (with blue shadow) indicate out-of-domain performance.

DATASET	DECODING	e-ViL S_{E}	N-GRAM SCORES								LEARNT BERTSC.
			B1	B2	B3	B4	R-L	MET.	CIDEr	SPICE	
A-OKVQA	BEAM SEARCH	**47.01**	54.75	41.39	**32.08**	**24.25**	**49.75**	**22.54**	**86.28**	**20.68**	**87.39**
	NUCLEUS ($p = 0.5$)	46.72	**55.53**	**41.63**	31.91	23.67	49.16	22.48	82.37	20.67	87.18
VCR	BEAM SEARCH	**37.02**	25.00	18.90	**14.87**	**11.54**	**27.07**	**15.66**	**38.77**	**25.03**	**80.68**
	NUCLEUS ($p = 0.1$)	35.10	**27.41**	**19.36**	14.50	10.73	26.18	15.21	34.99	21.88	80.52
VQA-X	BEAM SEARCH	38.13	39.91	26.30	17.99	12.46	31.69	19.11	42.10	18.15	84.95
	NUCLEUS ($p = 0.1$)	**39.67**	**44.92**	**28.88**	**19.04**	**12.55**	**33.08**	**20.07**	**44.28**	**19.19**	**85.21**

TABLE C.2: Explanation scores with automatic NLG for generated explanations from Q→AE with UMAE$_{\mathrm{ALL}}$ model. B1 B4 correspond to BLEU1 BLEU4, R-L means ROUGE-L and MET. means METEOR. The last two rows (with blue shadow) indicate out-of-domain performance.

C.2 Examples of Generated Explanations

Examples of the explanations generated with beam search and Nucleus sampling for A-OKVQA are shown in Figure C.2, and VCR in Figure C.1.

In general, Nucleus sampling achieves the best performance across the datasets. However, top-k and Typical sampling do not show improvement over beam search.

C.3 Examples of Issues in the Datasets

We show some of the issues in the datasets described in subsection 6.6.2.

Figure C.3 shows examples from VCR that require an understanding of the movie plot to generate answers. Figure C.4 shows examples from OK-VQA where questions and answers are subjective or ambiguous. Figure C.5 shows examples from A-OKVQA and VQA-X that either contain wrong answers, questions that do not need visual input or typos which severely impact the model generation ("house" should be "horse").
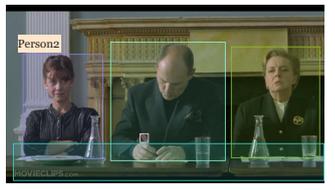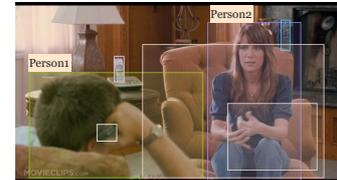
FIGURE C.1: Examples of generated answers and explanations generation for VCR.

Why is the woman wearing goggles?
Answer: protection

Ground Truth Explanations:
0) The woman is wearing goggles for protections.
1) The snow goggles this woman wears protects her eyes from the sun and other bits of debris she might encounter skiing downhill.
1) There is a lot of sun glare and snow flying up when skiing.

Explanation Generation:
Beam: The woman needs protection.
Nucleus: The woman is skiing and needs protection.

Answer + Explanation Generation:
Beam: protection, this is because The woman wants to protect her eyes.
Nucleus: protection, this is because The woman is wearing goggles to protect her eyes from the sun.

Question: What time of day is it likely right now?
Answer: morning

Ground Truth Explanations:
0) It is sunny out. there are fruit on the table, so it likely is time for breakfast.
1) The time is morning.
2) You can see the light shining in through the window and door.

Explanation Generation:
Beam: The people are using umbrellas because it's raining.
Nucleus: The people are using umbrellas to keep from getting wet.

Answer + Explanation Generation:
Beam: morning, this is because The sun is shining through the window.
Nucleus: morning, this is because There is light coming through the window.

Question: What has caused the elephants to turn brown?
Answer: dirt

Ground Truth Explanations:
0) These elephants have been rolling around in the mud.
1) The elephants are covering themselves in the naturally occurring brown substance that is near the water.
2) The elephants are dirty.

Explanation Generation:
Beam: The elephants are in mud.
Nucleus: The elephants are standing in dirt.

Answer + Explanation Generation:
Beam: mud, this is because The elephants are muddy.
Nucleus: mud, this is because The elephants are in the mud.

Question: What are the umbrellas placed in the sand to block?
Answer: sun

Ground Truth Explanations:
0) They are on a beach on a sunny day, and the umbrellas give cool and shade from the sun.
1) The umbrellas block sun.
2) Umbrellas provide shade.

Explanation Generation:
Beam: The umbrellas block the sun.
Nucleus: The umbrellas are blocking the sun.

Answer + Explanation Generation:
Beam: sun, this is because The umbrellas block the sun.
Nucleus: sun, this is because The umbrellas are placed in the sand to block the sun.

Question: Why are the people lining up?
Answer: boarding

Ground Truth Explanations:
0) The people are near a plane as if they want to get on.
1) The people are boarding.
2) The people are lined up facing the plane. people line up facing transportation vehicles when they are preparing to board.

Explanation Generation:
Beam: The people are lined up to board a plane.
Nucleus: The people are lined up to board a plane.

Answer + Explanation Generation:
Beam: board plane, this is because The people are lined up to board the plane.
Nucleus: board plane, this is because The people are boarding a plane.

Question: What is the white appliance the people are looking in being used to store?
Answer: food

Ground Truth Explanations:
0) The appliance is a fridge.
1) Perishable food is kept in this little refrigerator, which is too small for many people, but just right for one or two.
2) There are multiple edible items in the fridge already.

Explanation Generation:
Beam: The appliance is a refrigerator.
Nucleus: The appliance is a fridge.

Answer + Explanation Generation:
Beam: food, this is because The appliance is a fridge.
Nucleus: food, this is because The appliance is a fridge.

Question: Why are the people using umbrellas?
Answer: it's raining

Ground Truth Explanations:
0) There is rain and the people are being sheltered by umbrellas.
1) The people want rain protection.
2) They are protecting themselves from getting wet from the rain.

Explanation Generation:
Beam: The people are using umbrellas because it's raining.
Nucleus: The people are using umbrellas to keep from getting wet.

Answer + Explanation Generation:
Beam: rain, this is because The people are using umbrellas because it is raining.
Nucleus: it's raining, this is because The people are using umbrellas to stay dry.

Question: The layout of buildings most resembles which period?
Answer: historical european

Ground Truth Explanations:
0) The cars are on the left side of the road, so it is either the united kingdom or ireland.
1) The high pitched roofs and the stone architecture is from then
2) The layout is from the olden days.

Explanation Generation:
Beam: The buildings are old.
Nucleus: The buildings are from the late nineteenth century.

Answer + Explanation Generation:
Beam: medieval, this is because The buildings look like medieval buildings.
Nucleus: medieval, is because The buildings look like medieval buildings.

FIGURE C.2: Examples of generated answers and explanations for A-OKVQA.

Question: Why is Person 3 wearing a life jacket?

Answer Options:
0) ✔ The boat has a leak, and Person3 is scared of drowning.
1) The boat is sinking and the life jacket will hep them float.
2) Person3 is piloting the ship.
3) Person9 is wearing a life vest in case the ship sinks.

Generation: Person3 is on a boat.

Question: Why did Person1 drop Person3?
Answer: Person1 dropped Person 3 by accident.

Explanation Options:
0) ✔ Person1 can upon Person3 in the woods, and kissed her, and he dropped her off the bier.
1) Person2 is Person3's mother. Person3 is an infant and can't walk on his own.
2) Person3 is stuck in the toilet as Person1 is pulling her out.
3) Person3 is bent over and appears unsteady. Person1 looks concerned for her.

Generation: Person1 is kneeling over the body of Person3.

FIGURE C.3: Questions that require knowledge of the movie plots to generate the answers from VCR.

Is this legal or illegal?

Ground Truth Answers:
legal (6), illegal (4)

Generation: legal

Question: In which country are the transportation regulations loose enough to allow vehicles like these?

Ground Truth Answers:
india (8), china (2)

Generation: england

Ground Truth Answers:
15 minutes (4), 20 minutes (2), 25 minutes (2), minute (2)

Generation: 1 hour

Question: What nationality is this food?

Ground Truth Answers:
american (4), mediteranian (2), greek (2), asian (2)

Generation: italian

FIGURE C.4: Examples of subjective questions from OK-VQA.

Question: What country headquarters this plane company?
Answer: usa

Ground Truth Explanations:
0) The headquarters are the us.
1) The company name is virgin atlantic that was founded and has headquarters in london england.
2) The airplane has virgin atlantic livery. this company is based in england.

Question: How long does the average giraffe live?
Answer: 20-30 years

Ground Truth Explanations:
0) Giraffes can live a long time.
1) 20-30 years is the lifespan.
2) I looked up this answer on the internet since there is no way to tell the answer from the picture.

Question: What is the brown *house* doing?
Answer: walking

Ground Truth Explanations:
0) it has two legs up and two down and it is moving.
1) only two feet are touching the ground.
2) he is moving slowly on a mountain range.

FIGURE C.5: Issues in the datasets that severely impact the model generation: wrong answers (left, from A-OKVQA), questions that do not need visual input to answer (middle, from A-OKVQA), and typo (right, from VQA-X).

Appendix D

Additional Results for LLM-powered Data Augmentation

This section includes the following additional results: Table D.1, Table D.2, and Table D.3 show generated data in English with different LLMs on XCOPA, XWinograd, and XStoryCloze. Table D.4 and Table D.5 show the full result on XCOPA with Chat-GPT and GPT-4.

Fine-tuned	Train Data	LLM	AVG	EN	ET	HT	ID	IT	QU	SW	TA	TH	TR	VI	ZH
mBERT	GEN	Dolly-v2	54.0	63.4	52.0	52.2	54.0	53.8	47.6	48.6	53.4	**53.4**	52.8	50.4	58.2
		StableVicuna	53.5	62.4	51.6	49.2	55.8	55.8	50.0	**50.2**	50.2	52.6	51.0	50.4	56.0
		ChatGPT	56.0	64.8	54.8	52.6	58.0	57.4	49.8	48.4	**55.6**	52.8	53.2	53.0	59.0
		GPT-4	**58.2**	**69.2**	**59.2**	**54.0**	**60.6**	**59.2**	50.8	48.2	55.0	48.2	**53.8**	**57.6**	**61.0**
	GEN+ORI	Dolly-v2	54.4	59.8	52.6	53.2	53.0	56.4	**53.8**	**52.4**	50.4	**54.8**	49.8	52.6	58.8
		StableVicuna	55.6	65.2	53.4	50.4	59.0	60.0	51.6	50.4	49.4	52.0	52.4	54.0	58.2
		ChatGPT	54.6	59.6	56.4	**53.6**	53.8	51.4	51.4	51.6	**50.4**	52.6	**54.0**	55.0	59.2
		GPT-4	**59.3**	**72.6**	**58.8**	53.0	**62.0**	**61.0**	53.0	50.0	54.0	48.2	52.0	**57.6**	**64.6**
XLMR-Base	GEN	Dolly-v2	59.0	64.4	58.8	52.8	60.8	61.0	50.8	55.6	60.4	58.0	57.2	58.6	59.0
		StableVicuna	58.5	60.4	59.4	**53.6**	60.8	56.8	49.2	56.0	61.2	60.4	54.8	59.6	58.6
		ChatGPT	58.8	62.4	56.4	52.4	61.4	58.6	**52.2**	52.0	63.4	61.2	56.4	59.6	62.8
		GPT-4	**63.6**	**67.0**	**62.4**	52.0	**68.6**	**62.6**	51.8	**58.6**	65.4	64.8	63.2	66.6	69.6
	GEN+ORI	Dolly-v2	58.7	65.6	57.6	**52.2**	60.8	58.4	52.4	58.2	57.4	58.0	58.4	58.0	59.8
		StableVicuna	61.1	65.0	62.4	49.4	64.2	62.4	46.2	**60.4**	59.6	58.0	58.0	63.0	63.4
		ChatGPT	59.8	63.8	61.6	51.6	62.6	59.8	51.2	51.6	60.4	61.6	61.8	64.8	62.0
		GPT-4	**63.6**	**69.6**	**63.8**	51.2	**67.2**	**62.4**	**52.6**	58.4	**63.8**	**66.0**	64.2	66.8	69.4
XLMR-Large	GEN	Dolly-v2	59.6	62.4	58.6	49.6	64.8	59.2	50.6	56.8	60.8	58.8	57.0	61.0	63.0
		StableVicuna	65.7	71.4	66.2	50.4	71.4	70.2	50.0	60.0	64.0	63.6	68.0	68.2	69.8
		ChatGPT	65.2	71.2	64.6	51.6	70.8	66.6	**51.0**	58.8	66.0	68.2	69.0	68.8	68.8
		GPT-4	**73.6**	**83.2**	**71.2**	52.0	**81.2**	**78.2**	51.0	62.2	76.6	77.4	75.0	78.4	79.0
	GEN+ORI	Dolly-v2	66.4	74.2	62.8	**53.0**	72.0	70.4	46.2	61.6	65.6	66.2	69.6	67.6	70.6
		StableVicuna	69.9	76.0	69.8	51.2	75.0	74.2	51.2	64.4	70.2	71.6	72.2	72.6	75.4
		ChatGPT	69.5	76.4	69.8	48.2	76.0	72.8	50.8	63.4	67.8	70.8	70.2	73.4	77.8
		GPT-4	**73.7**	**84.6**	**70.4**	50.0	**80.8**	**80.2**	51.8	65.8	72.8	76.0	74.8	78.4	80.4

TABLE D.1: Accuracy on XCOPA with different LLM-generated English data.

Fine-tuned	Training data	LLM	AVG	EN	FR	JA	PT	RU	ZH
mBERT	GEN	Dolly-v2	**56.47**	**71.24**	53.01	52.45	**53.23**	54.92	53.97
		StableVicuna	53.73	54.94	**56.63**	50.26	50.57	52.06	57.94
		ChatGPT	56.00	54.94	54.22	**54.01**	52.09	**55.87**	64.88
		GPT-4	54.90	56.22	**56.63**	52.55	51.71	52.38	59.92
	GEN+ORI	Dolly-v2	**59.32**	71.24	57.83	**53.81**	56.65	59.05	57.34
		StableVicuna	58.46	57.94	63.86	**53.81**	**57.41**	58.41	59.33
		ChatGPT	58.26	56.65	66.27	53.60	56.27	**60.00**	56.75
		GPT-4	57.48	53.65	62.65	**54.43**	55.89	57.14	**61.11**
XLMR-Base	GEN	Dolly-v2	59.63	71.24	57.83	55.79	**57.03**	57.78	58.13
		StableVicuna	58.95	60.09	55.42	57.35	52.47	58.73	69.64
		ChatGPT	62.69	69.10	60.24	61.42	**57.03**	61.27	67.06
		GPT-4	**63.32**	69.10	**61.45**	61.52	56.65	60.95	**70.24**
	GEN+ORI	Dolly-v2	66.33	**75.54**	63.86	65.80	**64.26**	62.86	65.67
		StableVicuna	65.97	64.38	66.27	67.15	63.88	**65.71**	68.45
		ChatGPT	65.94	65.24	60.24	**68.93**	70.72	62.86	67.66
XLMR-Large	GEN	GPT-4	**66.88**	68.24	**67.47**	66.94	63.88	63.49	**71.23**
		Dolly-v2	**76.86**	87.55	67.47	**81.02**	**76.43**	74.29	74.40
		StableVicuna	68.22	74.25	63.86	68.20	66.16	63.81	73.02
		ChatGPT	73.20	81.97	66.27	73.10	66.92	72.38	78.57
		GPT-4	76.37	81.55	**74.70**	75.91	71.86	**75.24**	78.97
	GEN+ORI	Dolly-v2	83.10	**90.56**	79.52	85.19	84.03	80.95	78.37
		StableVicuna	82.02	83.26	80.72	83.84	86.31	**82.22**	75.79
		ChatGPT	83.22	85.84	80.72	**87.38**	85.93	80.95	78.50
		GPT-4	**83.52**	85.41	**81.93**	85.92	**86.69**	80.63	**80.56**

TABLE D.2: Accuracy on XWinograd with different LLM-generated English data.

Fine-tuned	Training data	LLM	AVG	EN	RU	ZH	ES	AR	HI	ID	TE	SW	EU	MY
mBERT	GEN	Dolly-v2	**68.7**	78.8	**71.3**	73.6	74.2	67.4	66.9	69.0	**65.0**	60.9	**66.8**	62.0
		StableVicuna	64.6	71.4	66.8	68.8	68.1	64.3	63.6	66.1	61.2	58.6	63.6	58.4
		ChatGPT	64.3	69.7	66.4	68.1	68.0	64.6	64.5	66.6	59.8	59.2	62.3	58.4
		GPT-4	68.0	75.5	70.8	73.3	70.4	**67.6**	**68.2**	69.6	63.1	**62.3**	65.4	**62.2**
	GEN+ORI	Dolly-v2	68.1	75.7	71.2	72.4	73.2	66.4	67.1	68.9	**64.5**	61.4	67.1	61.0
		StableVicuna	67.3	77.0	71.0	70.2	71.4	67.2	66.5	68.4	62.4	60.5	64.3	61.4
		ChatGPT	68.3	76.4	68.5	72.9	73.0	66.3	68.6	71.1	62.0	**62.0**	67.4	**63.4**
		GPT-4	**69.8**	79.5	73.1	75.3	73.4	68.1	69.8	71.9	64.1	**62.0**	68.9	61.6
XLMR-Base	GEN	Dolly-v2	75.8	81.4	79.2	80.3	78.0	73.6	74.7	80.7	73.0	68.8	72.2	71.7
		StableVicuna	69.6	72.3	71.1	71.5	70.4	68.3	70.4	72.1	68.4	65.7	68.0	67.7
		ChatGPT	67.4	69.7	68.9	68.5	68.7	66.1	68.2	68.7	67.0	63.7	65.6	66.6
		GPT-4	**74.6**	78.2	78.0	78.1	77.0	73.5	75.7	77.6	71.7	68.4	73.6	69.2
	GEN+ORI	Dolly-v2	76.5	81.5	80.0	80.5	79.4	75.1	75.0	79.6	74.5	71.5	72.3	72.6
		StableVicuna	74.2	79.2	77.4	77.8	76.4	74.0	74.5	78.2	70.2	67.6	71.7	69.6
		ChatGPT	74.5	78.0	76.6	78.8	76.2	72.9	73.9	78.9	71.5	69.6	72.3	71.0
		GPT-4	**79.3**	85.4	83.2	82.6	83.0	78.0	79.9	82.7	75.9	72.9	74.9	74.3
XLMR-Large	GEN	Dolly-v2	84.8	87.4	87.3	87.8	86.6	83.0	84.4	87.1	**84.1**	81.0	82.9	81.4
		StableVicuna	74.6	76.7	75.9	77.4	76.2	72.9	74.5	76.2	74.3	70.8	73.5	72.5
		ChatGPT	77.3	78.6	79.9	78.0	77.9	75.8	77.4	78.0	76.4	73.5	77.1	77.7
		GPT-4	**86.0**	88.5	88.2	88.2	88.0	84.9	85.7	87.8	83.7	81.3	85.6	84.3
	GEN+ORI	Dolly-v2	86.4	89.2	87.2	89.5	87.1	85.2	86.7	87.7	**85.0**	83.0	85.7	83.8
		StableVicuna	84.8	88.4	87.6	87.8	86.6	82.9	83.3	87.4	83.7	81.3	83.7	80.0
		ChatGPT	85.8	88.5	88.0	88.3	87.3	83.7	85.9	87.2	83.7	81.6	85.4	83.8
		GPT-4	**88.4**	92.3	91.5	91.5	90.5	86.4	88.4	91.1	84.8	83.1	87.4	85.2

TABLE D.3: Accuracy on XStoryCloze with different LLM-generated English data.

Model	Training Data	|Data|	AVG	EN	ET	HT	ID	IT	QU	SW	TA	TH	TR	VI	ZH
	ORI (BASELINE)	400	47.2	53.8	44.2	48.6	47.2	46.2	50.6	45.4	48.4	49.8	49.8	43.6	47.4
	GEN_{EN}	3.7k	56.0	64.8	54.8	52.6	58.0	57.4	49.8	48.4	55.6	52.8	53.2	53.0	59.0
	$GEN_{EN} + ORI$	4.1k	54.6	59.6	56.4	53.6	53.8	51.4	51.4	51.6	50.4	52.6	54.0	55.0	59.2
	$GEN_{EN} + ORI$ (TLV)	4.1k	57.6	68.0	55.4	54.0	61.2	59.8	51.8	51.2	55.8	54.4	52.2	53.4	59.2
	GEN_{EN}	28.6k	57.2	66.2	55.8	50.8	58.6	58.2	53.2	51.2	57.2	53.2	52.0	56.0	61.0
mBERT	$GEN_{EN} + ORI$	29k	57.0	66.6	55.4	51.4	59.2	58.6	52.4	50.8	53.6	53.2	50.0	54.8	62.8
	$GEN_{EN} + ORI$ (TLV)	29k	57.0	66.6	55.4	51.4	59.2	58.6	52.4	50.8	53.6	53.2	50.0	54.8	62.8
	GEN_{XX}	3.6k/lang	57.5	64.8	57.8	57.4	58.0	60.2	54.6	51.4	53.0	–	–	53.0	62.0
	$GEN_{XX} + ORI$	4k	56.8	59.6	58.8	54.6	56.2	61.2	53.6	54.6	53.6	–	–	52.0	60.2
	$GEN_{EN}^{Trans} + ORI$	4k	58.7	59.6	59.8	59.8	62.8	61.0	–	52.6	56.8	53.4	56.2	58.2	59.4
	$GEN_{EN}^{Trans} + ORI$	29k/lang	**60.6**	66.6	61.8	57.8	60.8	62.2	–	53.2	58.4	53.2	63.0	60.6	63.8
	ORI (BASELINE)	400	55.6	57.6	54.6	50.6	59.6	54.8	46.0	55.0	53.4	56.2	55.2	54.8	59.6
	GEN_{EN}	3.7k	58.8	62.4	56.4	52.4	61.4	58.6	52.2	52.0	63.4	61.2	56.4	59.6	62.8
	$GEN_{EN} + ORI$	4.1k	59.8	63.8	61.6	51.6	62.6	59.8	51.2	51.6	60.4	61.6	61.8	64.8	62.0
	$GEN_{EN} + ORI$ (TLV)	4.1k	60.7	63.2	61.6	51.4	64.8	61.2	51.2	53.6	62.6	63.0	58.2	61.0	66.6
	GEN_{EN}	28.6k	60.8	66.4	57.2	56.0	66.4	61.2	53.0	53.8	60.0	61.6	56.6	61.4	64.6
XLMR-Base	$GEN_{EN} + ORI$	29k	62.1	64.6	61.8	50.6	66.8	63.6	48.0	55.6	65.8	63.6	57.2	63.2	66.8
	$GEN_{EN} + ORI$ (TLV)	29k	60.9	66.4	61.8	49.8	66.2	59.8	54.6	53.4	62.4	63.8	58.2	62.8	65.8
	GEN_{XX}	3.6k/lang	58.8	62.4	57.0	55.6	61.4	59.0	55.6	54.4	56.8	–	–	60.6	62.0
	$GEN_{XX} + ORI$	4k	59.9	63.8	60.6	55.0	64.6	59.6	52.6	54.6	56.4	–	–	59.6	64.8
	$GEN_{EN}^{Trans} + ORI$	4k	61.1	63.8	60.0	58.0	65.0	60.8	–	53.8	60.2	66.2	56.6	62.6	66.0
	$GEN_{EN}^{Trans} + ORI$	29k/lang	**62.2**	64.6	63.2	57.2	64.8	61.2	–	55.0	61.2	59.2	59.5	64.2	68.4
	ORI (BASELINE)	400	64.4	71.4	62.8	51.4	69.0	65.8	52.0	60.6	62.0	64.0	61.2	69.4	66.6
	GEN_{EN}	3.7k	65.2	71.2	64.6	51.6	70.8	66.6	51.0	58.8	66.0	68.2	69.0	68.8	68.8
	$GEN_{EN} + ORI$	4.1k	69.5	76.4	69.8	48.2	76.0	72.8	50.8	63.4	67.8	70.8	70.2	73.4	77.8
	$GEN_{EN} + ORI$ (TLV)	4.1k	71.9	80.6	71.6	50.8	78.6	77.2	51.8	63.0	69.2	71.2	72.8	77.2	78.8
	GEN_{EN}	28.6k	71.8	80.6	74.4	51.0	78.4	75.2	51.2	63.4	69.8	70.6	69.8	75.6	77.4
XLMR-Large	$GEN_{EN} + ORI$	29k	**72.4**	81.0	73.8	54.4	80.2	75.2	48.8	61.4	70.4	73.8	70.4	75.6	79.8
	$GEN_{EN} + ORI$ (TLV)	29k	**72.4**	81.0	73.8	54.4	80.2	75.2	48.8	61.0	70.4	73.8	70.4	75.6	79.8
	GEN_{XX}	3.6k/lang	63.4	71.2	62.6	54.2	71.0	65.8	49.4	53.8	56.4	–	–	64.0	71.6
	$GEN_{XX} + ORI$	4k	65.2	76.4	62.4	55.2	75.0	62.2	54.0	58.2	55.4	–	–	66.2	76.2
	$GEN_{EN}^{Trans} + ORI$	4k	67.0	76.4	60.0	59.6	66.2	66.6	–	59.0	64.8	71.2	65.2	74.8	75.6
	$GEN_{EN}^{Trans} + ORI$	29k/lang	71.5	81.0	71.8	57.2	79.8	74.4	–	54.8	71.4	72.6	70.0	77.2	75.6

TABLE D.4: Full results on XCOPA (with ChatGPT-generated data). +TLV corresponds to including the original validation set in all Target Languages in the Validation set. Rows are sorted by the number of instances used in training. AVG shows average results for languages that are available in all settings (excl. QU, TH, TR).

Model	Training Data	AVG	EN	ET	HT	ID	IT	QU	SW	TA	TH	TR	VI	ZH
mBERT	*ORI*	47.2	53.8	44.2	48.6	47.2	46.2	50.6	45.4	48.4	49.8	49.8	43.6	47.4
	GEN_{EN}	58.2	69.2	59.2	54.0	60.6	59.2	50.8	48.2	55.0	48.2	53.8	57.6	61.0
	$GEN_{EN}+ORI$	59.3	72.6	58.8	53.0	62.0	61.0	53.0	50.0	54.0	48.2	52.0	57.6	64.6
	GEN_{XX}	60.2	69.2	59.4	56.2	60.2	63.8	54.4	55.2	54.0	–	–	61.2	62.2
	$GEN_{XX}+ORI$	61.8	72.6	61.2	58.2	62.2	66.4	54.4	57.4	53.4	–	–	63.0	61.8
	GEN_{EN}^{Trans}	61.4	69.2	59.2	56.8	65.4	65.2	–	53.4	56.8	52.6	59.6	61.8	65.0
	$GEN_{EN}^{Trans}+ORI$	62.6	72.6	58.6	55.2	65.6	65.4	–	53.8	62.6	53.2	58.8	64.6	65.4
XLMR-Base	*ORI*	55.6	57.6	54.6	50.6	59.6	54.8	46.0	55.0	53.4	56.2	55.2	54.8	59.6
	GEN_{EN}	63.6	67.0	62.4	52.0	68.6	62.6	51.8	58.6	65.4	64.8	63.2	66.6	69.6
	$GEN_{EN}+ORI$	63.6	69.6	63.8	51.2	67.2	62.4	52.6	58.4	63.8	66.0	64.2	66.8	69.4
	GEN_{XX}	63.2	67.0	60.8	56.4	68.6	62.4	57.4	58.2	60.2	–	–	64.6	70.4
	$GEN_{XX}+ORI$	64.0	69.6	62.2	56.2	68.6	63.8	56.8	57.8	61.2	–	–	66.8	70.0
	GEN_{EN}^{Trans}	62.5	67.0	60.0	55.6	66.0	62.4	–	58.0	60.4	64.4	64.6	64.0	68.8
	$GEN_{EN}^{Trans}+ORI$	63.9	69.6	61.6	56.6	68.4	65.2	–	58.2	60.2	68.0	62.6	66.0	69.6
XLMR-Large	*ORI*	64.4	71.4	62.8	51.4	69.0	65.8	52.0	60.6	62.0	64.0	61.2	69.4	66.8
	GEN_{EN}	73.6	83.2	71.2	52.0	81.2	78.2	51.0	62.2	76.6	77.4	75.0	78.4	79.0
	$GEN_{EN}+ORI$	73.7	84.6	70.4	50.0	80.8	80.2	51.8	65.8	72.8	76.0	74.8	78.4	80.4
	GEN_{XX}	72.8	83.2	75.2	55.2	78.4	76.0	52.4	63.0	68.2	–	–	77.8	78.6
	$GEN_{XX}+ORI$	74.6	84.6	77.0	56.0	82.2	77.0	56.0	65.0	73.8	–	–	76.2	80.0
	GEN_{EN}^{Trans}	71.0	83.2	72.4	55.6	79.4	78.2	–	60.6	67.8	77.8	72.6	64.0	77.4
	$GEN_{EN}^{Trans}+ORI$	74.1	84.6	74.2	57.2	82.0	77.4	–	62.2	75.0	75.2	72.8	74.4	79.6

TABLE D.5: Accuracy on XCOPA. GEN_{EN} and GEN_{XX} represents 3.7K and 3.6K data in English and target languages generated by GPT-4. AVG shows average results for languages that are available in all settings (excl. QU, TH, TR).

Bibliography

David Ifeoluwa Adelani et al. (2021). *MasakhaNER: Named Entity Recognition for African Languages*. In: *Transactions of the Association for Computational Linguistics* 9. Ed. by Brian Roark and Ani Nenkova, pp. 1116–1131.

Roee Aharoni, Shashi Narayan, Joshua Maynez, Jonathan Herzig, Elizabeth Clark, and Mirella Lapata (July 2023). *Multilingual Summarization with Factual Consistency Evaluation*. In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 3562–3591.

Kabir Ahuja et al. (Dec. 2023). *MEGA: Multilingual Evaluation of Generative AI*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 4232–4267.

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan (Nov. 2018). *Where is Your Evidence: Improving Fact-checking by Justification Modeling*. In: *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, pp. 85–90.

Hunt Allcott and Matthew Gentzkow (May 2017). *Social Media and Fake News in the 2016 Election*. In: vol. 31. 2, pp. 211–36.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould (2016). *SPICE: Semantic Propositional Image Caption Evaluation*. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 382–398.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018). *Bottom-up and Top-down Attention for Image Captioning and Cisual Question Answering*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. (2023). *PaLM 2 Technical Report*. In: *ArXiv*.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen (Nov. 2021). *MAD-G: Multilingual Adapter Generation for Efficient Cross-Lingual Transfer*. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4762–4781.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh (2015). *Vqa: Visual Question Answering*. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama (July 2020). *On the Cross-lingual Transferability of Monolingual Representations*. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4623–4637.

Mikel Artetxe and Holger Schwenk (2019). *Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond*. In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi (2024). *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*. In: *The Twelfth International Conference on Learning Representations*.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. (2023). *Factuality Challenges in the Era of Large Language Models*. In: *ArXiv*.

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni (July 2022). *ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking*. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, pp. 209–220.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). *Layer Normalization*. In: *ArXiv*.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong (Apr. 2021). *Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees*. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, pp. 3011–3020.

Kelsey Ball and Dan Garrette (Oct. 2018). *Part-of-Speech Tagging for Code-Switched, Transliterated Texts without Explicit Language Identification*. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3084–3089.

Satanjeev Banerjee and Alon Lavie (June 2005). *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72.

Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani (Nov. 2019). *Normalization of Indonesian-English Code-Mixed Twitter Data*. In: *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 417–424.

Lisa Bauer and Mohit Bansal (Apr. 2021). *Identify, Align, and Integrate: Matching Knowledge Graphs to Commonsense Reasoning Tasks*. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, pp. 2259–2272.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell (2021). *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?* In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623.

Stella Biderman et al. (July 2023a). *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. In: *Proceedings of the 40th International Conference on*

Machine Learning. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 2397–2430.

Stella Biderman et al. (July 2023b). *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 2397–2430.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko (2013). *Translating Embeddings for Modeling Multi-relational Data*. In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger. Vol. 26. Curran Associates, Inc.

Sebastian Borgeaud et al. (July 2022). *Improving Language Models by Retrieving from Trillions of Tokens*. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 2206–2240.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning (Sept. 2015). *A Large Annotated Corpus for Learning Natural Language Inference*. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 632–642.

Faeze Brahman, Baolin Peng, Michel Galley, Sudha Rao, Bill Dolan, Snigdha Chaturvedi, and Jianfeng Gao (Dec. 2022). *Grounded Keys-to-Text Generation: Towards Factual Open-Ended Generation*. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 7397–7413.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020a). *Language Models are Few-shot Learners*. In: *Advances in neural information processing systems* 33, pp. 1877–1901.

Tom Brown et al. (2020b). *Language Models are Few-Shot Learners*. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 1877–1901.

Ozan Caglayan, Pranava Madhyastha, and Lucia Specia (Dec. 2020). *Curious Case of Language Generation Evaluation Metrics: A Cautionary Tale*. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 2322–2328.

Iacer Calixto, Alessandro Raganato, and Tommaso Pasini (June 2021). *Wikipedia Entities as Rendezvous across Languages: Grounding Multilingual Language Models by Predicting Wikipedia Hyperlinks*. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 3651–3661.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu (Aug. 2021). *Knowledgeable or Educated Guess? Revisiting Language Models as Knowledge Bases*. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 1860–1874.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni (2021). *Autoregressive Entity Retrieval*. In: *International Conference on Learning Representations*.

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu (Nov. 2016). *Challenges of Computational Processing of Code-Switching*. In: *Proceedings of the Second Workshop on Computational Approaches to Code Switching*. Ed. by Mona Diab, Pascale Fung, Mahmoud Ghoneim, Julia Hirschberg, and Thamar Solorio. Austin, Texas: Association for Computational Linguistics, pp. 1–11.

Arun Chaganty, Ashwin Paranjape, Percy Liang, and Christopher D. Manning (Sept. 2017). *Importance Sampling for Unbiased On-demand Evaluation of Knowledge Base Population*. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1038–1048.

Chakraborty, Tanmoy, Md. Shad Akhtar, Kai Shu, H. Russell Bernard, Maria Liakata, Preslav Nakov, and Aseem Srivastava, eds. (May 2022). *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*. Dublin, Ireland: Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae (May 2020). *Corpus Creation for Sentiment Analysis in Code-Mixed Tamil-English Text*. English. In: *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*. Marseille, France: European Language Resources association, pp. 202–210. ISBN: 979-10-95546-35-1.

Ting-Yun Chang, Yang Liu, Karthik Gopalakrishnan, Behnam Hedayatnia, Pei Zhou, and Dilek Hakkani-Tur (Nov. 2020). *Incorporating Commonsense Knowledge Graph in Pretrained Models for Social Commonsense Tasks*. In: *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Ed. by Eneko Agirre, Marianna Apidianaki, and Ivan Vulić. Online: Association for Computational Linguistics, pp. 74–79.

Xiaojun Chen, Shengbin Jia, and Yang Xiang (2020). *A review: Knowledge Reasoning Over Knowledge Graph*. In: *Expert Systems with Applications* 141, p. 112948. ISSN: 0957-4174.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu (2020). *Uniter: Universal Image-Text Representation Learning*. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 104–120.

Alexander Chernyavskiy and Dmitry Ilvovsky (2020). *Recursive Neural Text Classification Using Discourse Tree Structure for Argumentation Mining and Sentiment Analysis Tasks*. In: *ISMIS*. Springer, pp. 90–101.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou (June 2021a). *InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training*. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 3576–3588.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei (Aug. 2021b). *Improving Pretrained Cross-Lingual Language Models via Self-Labeled Word Alignment*. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 3418–3430.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal (July 2021). *Unifying Vision-and-Language Tasks via Text Generation*. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 1931–1942.

Aakanksha Chowdhery et al. (2023). *PaLM: Scaling Language Modeling with Pathways*. In: *Journal of Machine Learning Research* 24.240, pp. 1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. (2022). *Scaling Instruction-Finetuned Language Models*. In: *ArXiv*.

Elizabeth Clark, Shruti Rijhwani, Sebastian Gehrmann, Joshua Maynez, Roee Aharoni, Vitaly Nikolaev, Thibault Sellam, Aditya Siddhant, Dipanjan Das, and Ankur Parikh (Dec. 2023). *SEAHORSE: A Multilingual, Multifaceted Dataset for Summarization Evaluation*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 9397–9413.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning (2020). *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. In: *ICLR*.

Kevin Clark and Christopher D. Manning (Aug. 2016). *Improving Coreference Resolution by Learning Entity-Level Distributed Representations*. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 643–653.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer,

and Veselin Stoyanov (July 2020a). *Unsupervised Cross-lingual Representation Learning at Scale*. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451.

Alexis Conneau and Guillaume Lample (2019). *Cross-lingual Language Model Pretraining*. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov (Oct. 2018). *XNLI: Evaluating Cross-lingual Sentence Representations*. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2475–2485.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov (July 2020b). *Emerging Cross-lingual Structure in Pretrained Language Models*. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6022–6034.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. (2022). *No Language Left Behind: Scaling Human-Centered Machine Translation*. In: *ArXiv*.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. (2023). *AugGPT: Leveraging ChatGPT for Text Data Augmentation*. In: *ArXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186.

Shizhe Diao, Tianyang Xu, Ruijia Xu, Jiawei Wang, and Tong Zhang (July 2023). *Mixture-of-Domain-Adapters: Decoupling and Injecting Domain Knowledge to Pretrained Language Models' Memories*. In: *Proceedings of the 61st Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers). Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 5113–5129.

Jia Ding, Yongjun Hu, and Huiyou Chang (2020). *BERT-Based Mental Model, a Better Fake News Detector*. In: *Proceedings of the 6th International Conference on Computing and Artificial Intelligence (ICCAI)*. Tianjin, China: Association for Computing Machinery, pp. 396–400.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith (2020). *Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping*. In: *ArXiv*.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio (Aug. 2021). *A Survey of Code-switching: Linguistic and Social Perspectives for Language Technologies*. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 1654–1666.

Alexey Dosovitskiy et al. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. In: *International Conference on Learning Representations*.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos (Nov. 2019). *Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks*. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1192–1197.

Nan Du et al. (July 2022). *GLaM: Efficient Scaling of Language Models with Mixture-of-Experts*. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 5547–5569.

Radhika Dua, Sai Srinivas Kancheti, and Vineeth N Balasubramanian (2021). *Beyond vqa: Generating Multi-Word Answers and Rationales to Visual Questions*. In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1623–1632.

Markus Eberts and Adrian Ulges (2020). *Span-based Joint Entity and Relation Extraction with Transformer Pre-training*. In: *ECAI*, pp. 2006–2013.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl (May 2018). *T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples*. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

Denis Emelin, Daniele Bonadiman, Sawsan Alqahtani, Yi Zhang, and Saab Mansour (Dec. 2022). *Injecting Domain Knowledge in Language Models for Task-oriented Dialogue Systems*. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 11962–11974.

Mahmood Farokhian, Vahid Rafe, and Hadi Veisi (2023). *Fake News Detection Using Parallel BERT Deep Neural Networks*. In: *Multimedia Tools and Applications*. ISSN: 1573-7721.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang (May 2022). *Language-agnostic BERT Sentence Embedding*. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 878–891.

Paolo Ferragina and Ugo Scaiella (2010). *TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities)*. In: *Proceedings of the 19th ACM international conference on Information and Knowledge Management*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch (June 2013). *PPDB: The Paraphrase Database*. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 758–764.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen (Dec. 2023). *Enabling Large Language Models to Generate Text with Citations*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor,

Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 6465–6488.

Nicolas Gontier, Siva Reddy, and Christopher Pal (2022). *Does Entity Abstraction Help Generative Transformers Reason?* In: *Transactions on Machine Learning Research.*

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (July 2017). *Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6904–6913.

Milan Gritta and Ignacio Iacobacci (Aug. 2021). *XeroAlign: Zero-shot Cross-lingual Transformer Alignment*. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 371–381.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. (2024). *Olmo: Accelerating the science of language models*. In: *arXiv preprint arXiv:2402.00838*.

Adam Grycner and Gerhard Weikum (Nov. 2016). *POLY: Mining Relational Paraphrases from Multilingual Sentences*. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2183–2192.

Liangke Gui, Borui Wang, Qiuyuan Huang, Alexander Hauptmann, Yonatan Bisk, and Jianfeng Gao (July 2022). *KAT: A Knowledge Augmented Transformer for Vision-and-Language*. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 956–968.

John J Gumperz (1977). *The Sociolinguistic Significance of Conversational Code-switching*. In: *RELC journal* 8.2, pp. 1–34.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya (Nov. 2020). *A Semi-supervised Approach to Generate the Code-Mixed Text using Pre-trained Encoder and Transfer Learning*. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 2267–2280.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang (July 2020). *Retrieval Augmented Language Model Pre-Training*. In: *Proceedings of the 37th*

International Conference on Machine Learning. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 3929–3938.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun (Oct. 2018). *FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation*. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 4803–4809.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (June 2016a). *Deep Residual Learning for Image Recognition*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016b). *Deep Residual Learning for Image Recognition*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Benjamin Heinzerling and Kentaro Inui (Apr. 2021). *Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries*. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, pp. 1772–1791.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi (Nov. 2021). *CLIPScore: A Reference-free Evaluation Metric for Image Captioning*. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 7514–7528.

Sepp Hochreiter and Jürgen Schmidhuber (1997). *Long Short-term Memory*. In: *Neural computation* 9.8, pp. 1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi (2020). *The Curious Case of Neural Text Degeneration*. In: *International Conference on Learning Representations*.

Benjamin Horne and Sibel Adali (May 2017). *This Just In: Fake News Packs A Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire Than Real*

News. In: *Proceedings of the International AAAI Conference on Web and Social Media* 11.1, pp. 759–766.

Yifan Hou, Wenxiang Jiao, Meizhen Liu, Carl Allen, Zhaopeng Tu, and Mrinmaya Sachan (Dec. 2022). *Adapters for Enhanced Modeling of Multilingual Knowledge and Text*. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 3902–3917.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (June 2019). *Parameter-Efficient Transfer Learning for NLP*. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 2790–2799.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen (2022). *LoRA: Low-Rank Adaptation of Large Language Models*. In: *International Conference on Learning Representations*.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig (June 2021). *Explicit Alignment Objectives for Multilingual Bidirectional Encoders*. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 3633–3643.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson (July 2020). *XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation*. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 4411–4421.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin (2023). *LoraHub: Efficient Cross-task Generalization via Dynamic Lora Composition*. In: *ArXiv*.

Kuan-Hao Huang, Wasi Ahmad, Nanyun Peng, and Kai-Wei Chang (Nov. 2021). *Improving Zero-Shot Cross-Lingual Transfer Learning via Robust Training*. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 1684–1697.

Pere-Lluís Huguet Cabot and Roberto Navigli (Nov. 2021). *REBEL: Relation Extraction By End-to-end Language Generation*. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 2370–2381.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave (2023). *Atlas: Few-shot Learning with Retrieval Augmented Language Models*. In: *Journal of Machine Learning Research* 24.251, pp. 1–43.

Bin Ji, Jie Yu, Shasha Li, Jun Ma, Qingbo Wu, Yusong Tan, and Huijun Liu (Dec. 2020). *Span-based Joint Entity and Relation Extraction with Attention-based Span-specific and Contextual Semantic Representations*. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 88–99.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig (2021). *Scaling Up Visual and Vision-Language Representation Learning with Noisy Text Supervision*. In: *International conference on machine learning*. PMLR, pp. 4904–4916.

Ming Jiang, Qiuyuan Huang, Lei Zhang, Xin Wang, Pengchuan Zhang, Zhe Gan, Jana Diesner, and Jianfeng Gao (Nov. 2019). *TIGEr: Text-to-Image Grounding for Image Caption Evaluation*. 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 2141–2152.

Xiaoze Jiang, Yaobo Liang, Weizhu Chen, and Nan Duan (June 2022). *XLM-K: Improving Cross-Lingual Language Model Pre-training with Multilingual Knowledge*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 10, pp. 10840–10848.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig (Nov. 2020). *X-FACTR: Multilingual Factual Knowledge Retrieval from Pretrained Language Models*. In: *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, pp. 5943–5959.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick (July 2017). *CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Melvin Johnson et al. (2017). *Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation*. In: *Transactions of the Association for Computational Linguistics* 5, pp. 339–351.

Mandar Joshi, Kenton Lee, Yi Luan, and Kristina Toutanova (2020). *Contextualized Representations Using Textual Encyclopedic Knowledge*. In: *ArXiv*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury (July 2020). *The State and Fate of Linguistic Diversity and Inclusion in the NLP World*. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6282–6293.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West (July 2022). *GenIE: Generative Information Extraction*. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 4626–4643.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. (2022). *Language Models (Mostly) Know What They Know*. In: *ArXiv*.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang (2021). *FakeBERT: Fake News Detection in Social Media with a BERT-based Deep Learning Approach*. In: *Multimedia tools and applications* 80.8, pp. 11765–11788.

Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz (Oct. 2021). *E-ViL: A Dataset and Benchmark for Natural Language Explanations in Vision-Language Tasks*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1244–1254.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis (2020). *Generalization through Memorization: Nearest Neighbor Language Models*. In: *International Conference on Learning Representations*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury (July 2020). *GLUECoS: An Evaluation Benchmark for Code-Switched NLP*. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 3575–3585.

Jitin Krishnan, Antonios Anastasopoulos, Hemant Purohit, and Huzefa Rangwala (Nov. 2021). *Multilingual Code-Switching for Zero-Shot Cross-Lingual Intent Prediction and Slot Filling*. In: *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 211–223.

Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury (July 2022). *"Diversity and Uncertainty in Moderation" are the Key to Data Selection for Multilingual Few-shot Transfer*. In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, pp. 1042–1055.

Tom Kwiatkowski et al. (2019). *Natural Questions: A Benchmark for Question Answering Research*. In: *Transactions of the Association for Computational Linguistics* 7, pp. 452–466.

Tuan Manh Lai, ChengXiang Zhai, and Heng Ji (2023). *KEBLM: Knowledge-Enhanced Biomedical Language Models*. In: *Journal of Biomedical Informatics* 143, p. 104392. ISSN: 1532-0464.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. In: *International Conference on Learning Representations*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš (Nov. 2020). *From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4483–4499.

Grandee Lee, Xianghu Yue, and Haizhou Li (2019). *Linguistically Motivated Parallel Data Augmentation for Code-Switch Language Modeling*. In: *Proc. Interspeech 2019*, pp. 3730–3734.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He (2018). *Stacked Cross Attention for Image-text Matching*. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 201–216.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro (2022). *Factuality Enhanced Language Models for Open-Ended Text Generation*. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., pp. 34586–34599.

Hector Levesque, Ernest Davis, and Leora Morgenstern (2012). *The Winograd Schema Challenge*. In: *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (July 2020). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7871–7880.

Patrick Lewis et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 9459–9474.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad (Apr. 2021). *MTOP: A Comprehensive Multilingual Task-Oriented Semantic Parsing Benchmark*. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 2950–2962.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang (2019). *VisualBERT: A Simple and Performant Baseline For Vision and Language*. In: *ArXiv*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. (2020). *Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks*. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 121–137.

Chin-Yew Lin and Franz Josef Och (July 2004). *Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics*. In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, pp. 605–612.

Chu-Cheng Lin, Xinyi Wang, Jonathan H. Clark, Yun Zhu, Han Lu, Chenxi Whitehouse, and Hongkun Yu (2024). *Multitask Multilingual Model Adaptation with Featurized Low-Rank Mixtures*. In: *ArXiv*.

Yu-Hsiang Lin et al. (July 2019). *Choosing Transfer Languages for Cross-Lingual Learning*. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3125–3135.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). *Microsoft Coco: Common Objects in Context*. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 740–755.

Xi Victoria Lin et al. (Dec. 2022). *Few-shot Learning with Multilingual Generative Language Models*. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 9019–9052.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu (Feb. 2015). *Learning Entity and Relation Embeddings for Knowledge Graph Completion*. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 29.1.

Chao Liu, Xinghua Wu, Min Yu, Gang Li, Jianguo Jiang, Weiqing Huang, and Xiang Lu (2019). *A Two-Stage Model Based on BERT for Short Fake News Detection*. In: *Knowledge Science, Engineering and Management*. Ed. by Christos Douligeris, Dimitris Karagiannis, and Dimitris Apostolou. Cham: Springer International Publishing, pp. 172–183.

Haokun Liu, William Huang, Dhara Mungra, and Samuel R. Bowman (Nov. 2020). *Precise Task Formalization Matters in Winograd Schema Evaluations*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 8275–8280.

Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen (2023). *RETA-LLM: A Retrieval-Augmented Large Language Model Toolkit*. In: *ArXiv*.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao (Aug. 2021). *MulDA: A Multilingual Data Augmentation Framework for Low-Resource Cross-Lingual NER*. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 5834–5846.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang (Apr. 2020). *K-BERT: Enabling Language Representation with Knowledge Graph*. In: 34.03, pp. 2901–2908.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. In: *ArXiv*.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh (July 2019). *Barack's Wife Hillary: Using Knowledge Graphs for Fact-Aware Language Modeling*. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 5962–5971.

Ilya Loshchilov and Frank Hutter (2019). *Decoupled Weight Decay Regularization*. In: *International Conference on Learning Representations*.

Holy Lovenia, Samuel Cahyawijaya, Genta Indra Winata, Peng Xu, Xu Yan, Zihan Liu, Rita Frieske, Tiezheng Yu, Wenliang Dai, Elham J Barezi, et al. (2022). *ASCEND: A Spontaneous Chinese-English Dataset for Code-switching in Multi-turn Conversation*. In: *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). *Vilbert: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks*. In: *Advances in neural information processing systems* 32.

Fuli Luo, Wei Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, and Luo Si (Aug. 2021). *VECO: Variable and Flexible Cross-lingual Pre-training for Language Understanding and Generation*. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 3980–3994.

Dau-Cheng Lyu, Tien-Ping Tan, Eng-Siong Chng, and Haizhou Li (2015). *Mandarin–English code-switching speech corpus in South-East Asia: SEAME*. In: *Language Resources and Evaluation* 49.3, pp. 581–600. ISSN: 1574-0218.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan (Dec. 2023). *Query Rewriting in Retrieval-Augmented Large Language Models*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 5303–5315.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun (Dec. 2023). *Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!* In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 10572–10601.

Pranava Madhyastha, Josiah Wang, and Lucia Specia (July 2019). *VIFIDEL: Evaluating the Visual Fidelity of Image Descriptions*. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 6539–6550.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi (2019). *OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge*. In: *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 3195–3204.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald (July 2020). *On Faithfulness and Factuality in Abstractive Summarization*. In: *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 1906–1919.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell (2023). *Locally Typical Sampling*. In: *Transactions of the Association for Computational Linguistics* 11, pp. 102–121.

Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa (Nov. 2019). *KnowledgeNet: A Benchmark Dataset for Knowledge Base Population*. 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 749–758.

George A. Miller (1993). *WORDNET: A Lexical Database for English*. In: *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky (Aug. 2009). *Distant Supervision for Relation Extraction without Labeled Data*. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, pp. 1003–1011.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein (2019). *Fake News Detection on Social Media Using Geometric Deep Learning*. In: *ArXiv*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen (June 2016). *A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories*. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 839–849.

Niklas Muennighoff et al. (July 2023). *Crosslingual Generalization through Multitask Finetuning*. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 15991–16111.

Shashi Narayan, Gonçalo Simões, Yao Zhao, Joshua Maynez, Dipanjan Das, Michael Collins, and Mirella Lapata (May 2022). *A Well-Composed Text is Half Done! Composition Sampling for Diverse Conditional Generation*. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 1319–1339.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald (2021). *Planning with Learned Entity Prompts for Abstractive Summarization*. In: *Transactions of the Association for Computational Linguistics* 9, pp. 1475–1492.

Roberto Navigli and Simone Paolo Ponzetto (2012). *BabelNet: The Automatic Construction, Evaluation and Application of A Wide-coverage Multilingual Semantic Network*. In: *Artificial Intelligence* 193, pp. 217–250. ISSN: 0004-3702.

Tapas Nayak and Hwee Tou Ng (Apr. 2020). *Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction*. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05, pp. 8528–8535.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend (July 2023). *DisentQA: Disentangling Parametric and Contextual Knowledge with Counterfactual Question Answering*. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 10056–10070.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein (Nov. 2020). *Zero-Shot Cross-Lingual Transfer with Meta Learning*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4547–4562.

Jeppe Nørregaard, Benjamin D. Horne, and Sibel Adali (July 2019). *NELA-GT-2018: A Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles*.

In: *Proceedings of the International AAAI Conference on Web and Social Media* 13.01, pp. 630–638.

Cathal O'Connor and Michelle Murphy (2020). *Going Viral: Doctors Must Tackle Fake News in the Covid-19 Pandemic*. In: *Bmj* 369.10.1136.

Robert Ormandi, Mohammad Saleh, Erin Winter, and Vinay Rao (2021). *Webred: Effective Pretraining and Finetuning for Relation Extraction on the Web*. In: *ArXiv*.

Ray Oshikawa, Jing Qian, and William Yang Wang (May 2020). *A Survey on Natural Language Processing for Fake News Detection*. English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, pp. 6086–6093. ISBN: 979-10-95546-34-4.

Long Ouyang et al. (2022). *Training Language Models to Follow Instructions with Human Feedback*. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., pp. 27730–27744.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang (Nov. 2021). *ERNIE-M: Enhanced Multilingual Representation by Aligning Cross-lingual Semantics with Monolingual Corpora*. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 27–38.

Jeff Z Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, et al. (2023). *Large Language Models and Knowledge Graphs: Opportunities and Challenges*. In: *ArXiv*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji (July 2017). *Cross-lingual Name Tagging and Linking for 282 Languages*. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1946–1958.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach (June 2018). *Multimodal Explanations: Justifying Decisions and Pointing to the Evidence*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8779–8788.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez (2023). *Gorilla: Large language Model Connected with Massive APIs*. In: *ArXiv*.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith (Nov. 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 43–54.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller (Nov. 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 2463–2473.

Fabio Petroni et al. (June 2021). *KILT: a Benchmark for Knowledge Intensive Language Tasks*. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 2523–2544.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder (Nov. 2020). *MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 7654–7673.

Mohammad Taher Pilehvar and Jose Camacho-Collados (June 2019). *WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations*. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1267–1273.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen (Nov. 2020). *XCOPA: A Multilingual Dataset for Causal Commonsense Reasoning*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 2362–2376.

Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen (Sept. 2019). *Modeling Language Variation and Universals: A Survey on Typological Linguistics for Natural Language Processing*. In: *Computational Linguistics* 45.3, pp. 559–601.

Archiki Prasad, Mohammad Ali Rehan, Shreya Pathak, and Preethi Jyothi (Nov. 2021). *The Effectiveness of Intermediate-Task Training for Code-Switched Natural Language Understanding*. In: *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 176–190.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali (July 2018). *Language Modeling for Code-Mixing: The Role of Linguistic Theory based Synthetic Data*. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1543–1553.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Mani Vegupatti, and John P. McCrae (2020). *Named Entity Recognition for Code-Mixed Indian Corpus using Meta Embedding*. In: *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 68–72.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che (July 2020). *CoSDA-ML: Multi-Lingual Code-Switching Data Augmentation for Zero-Shot Cross-Lingual NLP*. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization, pp. 3853–3860.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). *Language Models are Unsupervised Multitask Learners*. In: *OpenAI Technical Report*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020a). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. In: *Journal of Machine Learning Research* 21.140, pp. 1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020b). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. In: *Journal of Machine Learning Research* 21.140, pp. 1–67.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar (Nov. 2020). *XL-WiC: A Multilingual Benchmark for Evaluating Semantic Contextualization*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 7193–7206.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das (Aug. 2021). *Increasing Faithfulness in Knowledge-Grounded Dialogue with Controllable Features*. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, pp. 704–718.

Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka (May 2022). *mLUKE: The Power of Entity Representations in Multilingual Pretrained Language Models*. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 7316–7330.

Sebastian Riedel, Limin Yao, and Andrew McCallum (2010). *Modeling Relations and Their Mentions without Labeled Text*. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 148–163. ISBN: 978-3-642-15939-8.

Arra'Di Nur Rizal and Sara Stymne (May 2020). *Evaluating Word Embeddings for Indonesian–English Code-Mixed Text Based on Synthetic Data*. English. In: *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*. Marseille, France: European Language Resources Association, pp. 26–35. ISBN: 979-10-95546-66-5.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram (Apr. 2021). *GCM: A Toolkit for Generating Synthetic Code-mixed Text*. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, pp. 205–211.

Adam Roberts, Colin Raffel, and Noam Shazeer (Nov. 2020). *How Much Knowledge Can You Pack Into the Parameters of a Language Model?* In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 5418–5426.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon (2011). *Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning.* In: *AAAI spring symposium: logical formalizations of commonsense reasoning*, pp. 90–95.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky (2020). *A Primer in BERTology: What We Know About How BERT Works*. In: *Transactions of the Association for Computational Linguistics* 8, pp. 842–866.

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele (2017). *Movie Description*. In: *International Journal of Computer Vision* 123.1, pp. 94–120.

Gaetano Rossiello, Md. Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, Owen Cornec, and Alfio Gliozzo (2023). *KnowGL: Knowledge Generation and Linking from Text*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell (June 2016). *Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News*. In: *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*. Ed. by Tommaso Fornaciari, Eileen Fitzpatrick, and Joan Bachenko. San Diego, California: Association for Computational Linguistics, pp. 7–17.

Natali Ruchansky, Sungyong Seo, and Yan Liu (2017). *Csi: A hybrid deep model for fake news detection*. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 797–806.

Leonard Salewski, A Koepke, Hendrik Lensch, and Zeynep Akata (2022). *CLEVR-X: A Visual Reasoning Dataset for Natural Language Explanations*. In: *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*. Springer, pp. 69–88.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. (2022). *Bloom: A 176b-Parameter Open-Access Multilingual Language Model*. In: *ArXiv*.

Tal Schuster, Roei Schuster, Darsh J. Shah, and Regina Barzilay (June 2020). *The Limitations of Stylometry for Detecting Machine-Generated Fake News*. In: *Computational Linguistics* 46.2, pp. 499–510.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi (2022). *A-OKVQA: A Benchmark for Visual Question Answering Using World Knowledge*. In: *European Conference on Computer Vision*, pp. 146–162.

Alessandro Seganti, Klaudia Firląg, Helena Skowronska, Michał Satława, and Piotr Andruszkiewicz (Apr. 2021). *Multilingual Entity and Relation Extraction Dataset and Model*. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 1946–1955.

Rico Sennrich, Barry Haddow, and Alexandra Birch (Aug. 2016). *Neural Machine Translation of Rare Words with Subword Units*. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725.

Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu (May 2019a). *Combating Fake News: A Survey on Identification and Mitigation Techniques*. In: 10.3, pp. 1–42. ISSN: 2157-6904.

Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu (Apr. 2019b). *Combating Fake News: A Survey on Identification and Mitigation Techniques*. In: *ACM Trans. Intell. Syst. Technol.* 10.3. ISSN: 2157-6904.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani (June 2018). *Self-Attention with Relative Position Representations*. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, pp. 464–468.

Sheng Shen et al. (2024). *Mixture-of-Experts Meets Instruction Tuning: A Winning Combination for Large Language Models*. In: *The Twelfth International Conference on Learning Representations*.

Kai Shu, Suhang Wang, and Huan Liu (2019). *Beyond News Contents: The Role of Social Context for Fake News Detection*. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. WSDM '19. Melbourne VIC, Australia: Association for Computing Machinery, pp. 312–320. ISBN: 9781450359405.

Renato M Silva and Tiago A Almeida (2021). *How Concept Drift can Impair the Classification of Fake News*. In: *Anais do IX Symposium on Knowledge Discovery, Mining and Learning*. SBC, pp. 121–128.

Livio Soares, Daniel Gillick, Jeremy Cole, and Tom Kwiatkowski (Dec. 2023). *NAIL: Lexical Retrieval Indices with Efficient Non-Autoregressive Decoders*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 2574–2589.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al. (2022). *AlexaTM 20b: Few-shot Learning Using a Large-Scale Multilingual Seq2seq Model*. In: *ArXiv*.

Anirudh Srinivasan, Gauri Kholkar, Rahul Kejriwal, Tanuja Ganu, Sandipan Dandapat, Sunayana Sitaram, Balakrishnan Santhanam, Somak Aditya, Kalika Bali,

and Monojit Choudhury (June 2022). *LITMUS Predictor: An AI Assistant for Building Reliable, High-Performing and Fair Multilingual NLP Systems*. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.11, pp. 13227–13229.

Vivek Srivastava and Mayank Singh (Nov. 2020). *PHINC: A Parallel Hinglish Social Media Code-Mixed Corpus for Machine Translation*. In: *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*. Online: Association for Computational Linguistics, pp. 41–49.

George Stoica, Emmanouil Antonios Platanios, and Barnabas Poczos (May 2021). *Re-TACRED: Addressing Shortcomings of the TACRED Dataset*. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.15, pp. 13843–13850.

Jinyan Su, Claire Cardie, and Preslav Nakov (2023). *Adapting Fake News Detection to the Era of Large Language Models*. In: *ArXiv*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai (2020). *VL-BERT: Pre-training of Generic Visual-Linguistic Representations*. In: *International Conference on Learning Representations*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu (2019). *ERNIE: Enhanced Representation through Knowledge Integration*. In: *ArXiv*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le (2014). *Sequence to Sequence Learning with Neural Networks*. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger. Vol. 27. Curran Associates, Inc.

Hao Tan and Mohit Bansal (Nov. 2019). *LXMERT: Learning Cross-Modality Encoder Representations from Transformers*. 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)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 5100–5111.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied (Dec. 2022). *Revisiting DocRED - Addressing the False Negative Problem in Relation Extraction*. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural*

Language Processing. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 8472–8487.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan (Aug. 2021). *Multilingual Translation from Denoising Pre-Training*. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 3450–3466.

Ishan Tarunesh, Sushil Khyalia, Vishwajeet Kumar, Ganesh Ramakrishnan, and Preethi Jyothi (Apr. 2021). *Meta-Learning for Effective Multi-task and Multilingual Modelling*. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 3600–3612.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi (Aug. 2021). *From Machine Translation to Code-Switching: Generating High-Quality Code-Switched Text*. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 3154–3169.

Ian Tenney, Dipanjan Das, and Ellie Pavlick (July 2019). *BERT Rediscovers the Classical NLP Pipeline*. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601.

James Thorne and Andreas Vlachos (Aug. 2018). *Automated Fact Checking: Task Formulations, Methods and Future Directions*. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 3346–3359.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal (June 2018). *FEVER: a Large-scale Dataset for Fact Extraction and VERification*. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 809–819.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn (2023). *Fine-tuning Language Models for Factuality*. In: *ArXiv*.

Alexey Tikhonov and Max Ryabinin (Aug. 2021). *It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning*. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 3534–3546.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023a). *Llama: Open and Efficient Foundation Language Models*. In: *ArXiv*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. (2023b). *Llama 2: Open Foundation and Fine-tuned Chat Models*. In: *ArXiv*.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang (July 2019). *Neural Relation Extraction for Knowledge Base Enrichment*. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 229–240.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck (2018). *(Almost) Zero-Shot Cross-Lingual Spoken Language Understanding*. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6034–6038.

Clara Vania, Grace Lee, and Andrea Pierleoni (July 2022). *Improving Distantly Supervised Document-Level Relation Extraction Through Natural Language Inference*. In: *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*. Hybrid: Association for Computational Linguistics, pp. 14–20.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). *Attention is All you Need*. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., pp. 5998–600.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh (2015). *Cider: Consensus-based Image Description Evaluation*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant (Dec. 2022). *Overcoming Catastrophic Forgetting in Zero-Shot Cross-Lingual Generation*. In:

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 9279–9300.

Boxin Wang et al. (Dec. 2023). *Shall We Pretrain Autoregressive Language Models with Retrieval? A Comprehensive Study*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 7763–7786.

Cunxiang Wang, Pai Liu, and Yue Zhang (Aug. 2021). *Can Generative Pre-trained Language Models Serve As Knowledge Bases for Closed-book QA?* In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 3241–3251.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang (July 2022). *OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework*. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 23318–23340.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou (Aug. 2021). *K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters*. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 1405–1418.

William Yang Wang (July 2017). *"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection*. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 422–426.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang (2021). *KEPLER: A Unified Model for Knowledge Embedding and*

Pre-trained Language Representation. In: *Transactions of the Association for Computational Linguistics* 9, pp. 176–194.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao (July 2022). *Language Models as Knowledge Embeddings*. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*. Main Track. International Joint Conferences on Artificial Intelligence Organization, pp. 2291–2297.

Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec (Oct. 2023). *VQA-GNN: Reasoning with Multimodal Knowledge via Graph Neural Networks for Visual Question Answering*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21582–21592.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao (Dec. 2022). *AdaMix: Mixture-of-Adaptations for Parameter-efficient Model Tuning*. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 5744–5760.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi (July 2023). *Self-Instruct: Aligning Language Models with Self-Generated Instructions*. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 13484–13508.

Yuxia Wang et al. (Mar. 2024). *M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection*. In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Yvette Graham and Matthew Purver. St. Julian's, Malta: Association for Computational Linguistics, pp. 1369–1407.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao (2022). *SimVLM: Simple Visual Language Model Pretraining with Weak Supervision*. In: *International Conference on Learning Representations*.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen (May 2023). *Should You Mask 15% in Masked Language Modeling?* In: *Proceedings of the 17th Conference*

of the European Chapter of the Association for Computational Linguistics. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 2985–3000.

Chenxi Whitehouse, Monojit Choudhury, and Alham Aji (Dec. 2023). *LLM-powered Data Augmentation for Enhanced Cross-lingual Performance*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 671–686.

Chenxi Whitehouse, Fenia Christopoulou, and Ignacio Iacobacci (Dec. 2022). *EntityCS: Improving Zero-Shot Cross-lingual Transfer with Entity-Centric Code Switching*. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 6698–6714.

Chenxi Whitehouse, Fantine Huot, Jasmijn Bastings, Mustafa Dehghani, Chu-Cheng Lin, and Mirella Lapata (2023a). *Parameter-Efficient Multilingual Summarisation: An Empirical Study*. In: *ArXiv*.

Chenxi Whitehouse, Clara Vania, Alham Fikri Aji, Christos Christodoulopoulos, and Andrea Pierleoni (July 2023b). *WebIE: Faithful and Robust Information Extraction on the Web*. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 7734–7755.

Chenxi Whitehouse, Tillman Weyde, and Pranava Madhyastha (May 2023). *Towards a Unified Model for Generating Answers and Explanations in Visual Question Answering*. In: *Findings of the Association for Computational Linguistics: EACL 2023*. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 1648–1660.

Chenxi Whitehouse, Tillman Weyde, Pranava Madhyastha, and Nikos Komninos (2022). *Evaluation of Fake News Detection with Knowledge-Enhanced Language Models*. In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 16, pp. 1425–1429.

Adina Williams, Nikita Nangia, and Samuel Bowman (June 2018). *A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference*. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, pp. 1112–1122.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung (Nov. 2019). *Code-Switched Language Models Using Neural Based Synthetic Data from Parallel Sentences*. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 271–280.

Genta Indra Winata et al. (May 2023). *NusaX: Multilingual Parallel Sentiment Dataset for 10 Indonesian Local Languages*. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 815–834.

Thomas Wolf et al. (Oct. 2020). *Transformers: State-of-the-Art Natural Language Processing*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45.

Jialin Wu and Raymond Mooney (Aug. 2019). *Faithful Multimodal Explanation for Visual Question Answering*. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, pp. 103–112.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer (Nov. 2020). *Scalable Zero-shot Entity Linking with Dense Entity Retrieval*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 6397–6407.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi (2023). *Fine-Grained Human Feedback Gives Better Rewards for Language Model Training*. In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig (July 2020). *Predicting Performance for Natural Language Processing Tasks*. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8625–8646.

Rong Xiang, Mingyu Wan, Qi Su, Chu-Ren Huang, and Qin Lu (Dec. 2020). *Sina Mandarin Alphabetical Words:A Web-driven Code-mixing Lexical Resource*. In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 833–842.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav (2019). *Visual Entailment: A Novel Task for Fine-grained Image Understanding*. In: *ArXiv*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov (2020). *Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model*. In: *International Conference on Learning Representations*.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro (Nov. 2020). *MEGATRON-CNTRL: Controllable Story Generation with External Knowledge Using Large-Scale Language Models*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 2831–2845.

Weijia Xu, Batool Haider, and Saab Mansour (Nov. 2020). *End-to-End Slot Alignment and Recognition for Cross-Lingual NLU*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 5052–5063.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel (June 2021). *mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer*. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 483–498.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto (Oct. 2020a). *Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia*. In:

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online: Association for Computational Linguistics, pp. 23–30.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto (Nov. 2020b). *LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention*. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 6442–6454.

Jian Yang, Shuming Ma, Dongdong Zhang, ShuangZhi Wu, Zhoujun Li, and Ming Zhou (Apr. 2020). *Alternating Language Modeling for Cross-Lingual Pre-Training*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 9386–9393.

Yuan Yao, Jiaju Du, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun (Nov. 2021). *CodRED: A Cross-Document Relation Extraction Dataset for Acquiring Knowledge in the Wild*. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4452–4472.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun (July 2019). *DocRED: A Large-Scale Document-Level Relation Extraction Dataset*. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 764–777.

Keren Ye and Adriana Kovashka (2021). *A Case study of the Shortcut Effects in Visual Commonsense Reasoning*. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35, pp. 3181–3189.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant (2024). *Making Retrieval-Augmented Language Models Robust to Irrelevant Context*. In: *The Twelfth International Conference on Learning Representations*.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang (May 2022). *Dict-BERT: Enhancing Language Model Pre-training with Dictionary*. In: *Findings of the Association for Computational*

Linguistics: ACL 2022. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villav-icencio. Dublin, Ireland: Association for Computational Linguistics, pp. 1907–1918.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi (June 2019a). *From Recognition to Cognition: Visual Commonsense Reasoning*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6713–6724.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi (2019b). *Defending Against Neural Fake News*. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi (2021). *MERLOT: Multimodal Neural Script Knowledge Models*. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 23634–23651.

George Zerveas, Navid Rekabsaz, and Carsten Eickhoff (Dec. 2023). *Enhancing the Ranking Context of Dense Retrieval through Reciprocal Nearest Neighbors*. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 10779–10803.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao (2021). *Vinvl: Revisiting Visual Representations in Vision-Language Models*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. (2022). *Opt: Open Pre-trained Transformer Language Models*. In: *ArXiv*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi (2020). *BERTScore: Evaluating Text Generation with BERT*. In: *International Conference on Learning Representations*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning (Sept. 2017). *Position-aware Attention and Supervised Data Improve Slot Filling*. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language*

Processing. Copenhagen, Denmark: Association for Computational Linguistics, pp. 35–45.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu (July 2019). *ERNIE: Enhanced Language Representation with Informative Entities*. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 1441–1451.

Zhiyuan Zhang, Xiaoqian Liu, Yi Zhang, Qi Su, Xu Sun, and Bin He (Nov. 2020). *Pretrain-KGE: Learning Knowledge Representation from Pretrained Language Models*. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 259–266.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang (Nov. 2020). *LIMIT-BERT : Linguistics Informed Multi-Task BERT*. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 4450–4461.

Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Yuchen Lin, Meng Jiang, and Wenhao Yu (May 2022). *Knowledge-Augmented Methods for Natural Language Processing*. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Ed. by Luciana Benotti, Naoaki Okazaki, Yves Scherrer, and Marcos Zampieri. Dublin, Ireland: Association for Computational Linguistics, pp. 12–20.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (Dec. 2015). *Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books*. In: *The IEEE International Conference on Computer Vision (ICCV)*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving (2019). *Fine-tuning Language Models From Human Preferences*. In: *ArXiv*.