Structured Prediction for Compute Efficient and High Accuracy NLP

by

Hassan S. Shavarani

M.Sc., Simon Fraser University, 2016
B.Sc., Amirkabir University of Technology, 2014

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

© Hassan S. Shavarani 2024
SIMON FRASER UNIVERSITY
Summer 2024

Copyright in this work is held by the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.

Declaration of Committee

Name:	**Hassan S. Shavarani**
Degree:	**Doctor of Philosophy**
Thesis title:	**Structured Prediction for Compute Efficient and High Accuracy NLP**
Committee:	**Chair:** Nick Sumner Associate Professor, Computing Science

Anoop Sarkar
Supervisor
Professor, Computing Science

Angel Xuan Chang
Committee Member
Associate Professor, Computing Science

Fred Popowich
Examiner
Professor, Computing Science

Alan Ritter
External Examiner
Associate Professor
School of Interactive Computing
Georgia Institute of Technology

Abstract

Structured prediction in machine learning focuses on mapping a sequence of inputs to a sequence of outputs within a vast output space, with interconnected predictions, offering simplicity and speed while enhancing contextual understanding in NLP tasks.

In this dissertation, we revisit the applicability of structured prediction in modern, intricate NLP applications. We introduce a structured prediction-based approach for extracting linguistic knowledge from pre-trained encoder-only language models, and demonstrate the effectiveness of the extracted knowledge in enhancing translation quality of encoder-decoder models.

We showcase the efficacy of well-designed, simple structured prediction-based sequence labelling in handling complex entity linking with large entity vocabularies. Our proposed method, SpEL, not only simplifies and accelerates the process but also achieves state-of-the-art results on a prominent entity linking benchmark dataset.

Furthermore, we investigate *Entity Retrieval*, the application of our structured prediction-based entity linking framework as an alternative strategy to prevalent dense retrieval methods in retrieval-augmented question answering, particularly for factual questions about the real world. Our research underscores structured prediction as a compelling approach for modelling complex NLP tasks, particularly when prioritizing computational efficiency and high accuracy.

We conclude the dissertation with a review of additional contributions that either diverge from the primary focus or involve shared authorship, even if they pertain to the central theme of the dissertation.

Keywords: Structured Prediction, Entity Linking, Neural Machine Translation, Retrieval-Augmented Question Answering, Pretrained Language Modelling

Acknowledgements

I am deeply grateful to my senior supervisor, Dr. Anoop Sarkar, for his unwavering support and friendship throughout the years of my PhD. His guidance has been invaluable, and his mentorship has profoundly shaped my academic journey.

I would also like to thank the rest of my committee, Dr. Alan Ritter, Dr. Fred Popowich, Dr. Angel Chang, and Dr. Nick Sumner, for their constructive feedback. Their insightful comments and suggestions have significantly enhanced the quality of this dissertation.

I am also thankful to the past and present members of the Natlang Lab for their friendship and fruitful discussions. In particular, I am grateful to have collaborated with Ashkan Alinejad, Jetic Gu, Maryam Siahbani, and Nicolas Ong.

To my family and friends, especially my dear parents, I express my heartfelt gratitude for their continuous love and encouragement. Their support has been a constant source of energy and motivation.

Lastly, I want to thank my brilliant and beautiful wife, Nasim, as her belief in me has been a pillar of strength throughout this journey, and her endless encouragement and support has carried me through this program.

Table of Contents

Declaration of Committee ii

Abstract iii

Acknowledgements iv

Table of Contents v

List of Tables viii

List of Figures xi

Part I - Background 1

1 Introduction 2
- 1.1 Summary of Contributions . . . 3
- 1.2 Dissertation Outline . . . 3

2 Background 5
- 2.1 Language Modelling . . . 5
- 2.2 Pre-trained Language Models (PLMs) . . . 7
  - 2.2.1 Bidirectional Language Models . . . 8
  - 2.2.2 Causal Language Models . . . 10
  - 2.2.3 Large Language Models . . . 11
- 2.3 Information Extraction . . . 13
- 2.4 Structured Prediction . . . 14

Part II - Contributions 16

3 Structured Prediction for Machine Translation 17
- 3.1 Motivation . . . 17
- 3.2 Neural Machine Translation with BERT . . . 18

3.3 Linguistic Aspect Extraction from BERT . . . 19
3.3.1 Aspect Vectors . . . 19
3.3.2 Aspect Vector Extraction . . . 20
3.3.3 Integrating Aspect Vectors into the Translation Model . . . 22
3.4 Experiments and Results . . . 23
3.4.1 Data . . . 24
3.4.2 Linguistic Aspect Vector Extraction . . . 24
3.4.3 Linguistic Aspect Integrated Machine Translation . . . 29
3.5 Summary and Future Research . . . 31

4 Structured Prediction for Entity Linking **33**
4.1 Motivation . . . 33
4.2 Entity Linking Literature . . . 34
4.3 Conventional Structured Prediction-based Entity Linking Challenges . . . 37
4.4 SpEL: State-of-the-art Structured Prediction for Entity Linking . . . 39
4.5 Experiments and Results . . . 43
4.5.1 Data . . . 43
4.5.2 Evaluation Using GERBIL . . . 44
4.5.3 Setup . . . 44
4.6 SpEL Performance on AIDA . . . 45
4.7 Comparison to Large Language Models . . . 47
4.8 Practicality of the Fixed Candidate Sets . . . 49
4.9 SpEL Performance on Out-of-domain Data . . . 50
4.10 Summary and Future Research . . . 52

5 SpEL for Answering Entity-Centric Questions **53**
5.1 Motivation . . . 53
5.2 Retrieval for Retrieval-Augmentation . . . 54
5.3 Entity Retrieval for Question Answering . . . 55
5.4 Experiments and Analysis . . . 56
5.4.1 Setup . . . 56
5.4.2 Data . . . 57
5.4.3 Evaluation . . . 58
5.4.4 Entity Retrieval Performance using Question Entity Annotations . . . 59
5.4.5 Retrieval-Augmented Question Answering . . . 61
5.4.6 Entity Retrieval in absence of Question Entity Annotations . . . 62
5.4.7 Real-time Efficiency Analysis . . . 64
5.5 Related Studies . . . 65
5.6 Summary and Future Research . . . 65

6 Other Contributions **67**
6.1 Multi-class Multilingual Classification of Wikipedia Articles Using Extended Named Entity Tag Set 67
6.1.1 Dataset Collection and Annotation 68
6.1.2 Feature Selection and Models 69
6.1.3 Experiments and Results 71
6.2 Unified Examination of Entity Linking in Absence of Candidate Sets 72
6.2.1 Unified Black-Box Evaluation 72
6.2.2 Candidate Set Ablations 74
6.3 Additional Co-authored Contributions in Machine Translation 77

Part III - Summary **79**

7 Conclusion and Future Directions **80**

Bibliography **82**

List of Tables

Table 3.1 F-1 scores acquired after training the aspect extractor on German side of parallel data and passing the validation sets of each data set through trained aspect extractors. The `#tokens` column shows the number of tokens in the validation set. 25

Table 3.2 F-1 scores of fine-grained part-of-speech prediction of TIGER corpus test data (BERT encoded) fed to each of the trained aspect classifiers. The scores are calculated over a total of 7516 subword tokens in 358 test sentences of TIGER. Extractors trained on M30k, IWSLT, and WMT have not been provided with any part of TIGER before evaluation. . 26

Table 3.3 Classification scores of each aspect classifier when fed with other extracted aspect vectors. We expect the F-1 scores to be low so we can conclude that our aspect extractor truly excludes irrelevant information from each vector. 27

Table 3.4 The Transformer model settings for each dataset given the training data size. "N" is the number of layers in both encoder and decoder. Please see Section 3.2 for more information about model parameters. *The maximum number of subword tokens per batch. 27

Table 3.5 Evaluated cased BLEU score (calculated using `mteval-v14.pl` script) results on M30k, IWSLT, and WMT datasets. `#param` represents the number of trainable parameters (size of BERT model parameters [110.5M] has not been added to the model size for the aspect augmented and bert-freeze models since BERT is not trained in these settings). `runtime` is the total time the training script has ran and includes time taken for reading the data and training the model from scratch (iterating over the instances for all the epochs). *All the baseline results are achieved using our re-implementation of the mentioned papers.* 28

Table 3.6 Evaluated METEOR score results on M30k, IWSLT, and WMT datasets. 30

Table 3.7 Examples of improved translation quality of WMT data where *part-of-speech* aspect vectors have helped the model choose better words both syntactically and semantically. 32

Table 3.8 Examples of improved translation quality of WMT data where *word-shape* and *subword position* aspect vectors have helped the model choose a better sequence of subwords when it faces out-of-vocabulary tokens. 32

Table 4.1 Entity Linking evaluation results of SpEL compared to that of the literature over AIDA test sets. *#params on GPU* only considers the total number of parameters that will directly effect the cost of GPU acquisition and does not reflect upon the total amount of data loaded into/from main memory and disk. 46

Table 4.2 Mention Detection evaluation results of SpEL in comparison to the work of De Cao et al. (2021a) using their released evaluation code (from utils.py). As De Cao et al. (2021a) use PPRforNED candidate sets, we only compare the SpEL results calculated using these candidate sets. 47

Table 4.3 Comparison of the performance of SpEL (in no *mention-specific* candidate set setting) to zero and few shot GPT-3.5-turbo-16k (accessed on June 16, 2023) and GPT-4-0613 (for the best performing prompts we attempted; accessed on August 24, 2023). For few-shot experiments we constructed the prompt using the chain-of-thought (CoT) method of Wei et al. (2022). 48

Table 4.4 Entity Linking evaluation results of SpEL with a *fixed candidate set* size of 500K over AIDA test sets. Since the *context-aware* candidate sets require a mechanism for generating/looking up the candidate set during inference, we do not evaluate `testc` in this setting. 50

Table 4.5 Comparison of SpEL (with a *fixed candidate set* size of 500k) evaluation results with the literature on out-of-domain datasets. The best score is shown as bold and the second best is shown as underlined. †Results from (Kolitsas et al., 2018 - Table 2). ‡The "Oracle" results are calculated through feeding the gold annotations of each dataset to GERBIL, and depict the In-KB annotation quality of each dataset. . 51

Table 5.1 MRR scores comparing the retrieval quality of BM25, DPR, ANCE, and *Entity Retrieval* through the average of the reciprocal ranks of the first relevant document for each question. 60

Table 5.2 Question answering efficacy comparison between Closed-book and Retrieval-augmentation using BM25, DPR, ANCE, and *Entity Retrieval*. EM refers to the exact match between predicted and expected answers, disregarding punctuation and articles (`a`, `an`, `the`). * Results represent the average of two runs, accompanied by a margin of error based on a 99% confidence interval. 62

Table 5.3 Example questions from EntityQuestions (dev) to demonstrate the performance of *Entity Retrieval* in comparison to the other retrieval methods. 63
Table 5.4 Comparison of *Entity Retrieval* using SpEL identified entities to the best-performing dense and sparse retrieval methods of Table 5.2 on the StrategyQA dataset. Given the expected boolean results for StrategyQA questions, we restricted LLaMA-3 to generate only one token. *Acc.* indicates the fraction of answers that correctly match the expected Yes or No responses in the dataset, while *Inv #* represents the count of labels that are neither Yes nor No, but another invalid answer. ⋆ Results represent the average of two runs, accompanied by a margin of error based on a 99% confidence interval. 64
Table 5.5 Comparison of the required resources for each retrieval method in real-time execution. The reported total time values exclude the time taken to load the indexes and models, focusing solely on the time used to answer the questions. 65

Table 6.1 Statistics about the collected *Shinra 5-Language Categorization Dataset.* 69
Table 6.2 5-fold cross validation classification accuracy of the predicted labels for the fine-grained labels in SHINRA-5LDS dataset. † While we aimed to maintain settings comparable to their model, a fair comparison between our results and theirs is unfair due to disparities in dataset size and class numbers between our experiments and theirs. 71
Table 6.3 Comparison of recent entity linking systems within the unified black-box testing framework of GERBIL + `gerbil_connect`. Difference column reports the difference between our unified evaluation environment and the originally reported numbers. We have assessed all models twice for consistency. Except for (De Cao et al., 2021b), all models yielded identical scores, while De Cao et al. (2021b) showed a low variance of 0.08 in the results. Thus, the results mirror those reported by GERBIL, with the exception of (De Cao et al., 2021b), which is averaged over two runs. 73
Table 6.4 Comparison of entity linking systems after a) running the model with no access to hand-crafted candidate sets b) modifying the model to consider the entire AIDA in-domain vocabulary as the candidate set. 75

List of Figures

Figure 2.1 Transformer Architecture (from Vaswani et al., 2017). 7
Figure 2.2 A generic structured prediction model. 15
Figure 2.3 Example application of structured prediction for part-of-speech tagging. 15

Figure 3.1 Aspect extraction from the BERT embedding of the subword `_uar` in the German sentence: `Bucht die besten Hostels in Ouarzazate über Hostelsclub` (with English translation: *Book the best hostels in Ouarzazate via Hostelsclub*). 20
Figure 3.2 The extracted aspect vectors pass through the aspect classifiers to assure high correlation between the extracted information and the expected aspect tags. 21
Figure 3.3 An auto-encoder structure ensures the integrity of the information relayed through extracted aspect information. 21
Figure 3.4 Each pair of aspect vectors (except the *left-over* aspect) contributes in calculation of the dissimilarity training objective. 22
Figure 3.5 Integration of Extracted Aspect Vectors into the machine translation framework. The right hand side part of this figure is taken from Vaswani et al. (2017). 23

Figure 4.1 The expected structured prediction based entity linking output for example sentence: “Barack Obama wrote A Promised Land.” 33
Figure 4.2 Example output of structured prediction-based entity linking in practice. The results may diverge significantly from the anticipated outcome. 34
Figure 4.3 The two fine-tuning steps proposed in prior research to adjust the pre-trained language model for structured prediction-based entity linking. Step 1: General knowledge fine-tuning, Step 2: Domain specific fine-tuning. 38
Figure 4.4 SpEL, a structured prediction modelling framework for entity linking. In this example, we demonstrate top 3 most probable entities (including `O`) for each tokenized subword. 42

Figure 4.5 The three fine-tuning steps proposed to tune the pre-trained language model for SpEL. Step 1: Mention-aware general knowledge fine-tuning, Step 2: Mention-agnostic general knowledge fine-tuning, Step 3: Domain specific fine-tuning. 43

Figure 5.1 *Entity Retrieval* simplifies the process of obtaining augmentation documents by replacing the need to search through large indexed passages with a straightforward lookup. 54

Figure 5.2 The answer to `Who is the composer of The Swan Lake ballet?` can be found in the first paragraph of `Swan Lake` Wikipedia article. 56

Figure 5.3 nDCG@k scores comparing the quality of BM25, DPR, ANCE, and *Entity Retrieval* by considering both the relevance and the position of documents in the top k retrieved passages for each question. . . 60

Figure 5.4 Retrieval Accuracy scores showcasing the correlation between the number of retrieved documents and the expected answers' coverage in EntityQuestions (dev) subset. 61

Figure 6.1 Unified categorization feature extraction schema from Wikipedia articles. 70

Figure 6.2 Entity linking error distribution in four categories of over-generated (gray, vertical), under-generated (red, horizontal), incorrect entity (teal, north east) and incorrect mention (blue, north west) before candidate set ablations (left) and after the ablations (right). The y-axis is the error analysis ratio as described below. 76

Figure 6.3 Entity linking micro precision (blue, north east) and recall (red, north west) score differences over `testa` between model's original configuration and candidate set ablation configuration. 77

Part I

Background

Chapter 1

Introduction

Natural Language Processing (NLP) is a specialized field of artificial intelligence that focuses on understanding, manipulating, and generating language across various modalities, including text, speech, and even artificial languages like source code. Over the years, NLP has flourished, yielding a plethora of useful applications such as text classification, sentiment analysis, machine translation, question answering, and speech recognition.

In today's data-driven world, where vast amounts of unstructured text data are generated daily, NLP plays a crucial role in extracting insights, automating processes, and enhancing user experiences (Shah et al., 2023). Its relevance lies in its ability to analyze and comprehend human language at scale, facilitating applications in various domains such as customer service (Mashaabi et al., 2022), healthcare (Sezgin et al., 2023), finance (Schlaubitz, 2021), management (Kang et al., 2020), and education (Alqahtani et al., 2023). NLP enables organizations to derive valuable insights from textual data, improve decision-making processes, and create personalized experiences for users. As such, NLP continues to be indispensable in unlocking the potential of the vast amounts of text data available in today's digital age.

NLP has undergone several waves of technological advancement, progressing from specialized task-specific models to task-agnostic feature learners, transferable task solvers addressing various NLP tasks, and eventually evolving into general-purpose task solvers, broadening its application domain to encompass real-world tasks that were once beyond its reach in the early years of statistical NLP (Zhao et al., 2023). A consistent trend across these waves has been the refinement of NLP solutions, resulting in increased accuracy and utility, albeit accompanied by the growth in size and complexity.

The introduction and widespread adoption of GPT (OpenAI, 2023), Claude[1], and Gemini (Team et al., 2023) represent a pinnacle in this progression, enabling the creation of large, general-purpose conversational chatbots that have benefited diverse user groups[2]. However, these advancements are not without challenges. The significant investment re-

[1] `https://www.anthropic.com/news/claude-3-family`, accessed on May 9, 2024.

[2] Unfortunately, speakers of low-resource languages have not equally benefited from these advanced technologies.

quired for hardware, along with the environmental impact of their energy consumption, raises concerns about global warming (Patterson et al., 2021). Additionally, accessibility issues, both in terms of cost and infrastructure, limit the availability of these models to marginalized communities and research labs, prompting a need for model simplification and democratization to ensure universal access. This imperative is heightened when considering scaling laws (Kaplan et al., 2020b) and the trajectory of NLP solutions, exacerbating the aforementioned challenges.

This dissertation is one practical step towards the mentioned objectives. We leverage advanced NLP techniques to demonstrate the feasibility of enhancing performance and accuracy while prioritizing model simplification. Among all possible simplification frameworks, we focus on Structured Prediction (Section 2.4) and emphasize the extensive reuse of pre-trained models (Section 2.2) and the creation of such reusable models when unavailable for specific tasks. While our approach exemplifies one pathway to simplification, it is not exhaustive; rather, it serves as a paradigm for the community to integrate simplification and efficient design considerations. Our contributions aim to inspire a broader commitment to advancing both accuracy and efficiency simultaneously within the field.

1.1 Summary of Contributions

The dissertation makes primary contributions in the following directions:

- Improving neural **machine translation** by employing a structured prediction-based method to extract linguistic knowledge from pre-trained encoder-only language models. This approach utilizes structured prediction to extract valuable linguistic information from pre-trained models, enhancing translation performance without additional data or computational overhead in training the translation model or in inference.
- Designing a simple yet effective structured prediction-based **entity linking** approach capable of handling large entity vocabularies. Amidst the trend towards complex, large-scale generator-style models, our method demonstrates the potential of structured prediction using pre-trained language models to achieve state-of-the-art entity linking results.
- Improving retrieval-augmented **question answering** through our proposed structured prediction-based entity linking approach. While retrieval-augmentation is commonly employed to enhance question answering systems, we demonstrate the efficacy of entity linking as a powerful alternative to solely retrieval-based methods, particularly in entity-centric question scenarios.

1.2 Dissertation Outline

The chapters in this dissertation are organized as follows:

Chapter 2 provides relevant background information encompassing classic and pre-trained language models as well as structured prediction. We comprehensively discuss

bidirectional and causal language models, alongside recent advancements in large language models.

Chapter 3 discusses our proposed approach of employing structured prediction to extract linguistic knowledge from pre-trained language models and integrating them into the encoder-decoder translation models. The chapter reproduces results which we have originally published in (Shavarani and Sarkar, 2021). The associated github repository for this project is accessible at `https://github.com/sfu-natlang/SFUTranslate`.

Chapter 4 provides our innovative structured prediction-based entity linking framework. We carefully examine three key challenges encountered in traditional structured prediction-based models and present our novel solutions, leading to the development of our cutting-edge entity linking model, SpEL. By rigorously evaluating SpEL within a renowned entity linking framework, we demonstrate its superior performance against strong generative and non-generative entity linking models. The chapter reproduces results which we have originally published in (Shavarani and Sarkar, 2023). The associated github repository for this project is accessible at `https://github.com/shavarani/SpEL`.

Chapter 5 studies an application of our structured prediction-based entity linking framework for *Entity Retrieval*, our proposed retrieval strategy for enhancing retrieval-augmented question answering systems. Through empirical analysis, we show that *Entity Retrieval* outperforms conventional dense retrieval methodologies for entity-centric questions, all the while simplifying the task by reducing it to entity linking coupled with the retrieval of the initial sentences from corresponding knowledge base articles. The chapter reproduces results which we have originally published in (Shavarani and Sarkar, 2024). The associated github repository for this project is accessible at `https://github.com/shavarani/EntityRetrieval`.

Chapter 6 reviews additional contributions that either diverge from the primary focus or involve shared authorship, even if they follow the central theme of the dissertation. In this chapter, we present our findings from benchmarking multi-lingual multi-class classification of Wikipedia using our created SHINRA-5LDS dataset. Subsequently, we present the benchmarking results for recent entity linking systems in availability or absence of a specific resource known as candidate sets (Section 4.2).

Chapter 7 concludes the dissertation through a comprehensive summary of the key findings and contributions, and discusses future directions.

Chapter 2

Background

In this chapter, we delve into the essential background information that forms the foundation of this dissertation. We begin with a concise discussion on language modeling, a crucial component in both our proposed and enhanced frameworks. This discussion swiftly transitions into an exploration of pre-trained language models, highlighting their two distinct types: bidirectional and causal language models. In each category, we spotlight some exemplary pre-trained models. Next, we discuss large language models, which predominantly align with the structure of causal language models, yet bear significant differences. A selection of commonly adopted large language models will also be discussed. Subsequently, we shift our attention to information extraction and structured prediction, focusing on the latter as a specific technique employed for information extraction. Our aim is for this chapter to provide sufficient context to understand how the integration of structured prediction and pre-trained language models can advance the primary objective of this dissertation: enhancing NLP performance while advocating for simplicity in design.

2.1 Language Modelling

Language modeling has been a captivating challenge for several decades (Shannon, 1951). The primary objective of language modeling is to predict the most likely word that follows a given context, considering a specific language $\mathcal{L}$, such as English or French; enabling computers to comprehend and generate coherent and contextually appropriate language.

A language model M (Equation 2.1) is a probabilistic function that assigns a probability distribution over a sequence of words $w_1, w_2, \ldots, w_n$ in language $\mathcal{L}$; aiming to capture the likelihood of observing a particular sequence of words within the language.

Formally, given a sequence of words $w_1, w_2, \ldots, w_n$, the language model M computes the probability of the entire sequence $P(w_1, w_2, \ldots, w_n)$ as the product of the conditional probabilities of each word in the sequence, conditioned on the preceding words:

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_1, w_2, \ldots, w_{i-1}) \tag{2.1}$$

The task of a language model is to estimate these conditional probabilities based on a given training dataset of language samples. By learning the patterns and relationships

between words in the training data, the language model can then generate new sequences of words or predict the likelihood of unseen sequences in $\mathcal{L}$.

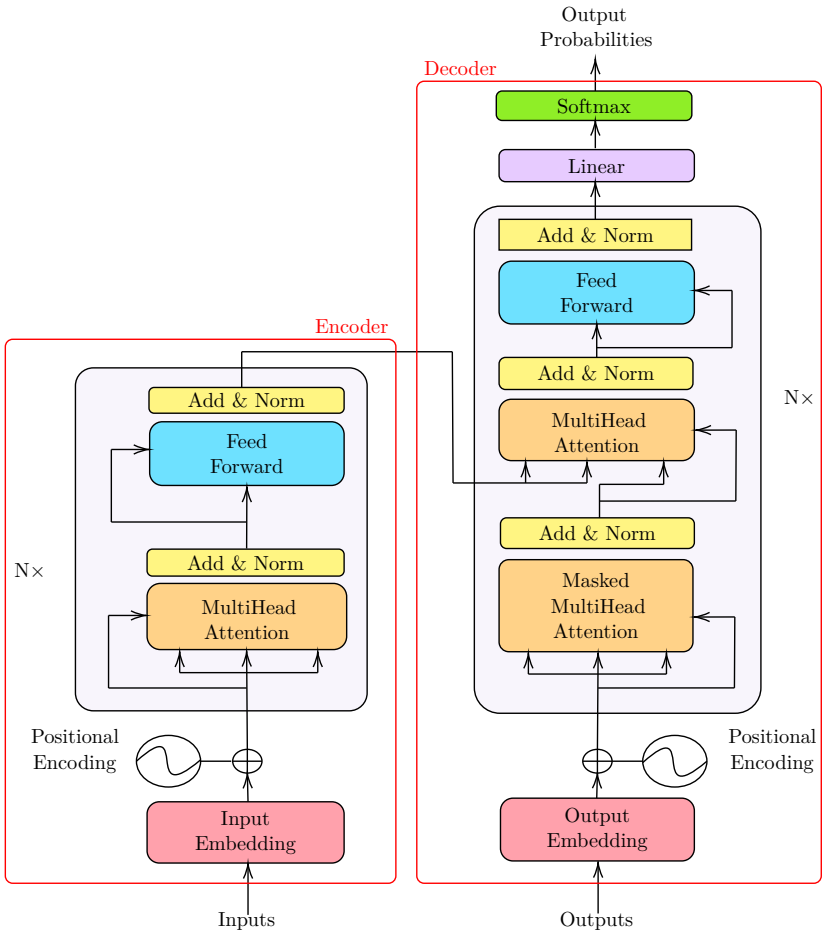
Language models are versatile and can operate at different levels of granularity, including character-level, word-level, and phrase-level. At the character-level, the model predicts the next character based on the preceding characters, while at the word-level, it predicts the next word given the previous words. For certain tasks involving fixed expressions, phrase-level modeling can also be utilized. The adaptability of language models to various levels of granularity enables them to be tailored to specific downstream language processing tasks.

The work of Bengio et al. (2000) stands as one of the pioneering efforts in proposing the use of neural networks for language modeling. Moreover, the introduction of recurrent neural network-based language models (Mikolov et al., 2010) and dense embedding vectors (Mikolov et al., 2013; Pennington et al., 2014) played a crucial role in revolutionizing the architectural design of language models. Prior to these breakthroughs, language models were constructed based on collecting n-gram frequencies from corpora. However, the advancements in recurrent neural network-based models and dense embedding vectors shifted the paradigm from traditional n-gram techniques to the utilization of deep neural networks (Hinton et al., 2012) for calculating probability distributions. This transition to deep learning methods empowered language models to capture more complex language patterns and semantic relationships, leading to significant advancements in various natural language processing tasks.

Initially, dense word embedding vectors were trained as fixed representations, serving as replacements for their string representations. This approach aimed to capture the semantic meaning of words by encoding them into continuous vector spaces. However, a limitation of early dense word embeddings was that they treated polysemous words, such as "bank" with multiple meanings (e.g., a financial institution and a riverbank), as having the same vector representation. As a result, the context and different meanings associated with such words were not fully captured, leading to potential ambiguity in downstream NLP tasks.

Contextualized word representations (McCann et al., 2017; Peters et al., 2018) represent a significant advancement over fixed word embedding representations. Unlike traditional fixed embeddings, contextualized word representations tailor dense vectors for each word based on the context words they appear with, allowing for a more nuanced and context-aware understanding of word meanings. The ELMo model (Peters et al., 2018) was among the pioneers in this area, suggesting the expansion of word representations from one vector to a set of vectors output from different layers of a deep neural network, utilizing a multi-layered bidirectional LSTM (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005). This approach led to improved language understanding and paved the way for subsequent contextual word representation models.

Figure 2.1: Transformer Architecture (from Vaswani et al., 2017).



2.2 Pre-trained Language Models (PLMs)

Seven years ago, Vaswani et al. (2017) published a seminal paper aimed at revolutionizing sequence-to-sequence mapping tasks such as translation. In contrast to LSTMs, their proposed architecture, Transformer (Figure 2.1), dispensed with recurrence for encoding input sequences. This facilitated parallelization during training, leading to enhanced precision and faster training. Additionally, it facilitated non-autoregressive transformation in sequence-to-sequence modeling (e.g. Schmidt et al., 2022), which was unattainable with recurrent networks. The emphasis on the multi-head attention module in Transformers also endowed them with greater efficiency in capturing long-range dependencies.

Subsequent research explored the potential of the Transformer architecture, showcasing the language modeling prowess of both encoder and decoder individually (Radford et al., 2018; Devlin et al., 2019). These investigations confirmed that a stack of transformer layers

can function as potent linguistic feature extractors when pre-trained[1] on sufficiently large monolingual datasets. Shortly thereafter, pre-training became a prevalent strategy for crafting versatile base language models, ushering in a paradigm shift towards pipelines comprised of fine-tuned iterations of these models to facilitate various language processing tasks.

At this juncture, two primary threads of base models emerged: one emphasizing language processing, which presupposed access to the entire input and employed bidirectional encoding for each input token, resulting in *Bidirectional Language Models*; and the other focusing on language generation, which assumed the autoregressive nature of next-token prediction, giving rise to *Causal Language Models*. We will now delve into each thread in greater detail.

2.2.1 Bidirectional Language Models

The lineage of encoder-only Transformer architectures commenced with the work of Devlin et al. (2019), introducing BERT (Bidirectional Encoder Representations from Transformers). BERT-style models were principally trained to create transferable NLP task solvers capable of addressing multiple NLP tasks with minimal fine-tuning. Architecturally, aside from harnessing the non-recurrent Transformer framework, which bolstered BERT's computational efficiency compared to its LSTM-based predecessors, pre-training BERT on large textual corpora endowed the model with a profound linguistic expertise, encompassing intricate language patterns and semantics. This pre-training regimen facilitated BERT in capturing contextual nuances comprehensively. The simplicity of BERT fine-tuning, achieved by appending an affine transformation layer atop BERT's contextual representations, enabled seamless adaptation for classification tasks. This fine-tuning versatility, coupled with its adeptness in capturing bidirectional contextual cues, ensured its impressive performance across diverse NLP tasks, catalyzing novel research directions and inspiring the development of numerous state-of-the-art models and applications in NLP.

The pre-training methodology for BERT was both straightforward and remarkably effective. It entailed presenting an extensive corpus of text to the model, containing sentences with masked words. The model's objective was to predict a probability distribution for each masked token, indicating the most probable replacement word from its entire vocabulary. This pursuit, known as *masked language modeling*, marked a significant departure from preceding language models that predicted the next word in a sentence. BERT's masked language modeling objective facilitated a broader linguistic comprehension by exploring unseen contexts and completing sentences with missing words. This approach proved pivotal in augmenting BERT's language understanding capabilities and setting it apart from antecedent models.

[1]Pre-training was a prevalent technique predating Transformers and was frequently employed in generating dense embedding vectors (Mikolov et al., 2013; Pennington et al., 2014).

While BERT represented a watershed moment in natural language processing, it was not without its limitations. Subsequent research endeavors focused on refining and enhancing BERT's performance across various dimensions. These endeavors aimed to improve both the model's accuracy and its computational efficiency. The pursuit of these optimizations was pivotal in rendering pre-trained language models more accessible and viable for real-world applications, while also propelling the frontier of state-of-the-art performance in NLP tasks. These iterative enhancements built upon BERT's foundational framework have led to a dynamic landscape of language models, fueling ongoing progress and innovation in the field of natural language processing.

Below, we provide insights into some notable members of the extended BERT family. It is worth noting that BERT is available in two sizes: base (with 110 million parameters) and large (with 340 million parameters), and extensions of BERT typically maintain this structure by offering models of corresponding size unless if the extension specifically targets optimizing the model size.

- **RoBERTa** (Liu et al., 2019) introduced larger batch sizes, dynamic masking, and longer training duration to achieve improved performance over BERT.
- **ERNIE** (Zhang et al., 2019) integrated external knowledge sources during pre-training to enhance BERT's understanding of domain-specific and factual information.
- **DistilBERT** (Sanh et al., 2019) distilled the knowledge from BERT into a smaller, more efficient architecture while retaining its performance to facilitate deployment in resource-constrained environments.
- **ALBERT** (Lan et al., 2020) addressed BERT's computational inefficiencies by implementing parameter-sharing techniques, achieving comparable performance with significantly reduced model size and computational cost.
- **ELECTRA** (Clark et al., 2020) introduced a novel pre-training objective called replaced token detection, which improved training efficiency while maintaining performance comparable to BERT.
- **Longformer** (Beltagy et al., 2020) adapted the self-attention mechanism to efficiently handle longer sequences, making it suitable for processing documents and other lengthy text inputs.
- **Reformer** (Kitaev et al., 2020) introduced reversible layers and locality-sensitive hashing techniques to enable efficient processing of long sequences while reducing memory consumption.
- **DeBERTa** (He et al., 2021) incorporated disentangled attention and enhanced decoding mechanisms to enhance BERT's contextual understanding and generation capabilities.

2.2.2 Causal Language Models

While BERT rose to prominence in the early stages following the advent of the Transformer architecture, GPT (Generative Pre-Training; Radford et al., 2018) introduced an alternative approach by employing pre-training with a decoder-only Transformer architecture, albeit without resorting to masked language modeling. Instead, GPT continued to refine the conventional next-word prediction (sentence completion) paradigm, akin to earlier language models. In contrast to BERT's emphasis on task-specific fine-tuning, GPT proposed to address various NLP tasks, including classification, entailment, sentence similarity, and question answering, through the sentence completion objective. This strategy necessitated multiple iterations of GPT models to match the performance of BERT-style models, but it succeeded in garnering significant attention and interest within the NLP community at the time of composing this chapter. Despite charting a divergent course from BERT, GPT's adaptability in tackling a spectrum of NLP tasks stands as a noteworthy accomplishment, solidifying its status as a significant contender in the realm of language models.

A critical point of comparison between BERT-style models and GPT-style models lies in their approach to contextual understanding and generation in inference. BERT, with its bidirectional encoding and focus on fine-tuning, excels in tasks requiring a comprehensive understanding of input context and precise classification, making it well-suited for tasks like sentiment analysis and named entity recognition (Section 2.3). In contrast, GPT, with its autoregressive nature and emphasis on sentence completion, exhibits a strength in generating coherent and contextually appropriate text[2], making it particularly adept in tasks like text generation and dialogue systems. This distinction underscores the importance of considering task-specific requirements and objectives when selecting between these two paradigms.

Below, we provide insights into some of the early pre-trained causal language models.

- **GPT-1** (Radford et al., 2018) introduced the concept of autoregressive language modeling at scale, demonstrating impressive results in generating coherent and contextually relevant text across various domains. However, it faced limitations in capturing long-range dependencies due to its unidirectional architecture, prompting subsequent iterations to explore solutions for enhanced context understanding.

- **GPT-2** (Radford et al., 2019) represented a significant leap in scale and performance, boasting a larger model size and dataset, which resulted in more fluent and diverse text generation capabilities. Despite its success, concerns were raised about the potential misuse of GPT-2 for generating misleading or harmful content, leading to a phased release strategy by its creators[3].

[2]For this reason, they are also referred to as generative language models.

[3]This strategy influenced subsequent iterations, wherein model weights were not publicly released, and access was restricted to APIs.

2.2.3 Large Language Models

Decoder-only Transformer-based language models underwent significant advancements with GPT-3 (Brown et al., 2020), which advocated scaling the model parameters by 10 times more than previous dense language models, reaching 175 billion parameters[4]. GPT-3 catalyzed a series of studies that focused on enlarging the parameter size of the model beyond what was previously considered feasible, while also leveraging larger training datasets in accordance with the scaling laws of neural language modeling (Kaplan et al., 2020a).

*Large language Model*s (LLMs), like their predecessor causal language models, relied on the next token prediction objective for training. However, they underwent instruction tuning (Ouyang et al., 2022) and human preference alignment (Bai et al., 2022), rendering them much more adept than mere sentence completion tools. As we compose this dissertation, the term 'large' in large language models refers to the parameter size of the neural language model, which can reach approximately one trillion parameters[5]. These models are exceedingly costly to train, but their development is motivated by their remarkable capabilities, as they are typically trained on vast datasets comprising natural language and source code.

These models fall into two categories: closed-source (developed and maintained by industrial companies like OpenAI, Meta, Google, Amazon, etc.) and open-source (publicly available or open weight). Closed-source large language models typically offer access via API, with costs based on the number of input and output tokens exchanged. Despite their expense, they often boast ongoing enhancements, leading to better performance. However, challenges arise from the lack of transparency regarding their parameter size, internal architecture, and training data specifics, complicating the attribution of performance successes or failures, often relying on speculative inference rather than empirical validation in research. Below, we review a few such language models.

- **GPT-3** (Brown et al., 2020) continues the GPT lineage by introducing the concept of large language models, boasting a 175-billion parameter model, ten times larger than its predecessors.
  - **GPT-3.5** representes a significant milestone in the evolution of large language models, and introduces enhancements to the architecture and training methodology, potentially surpassing GPT-3's parameter count and refining its performance across various natural language understanding tasks. Commonly, when mention-

[4]Although this approach significantly improved the representational capabilities of large language models compared to both causal and bidirectional language models, it also led to higher computational costs, thereby intensifying research budget constraints.

[5]For a chronological overview of large language models exceeding 10 billion parameters, readers are directed to (Zhao et al., 2023).

ing GPT-3, it implicitly refers to GPT-3.5 due to its widespread adoption and improved capabilities.

- **GPT-4** (OpenAI, 2023) is the last member of the GPT lineage (at the time of writing this dissertation) with undisclosed specifications. However, considering the scale of GPT-3, it is reasonable to speculate that GPT-4 exceeds 200 billion parameters, with discussions hinting at a potential implementation as a mixture of experts model[6].

- **Claude**[7], developed by Anthropic, is another closed-source large language model known for its robust performance and innovative training techniques, although specific details regarding its architecture and parameter count remain undisclosed.

- **Gemini** (Team et al., 2023) is a multimodal large language model from Google DeepMind in three variants: Ultra (largest), Pro, and Nano (smallest), positioned as a contender against GPT-4.

Open-source large language models are publicly available for download and local execution. Such models are quite easy to fine-tune using *parameter efficient fine-tuning* techniques (e.g. Hu et al., 2021). Below, we review a few such language models.

- **LLaMA-1** (Touvron et al., 2023a) was initially offered in 6.7B, 13B, 32.5B, and 65.2B parameter sizes, each varying in the number of heads in the multi-head attention mechanism: 32, 40, 52, and 64, respectively, contrasting with the original Transformer's 8 heads. It diverged from the conventional Transformer design through its utilization of a modified multi-head attention mechanism known as grouped multi-query attention (Ainslie et al., 2023). As well, LLaMA employed token representations with dimensions tailored to its model size, as exemplified by the 6B model's 4096-dimensional tokens, expanding to 5120 dimensions for the 13B model, 6656 for the 32.5B variant, and 8192 for the largest 65.2B model, in contrast to the fixed 512-dimensional token representation in the original Transformer architecture. Lastly, LLaMA used a context length of 2048 tokens, and distinguished itself by employing dynamic rather than static input embeddings, which were learned during the training process.

- **LLaMA-2** (Touvron et al., 2023b), available in 7B, 13B, and 70B size, extended the context length to 4096 tokens, double that of LLaMA-1, and benefited from more robust training, having been pre-trained on 40% more data. Furthermore, the incorporation of reinforcement learning from human feedback (RLHF; Christiano et al., 2017; Bai et al., 2022) in fine-tuning LLaMA-2's chat models enhanced alignment with human expectations, improving the quality and relevance of generated responses.

[6]Referenced from `https://en.wikipedia.org/wiki/GPT-4#Background`, accessed on May 9, 2024.

[7]`https://www.anthropic.com/news/claude-3-family`, accessed on May 9, 2024.

- **LLaMA-3**[8], the latest member of the LLaMA lineage (at the time of writing this dissertation), introduces 8B, 70B and 400B sized models, and increases the maximum context window size to 8192 tokens while training on a substantially larger dataset of 15 trillion tokens, emphasizing quality and incorporating diverse language data beyond English. The training of LLaMA-3 surpasses conventional practices by training approximately 75 times beyond the Chinchilla (Hoffmann et al., 2022) *compute optimal* point for an 8B model, resulting in a highly capable yet comparatively smaller model.

- **Mixtral-8x7B** (Jiang et al., 2024) represents a novel approach to large language modeling, utilizing a sparse mixture of 8 expert models within its architecture. Despite its name suggesting a 56 billion parameter model, Mixtral-8x7B contains 46.7 billion parameters, with shared modules such as self-attention among its expert sub-networks. This sparse activation scheme enables efficient inference, with only 2 experts active at any given time, resulting in faster performance on consumer hardware compared to models of similar size. Mixtral's design encourages computational and parameter efficiency while promoting specialized feature learning for diverse inputs, leading to improved generalization and performance across various tasks, including multilingual and coding domains.

2.3 Information Extraction

Information Extraction (Cowie and Lehnert, 1996; Grishman, 2019) focuses on extracting structured information from unstructured text, serving as a means for transforming raw text into actionable knowledge, thereby enhancing language comprehension and reasoning capabilities. This process operates on documents that follow similar templates but diverge in content, enabling automatic extraction of relevant facts. PLUM (Probabilistic Language Understanding Model; Ayuso et al., 1992) was one of the early applications contributing to the birth of information extraction.

A spectrum of tasks falls under information extraction. Among these, we highlight a number of key tasks:

- **Named Entity Recognition** (NER; Tjong Kim Sang and De Meulder, 2003; Nadeau and Sekine, 2007; Lample et al., 2016; Wang et al., 2021, 2023): Identifying and classifying entities mentioned in the text, such as names of persons, organizations, locations, dates, and numerical expressions.

- **Entity Linking** (Hoffart et al., 2011; Kolitsas et al., 2018; Shavarani and Sarkar, 2023): Extending the scope of NER to identify and classify any entity referenced in text by associating it with entries in a knowledge base. Unlike NER, which typically deals

[8] `https://llama.meta.com/llama3/`.

with a limited number of classes, entity linking entails matching entities to hundreds of thousands of entries in the knowledge base. In Section 4.2, we get into more details and provide a complete literature review on recent entity linking contributions.

- **Relation Extraction** (Miller et al., 1998; Zelenko et al., 2003; Miwa and Bansal, 2016; Ma et al., 2023): Determining the relationships between entities mentioned in the text, such as `MergerBetween(company1, company2, date)` from news articles reporting corporate mergers.

- **Event Extraction** (Grishman et al., 2005; Ji and Grishman, 2008; Wang et al., 2022; Huang et al., 2023): Identifying and extracting events or activities described in the text, along with relevant attributes such as time, location, and outcome.

- **Coreference Resolution** (Carter, 1987; Ng, 2010; Poesio et al., 2023): Resolving references to the same entity across the text, ensuring consistency and accuracy in the extracted information.

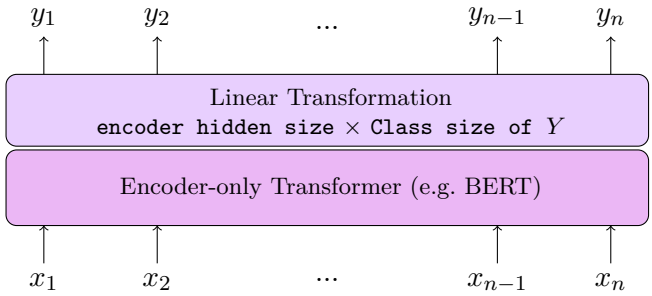
- **Template Filling** (Sundheim, 1991; Du et al., 2021): Populating predefined templates with information extracted from the text, where each template represents a specific structure or pattern of information.

The task of *Part-of-Speech* tagging involves attributing grammatical categories, including nouns, verbs, adjectives, and more, to each word within a sentence. While not directly classified under information extraction, it has proven beneficial to various language processing tasks, including information extraction (e.g. Suzuki et al., 2018; Ali et al., 2021, *inter alia*). Its applicability in information extraction roots in two primary factors: first, both tasks can be represented as sequence tagging, and second, when integrated, they can mutually benefit from transfer learning. As an example, the recognition of nouns in part-of-speech tagging may contribute to the identification of named entities in NER. In the next section, we focus on sequence labeling which serves as a central theme throughout this dissertation, and explore structured prediction as our approach to modeling sequence labeling tasks.

2.4 Structured Prediction

Structured prediction is a machine learning task focused on mapping a sequence of inputs $x_1, ..., x_n$ to a corresponding sequence of outputs $y_1, ..., y_n$ across an expansive output space, wherein each prediction is interconnected with others. Earlier studies in the literature utilized techniques such as HMM (Rabiner, 1989), CRF (Lafferty et al., 2001), and structured perceptron (Collins, 2002) to model language processing tasks through structured prediction (Dev et al., 2021). Subsequent to the emergence of pre-trained language models, a novel paradigm for structured prediction arose, wherein an encoder-only Transformer-based language model was employed for feature extraction, while a linear layer atop it facilitated the mapping of these extracted features into the output space (Figure 2.2). The inherent

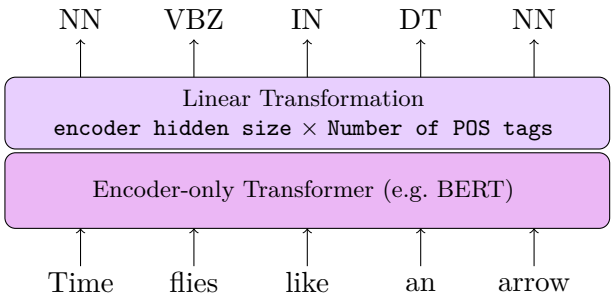
Figure 2.2: A generic structured prediction model.



capture of inter-label connectivity was posited to occur through the multi-head attention modules embedded within the language model.

BERT proposed employing this architecture for NER as an example, and soon various sequence labeling extensions such as part-of-speech tagging (Figure 2.3) were proposed. This method was compelling due to its amalgamation of the ease of utilizing a potent pre-trained model as a feature extractor with its linguistic comprehension accuracy, particularly evident in scenarios featuring ambiguity, as illustrated by the prediction of the *verb* part-of-speech for the word "flies" which could also be construed as a *noun* when considered in isolation, regardless of its surrounding predictions in Figure 2.3.

Figure 2.3: Example application of structured prediction for part-of-speech tagging.



Structured prediction is broadly applicable across numerous natural language processing tasks, such as named entity recognition (Wang et al., 2021), relation extraction (Han et al., 2019; Wang et al., 2022), event extraction (Li et al., 2013), coreference resolution (Liu et al., 2022), dependency parsing (Zhou et al., 2015), and semantic parsing (Xiao et al., 2016). In this dissertation, we focus on the utilization of structured prediction in modeling complex language processing tasks including machine translation, entity linking and question answering, with particular emphasis on compute efficiency while striving for enhanced accuracy in results.

Part II

Contributions

Chapter 3

Structured Prediction for Machine Translation

In this chapter, we study the application of structured prediction (Section 2.4) to improve the linguistic knowledge in neural machine translation. Notably, this chapter reproduces results which we have originally published in (Shavarani and Sarkar, 2021).

3.1 Motivation

Probing studies into large contextual word embeddings such as BERT have shown that these deep multi-layer models essentially reconstruct the traditional NLP pipeline capturing syntax and semantics (Jawahar et al., 2019); information such as part-of-speech tags, constituents, dependencies, semantic roles, coreference resolution information (Tenney et al., 2019a,b) and subject-verb agreement information can be reconstructed from BERT embeddings (Goldberg, 2019). We wish to extract the relevant pieces of linguistic information related to various levels of syntax from BERT in the form of dense vectors and then use these vectors as linguistic "experts" that machine translation models can consult during translation.

But can syntax help improving machine translation? Linzen et al. (2016); Kuncoro et al. (2018); Sundararaman et al. (2019) have reported that learning grammatical structure of sentences can lead to higher levels of performance in NLP models. In particular, Sennrich and Haddow (2016) show that augmenting translation models with explicit linguistic annotations improves translation quality.

In this direction, Sundararaman et al. (2019) identify *part-of-speech*, *case*, and *subword position*[1] as essential linguistic information to improve the quality of both BERT and the translation model. They extract each linguistic feature using the Viterbi output of separate models, embed the extracted linguistic information (similar to trained word embeddings) and append these vectors to the token embeddings.

We approach this problem from the novel perspective of extracting linguistic information encoded in BERT using a structured prediction framework and injecting such information into the translation model.

[1]A subword is a tokenization unit smaller than a word and larger than a character, aiding Transformer models in segmenting unknown words into recognizable units for processing. See (Sennrich et al., 2016) for more details of an example such segmentation.

3.2 Neural Machine Translation with BERT

Machine translation is the problem of transforming an input utterance sequence X in source language l_f into another utterance sequence Y (possibly with varying length) in target language l_e. Machine translation models search among all possible sequences in target language to find the most probable sequence based on the probability distribution of Equation 3.1.

$$P(y|X, y \in l_e) = \prod_{i=0}^{|max\ len|} p(y_i|X, y_0, ..., y_{i-1}) \tag{3.1}$$

Neural machine translation (NMT) tries to model the probability distribution $p(y|X)$ using neural networks by taking advantage of deep learning techniques. Transformers are encoder-decoder architectures that are commonly used for translation tasks. In Transformers, the input (in one-hot format) is passed through N layers of encoder and N layers of decoder. In each layer, the layer input passes through multiple attention heads (h heads; each considered a specialist in a different sentence-level linguistic attribute) and then gets transformed to the input for the next layer using a two layer feed-forward perceptron module with input size of d_{model} and hidden layer size of d_{ff}. The final probability distribution $p(y|X)$ is generated using an affine transformation applied to the output of the last feed-forward module in the N^{th} decoder layer.

Effective application of BERT in machine translation has been studied in a number of recent research projects. Clinchant et al. (2019) replace the encoder token embeddings of the Transformer model with BERT contextual embeddings. They also experiment with initializing all the encoder layers of the translation model with BERT parameters, in which case they report results on both freezing and fine-tuning the encoder parameters during training. In their experiments, BERT embeddings can help with noisy inputs to the translation model, but otherwise do not help improving translation performance.

Imamura and Sumita (2019) suggest that replacing the encoder layer with BERT embeddings and fine-tuning BERT while training the decoder leads to a *catastrophic forgetting* phenomenon where useful information in BERT is lost due to the magnitude and number of updates necessary for training the translation decoder and fine-tuning BERT. They present a two-step optimization regime in which the first step freezes the BERT parameters and trains only the decoder while the next step fine-tunes the encoder (BERT) and the decoder at the same time. Yang et al. (2020) also try to address the *catastrophic forgetting* phenomenon by thinking of BERT as a teacher for the encoder of the neural translation model (student network). They propose a dynamic switching gate implemented as a linear combination of the encoded embeddings from BERT and the encoder of the translation model. Zhang et al. (2021) adopt a similar approach, employing a three-phase optimization strategy that gradually unfreezes model parameters to address *catastrophic forgetting* during fine-tuning.

Weng et al. (2019) use multiple multi-layer perceptron (MLP) modules to combine the information from different layers of BERT into the translation model. To make the most out of the fused information, they also alter the translation model training objective to contain auxiliary knowledge distillation (Hinton et al., 2015) parts concerned with the information coming from the language model. Zhu et al. (2020) inject BERT into all layers of the translation model rather than only the input embeddings. Their model uses an attention module to dynamically control how each layer interacts with the representations. In both of these works, the training of the Transformer for translation becomes quite brittle and is prone to diverge to local optima.

3.3 Linguistic Aspect Extraction from BERT

Since BERT contextual embeddings contain a variety of information (linguistic and non-linguistic), extraction of relevant information plays an important role in further improvement of the downstream tasks. In the rest of this section, we define *aspect vectors* as single-purpose dense vectors of extracted linguistic information from BERT, discuss how aspect vectors can be extracted using structured prediction, and explain how to integrate aspect vectors into the translation model.

3.3.1 Aspect Vectors

To start the information extraction process, we must first select a limited (desired) set of linguistic attributes to identify in BERT embeddings. This attribute set can contain a number of linguistic aspects (e.g. part-of-speech). Each linguistic aspect itself will be defined over a possible aspect tag set (e.g. the set of $\{NOUN, ADJ, ...\}$ in part-of-speech). We show a linguistic attribute set with $\mathbb{A}$, show a generic aspect with a and point to its relative tag set with t_a.

We choose our linguistic attribute set ($\mathbb{A}$) as Sundararaman et al. (2019) suggest, however, we replace '*case*' with '*word-shape*'[2] since we believe the complete shape of the word is much more informative specially in subword settings. In addition, we consider a two-level hierarchy in part-of-speech tags to benefit from both higher accuracy in exploring the syntactic search space and lower model confusion in cases where the fine-grained tags are not helpful. Therefore, we consider coarse-grained and fine-grained *part-of-speech* (CPOS and FPOS), *word-shape* (WSH), and *subword position*[3] (SWP) to form our experimental linguistic attribute set ($\mathbb{A}$). Other linguistic attributes such as dependency parses, sentiment, or the key information of the source text (Hu et al., 2023) could be considered as aspects in our model but we leave them for future work.

[2]Representing capitalization (changing alphabet to `x` or `X`), punctuation, and digits (changing digits to `d`). As an example for *word-shape*, the subword *##arxiv.* in the token '*myarxiv.org*' will turn to *##xxxx.*.

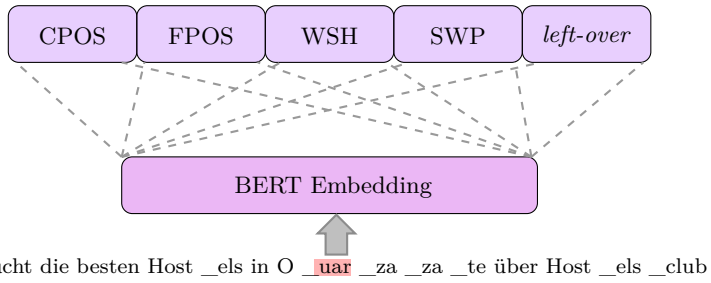
[3]Encoding the word with one of the three labels "`Begin`", "`Inside`", or "`Single`".

Given the definition of a linguistic aspect and inspired by the information bottleneck idea (Tishby and Zaslavsky, 2015), we define an *aspect vector* as a single-purpose dense vector extracted from BERT and containing information about a certain linguistic aspect of a particular (subword) token in the input sequence. Aspect vectors can be interpreted as feature values equivalent to a specific key (aspect).

3.3.2 Aspect Vector Extraction

For each embedding vector $\mathbf{E}$ and linguistic aspect a, we define M_a as an aspect-extraction function where $\mathbf{e}_a = M_a(\mathbf{E})$ is a single-purpose dense vector containing maximum aspect information and minimum irrelevant other information. Figure 3.1 demonstrates a number of such aspect extraction functions besides each other.

Figure 3.1: Aspect extraction from the BERT embedding of the subword `_uar` in the German sentence: `Bucht die besten Hostels in Ouarzazate über Hostelsclub` (with English translation: *Book the best hostels in Ouarzazate via Hostelsclub*).



We ensure the aspect encoding power of $\mathbf{e}_a$ by retrieving its equivalent tag in t_a using a classifier. The aspect prediction loss for a linguistic attribute set $\mathbb{A}$ of size n can be calculated as the average cross entropy loss ($\mathcal{L}_{CE}$) between the classifier prediction and the expected aspect tags for each aspect (Equation 3.2). Figure 3.2 depicts the relation of the extracted vectors and the aspect classifiers.

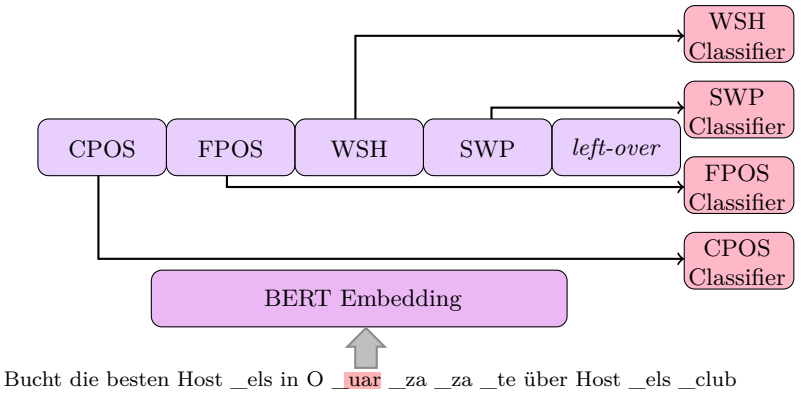
$$\mathcal{L}_a = \frac{1}{n} \sum_{i=0}^{|n|} \mathcal{L}_{CE}^i \tag{3.2}$$

We also ensure information integrity[4] of $\mathbf{e}_a$ by concatenating all the aspects (in addition to a "*left-over*" aspect equivalent to all the other non-interesting information) and reconstructing the original embedding vector $\mathbf{E}$ from them[5] in reconstruction vector $\mathbf{R}$. The reconstruction loss ($\mathcal{L}_r$) for the extracted aspect vectors can be calculated as the euclidean distance of the reconstruction vector $\mathbf{R}$ and the original embedding vector $\mathbf{E}$ (Equation 3.3). Figure 3.3 demonstrates this embedding reconstruction process.

[4]We don't expect M_a to change the information inside $\mathbf{E}$ but rather to extract the relevant information.

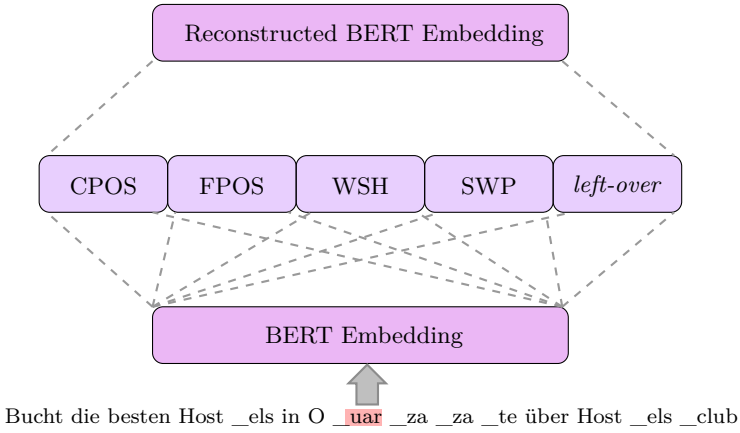
[5]This idea is analogous to stack-propagation (Zhang and Weiss, 2016) in which propagating the information loss for two tasks helps improving the quality of the encoded representations.

Figure 3.2: The extracted aspect vectors pass through the aspect classifiers to assure high correlation between the extracted information and the expected aspect tags.



$$\mathcal{L}_r = ||\mathbf{R} - \mathbf{E}||^2 \tag{3.3}$$

Figure 3.3: An auto-encoder structure ensures the integrity of the information relayed through extracted aspect information.

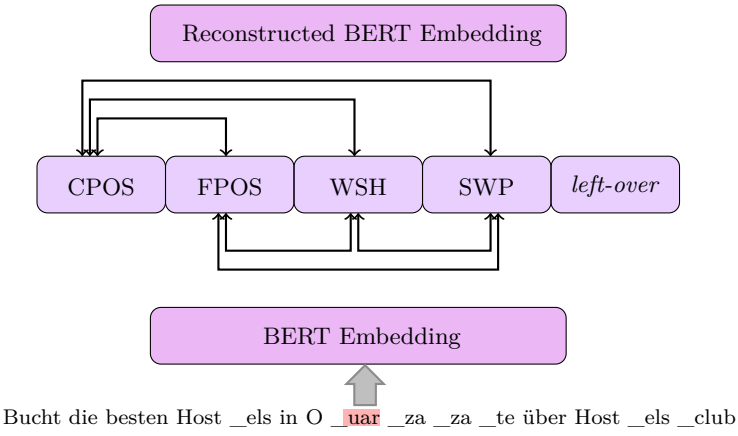


In addition, since our aspect extractor is similar in architecture to a multi-head attention module (with a difference in the fact that we know for what exactly each head will be responsible), to prevent learning redundant representations (Michel et al., 2019), we add the average euclidean similarity ($\mathcal{L}_s$) of each pair of aspect vectors to the training loss function (Equation 3.4). Figure 3.4 demonstrates the aspect vector pairs considered in calculation of the dissimilarity training objective.

$$\mathcal{L}_s = 1 - \left(\frac{1}{n(n-1)} \sum_{i=0}^{|n|} \sum_{j \neq i=0}^{|n|} ||e_i - e_j||^2 \right) \tag{3.4}$$

The aspect extractor will be trained over the accumulation of the three mentioned loss components (Equation 3.5).

Figure 3.4: Each pair of aspect vectors (except the *left-over* aspect) contributes in calculation of the dissimilarity training objective.



$$\mathcal{L}_{fe} = \mathcal{L}_a + \mathcal{L}_r + \mathcal{L}_s \tag{3.5}$$

As another important point, a BERT model has multiple encoder layers as well as an embedding layer. Choosing the proper layer which contains all of our desired aspects is not simply possible since different layers specialize in different linguistic aspects (Jawahar et al., 2019; Tenney et al., 2019a).

Therefore, as Peters et al. (2018) suggest, we define BERT embedding vector $\mathbf{E}$ as a weighted sum of all BERT layers (of size ℓ) using Equation 3.6 where α weights are learnable parameters and will be trained along with the other aspect extractor parameters.

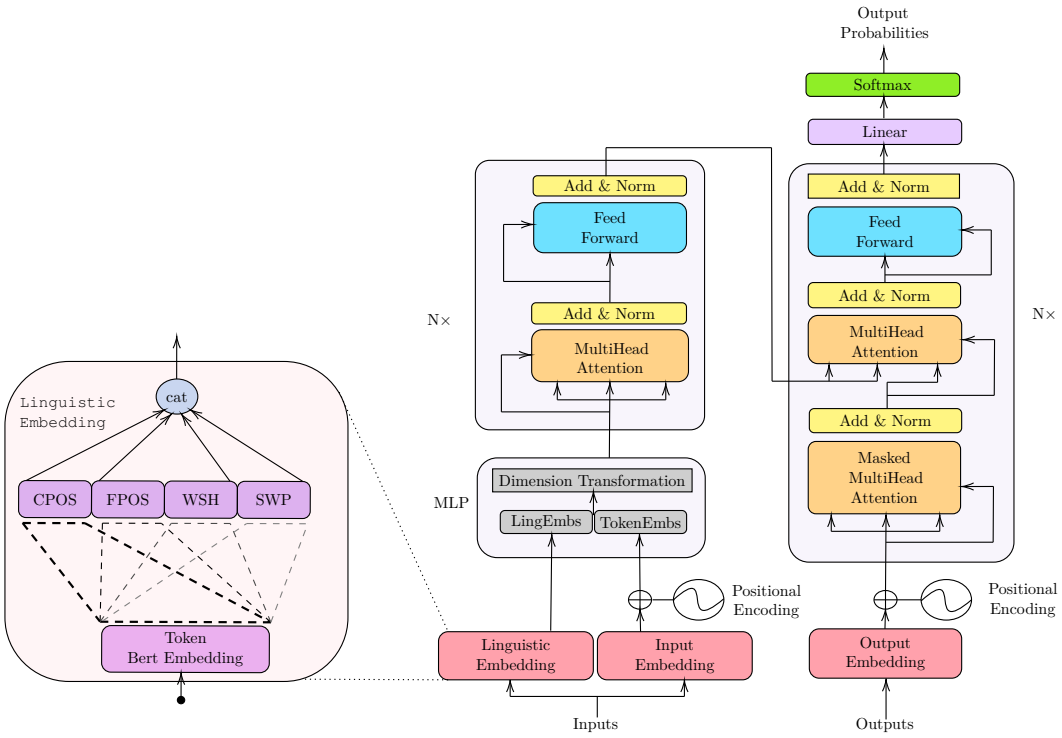
$$\mathbf{E} = \sum_{j=0}^{\ell} \alpha_j \mathbf{E}_j^{BERT} \tag{3.6}$$

3.3.3 Integrating Aspect Vectors into the Translation Model

Once the aspect vectors are created, we discard the classifiers and the reconstruction layers and place the encoder part of our trained aspect extractor (the mapping from BERT embeddings to aspect vectors) in an input integration module designed to augment the neural translation model input with aspect vectors[6].

[6]We use the same BERT subword model to provide our translation model with subword tokens.

Figure 3.5: Integration of Extracted Aspect Vectors into the machine translation framework. The right hand side part of this figure is taken from Vaswani et al. (2017).



The integration module (constructed using a two layer perceptron network) receives the concatenated aspect vectors (we call this concatenated vector a linguistic embedding[7]) and the token embedding (inherited from the Transformer model), and maps the linguistic embedding into a vector of the same size as the token embedding. Then, it projects the concatenation of both embeddings to a vector with the same size as the token embedding of the original Transformer model[8]. Figure 3.5 illustrates the process, wherein the linguistic embedding module functions as a structured prediction-based unit that maps each subword in the input to a dense vector representing its extracted linguistic embedding. Each aspect vector extracted in this module encapsulates the corresponding aspect information for the subword, taking into account the adjacent subwords' aspect content.

3.4 Experiments and Results

In this section, we initially examine our designed aspect extractor and report its classification accuracy scores. Next, we integrate the extracted aspect vectors into the translation

[7]This embedding vector can be similar to what a factor token contains in Factored-NMT (Garcia-Martinez et al., 2016) with a difference that it is generated in the space of linguistic aspects and does not need an embedding layer.

[8]This step is necessary to prevent any change in other parts of the model which would make comparison of the results unfair due to effects on the number of parameters and the learning capability of the model.

framework as explained in Section 3.3.3 and study the effects of integrated vectors on the performance of the models.

3.4.1 Data

We choose three German (which has explicit and nuanced linguistic features) to English datasets in different data sizes[9] to examine our proposed framework.

Multi30k[10] (M30k; our small dataset) contains a multilingual set of image descriptions in German, English and French. For this reason, we also consider experimenting on German to French as our second small dataset. The M30k data contains 29K training sentences, 1014 validation sentences (*val*) and 1000 test sentences (*test*2016).

IWSLT[11] (our medium sized dataset; Cettolo et al., 2012) contains sentences that are quite different from M30k since they are composed from the transcriptions of TED talks as well as dialogues and lectures[12]. The IWSLT data contains 208K training sentences, 888 validation sentences (*dev*2010) and multiple test sets (*tst*2010 to *tst*2015 with 1568, 1433, 1700, 993, 1305, and 1080 sentences, respectively).

WMT[13] (our large dataset with 4.5M training sentences) contains sentences from the proceedings of the European Parliament as well as web crawled news articles. We remove 0.05% of the training data (2290 sentences; lines with numbers divisible by 2000) and use it as the validation set (we call it *wmt_val*) and take *newstest* data from 2014 to 2019 as our test sets (with 3003, 2169, 2999, 3004, 2998 and 1997 sentences, respectively).

We remove train data sentences longer than 100 words and uncase and normalize both side sentences using `MosesPunctNormalizer`[14] before tokenization. The reference side of the test data remains unaltered in all the steps of our experiments.

3.4.2 Linguistic Aspect Vector Extraction

In this section, we examine our linguistic aspect extractor training procedure and analyze the quality of the extracted aspect vectors.

[9] In this chapter, we categorize dataset sizes into three groups: small, medium, and large. We define *small* as a dataset with tens of thousands of parallel sentences, *medium* as one with hundreds of thousands, and *large* as a dataset with millions of parallel sentences.

[10] AKA *Flickr30K* provided in task 1 of WMT17 multimodal machine translation.

[11] 2017 was the last year that the data for this task got updated.

[12] While the talks are quite polished, they still contain many verbal structures and sometimes even sounds (e.g. "`Imagine an engine going clack, clack, clack, clack, clack, clack, clack.`").

[13] Europarl+CommonCrawl+NewsCommentary. please note that in the later years this training set remained the same, but ParaCrawl data was added to it. We do not use ParaCrawl data since it is quite noisy and we aim to limit the effects of uncontrolled variables in our training data.

[14] `https://github.com/alvations/sacremoses/`.

	Subword Level					Word Level			
	CPOS	**FPOS**	**WSH**	**SWP**	**#tokens**	**CPOS**	**FPOS**	**WSH**	**#tokens**
M30k	96.88	96.18	99.79	99.93	16096	97.95	97.34	99.74	12823
IWSLT	92.69	90.48	99.73	97.14	22687	94.84	93.07	99.69	19039
WMT	92.64	91.60	97.74	98.94	70139	94.86	94.01	97.38	55135

Table 3.1: F-1 scores acquired after training the aspect extractor on German side of parallel data and passing the validation sets of each data set through trained aspect extractors. The `#tokens` column shows the number of tokens in the validation set.

We use the SpaCy German tagger model to acquire our intended linguistic aspect labels. Since SpaCy is trained on word-level while BERT is trained on subword level, we align the sequences using a heuristic divide-and-conquer monotonic alignment algorithm which finds the parts of the two sequences that are certainly equal and aligns the parts in between using recursive calls to itself. The fine-grained part-of-speech tagger in SpaCy[15] is pre-trained on TIGER Corpus (Smith et al., 2003) and inherits its 55 fine-grained tags from TIGER treebank. The coarse-grained SpaCy part-of-speech tagger has been trained by defining a direct mapping from 55 tags of the TIGER treebank to the 16 tags in the Universal Dependencies v2 POS tag set[16].

We use a 12-layer German pre-trained BERT base model for encoding the source sentences in aspect extractors. We use an uncased model as our translation model performs on lowercased data and the results are recased using the Moses recaser so that the results are cased BLEU scores comparable to other systems. We pass the BERT-encoded source sentences through a single perceptron middle layer of size 1000. We divide the output of this layer to '*number of aspects + 1*' splits to form our desired aspect vectors (of size 200).

We implement our aspect extractors using `pytorch` and initialize them using Xavier initialization (Glorot and Bengio, 2010). We perform backpropagation using SGD (initial learning rate of 0.05, momentum value of 0.9, gradient clip norm of 5.0). To cope with inequality in the frequency of the different tags in each aspect tag set (t_a; Section 3.3.1), we practice weighted backpropagation with weights proportional to the inverse frequency of each tag. We decay learning rate with a factor of 0.9 when the loss value stops improving.

We train three different aspect extractors, one for each dataset and feed in the source sentences of the dataset to our model in batches of size 32 for 3 epochs. Table 3.1 shows F-1 scores of classifying the validation set data using different aspect vectors after training the aspect extractors on the train set sentences. Please note that for calculating the word-level scores, in cases of disagreement between different subword tokens, the subword prediction of the first subword token has been counted as the prediction for the word label.

[15]At the time, SpaCy reported 96.52% accuracy for this model.

[16]`https://universaldependencies.org/v2/postags.html`.

We also validate our trained (on M30k, IWSLT, and WMT) aspect extractors against the manual annotations of TIGER treebank with which the SpaCy fine-grained part-of-speech tagger has been trained. We train an extra aspect extractor using the train set of TIGER corpus and test all four trained aspect extractors against TIGER data test set[17]. This experiment evaluates the absolute power of our structured prediction based aspect extractors in performing the aspect classification task. Please note that our goal in this experiment is not to achieve the state-of-the-art fine-grained part-of-speech tagging results as our aspect extractors receive their input from BERT and do not directly access the tagged input sentences. Table 3.2 contains the results of comparison between predictions of different aspect extractor classifiers and TIGER gold labels.

Aspect Extractor Training Data	**FPOS**	**SWP**
M30k	79.39	90.63
IWSLT	77.80	88.34
WMT	82.13	91.42
TIGER	84.64	92.64

Table 3.2: F-1 scores of fine-grained part-of-speech prediction of TIGER corpus test data (BERT encoded) fed to each of the trained aspect classifiers. The scores are calculated over a total of 7516 subword tokens in 358 test sentences of TIGER. Extractors trained on M30k, IWSLT, and WMT have not been provided with any part of TIGER before evaluation.

Uniqueness of Information in Linguistic Aspect Vectors. Considering the high F-scores for each aspect category in each dataset (Table 3.1), we can conclude that our aspect extractor maximizes the relevant information extraction from BERT embeddings. The loss in Equation 3.4 maximizes the distance between aspect vectors. To test whether this leads to a diverse set of aspect vectors, each specialized to their own linguistic attributes, we consider each aspect category a, after training the aspect extractors. We take each of the other extracted aspect vectors a' (except the "*left-over*" vector) and use each of them to train a new classifier that predicts the right class for category a based on aspect vector a'. This will test the correlation between the information in aspect vectors a' and the tags in category a. If the classification scores for this counterfactual test are high then our model has failed in fine-tuning each aspect vector to predict a particular linguistic aspect. We compare the classification scores to a trivial baseline: always predict the most frequent class. Table 3.3 shows the results of this counterfactual test on the aspect extractor trained on TIGER data. We can see that the average F-1 scores are very low when we use counterfactual aspect vectors to predict a linguistic aspect on which it was not fine-tuned (e.g. use aspect vector

[17]We use `german_tiger_test_gs.conll` in the version of TIGER released in *2006 CoNLL Shared Task - Ten Languages*.

TIGER test	Subword Level			
	CPOS	**FPOS**	**WSH**	**SWP**
most frequent class	NOUN	NN	xxxx	single
percentage in total	27.12	27.07	39.07	59.92
average classification F-1	1.89	0.23	12.20	42.97
`#tokens`	$7516 \times 3 = 22548$			

Table 3.3: Classification scores of each aspect classifier when fed with other extracted aspect vectors. We expect the F-1 scores to be low so we can conclude that our aspect extractor truly excludes irrelevant information from each vector.

Dataset	**WMT**	**IWSLT**	**M30k**
N	6	6	4
d_{model}	512	256	256
d_{ff}	2048	512	512
h	8	4	4
opt factor	1	2	1
opt warmup	4000	8000	2000
grad accumulation	8	2	1
batch size*	4096	4096	2560
epochs	7	20	20

Table 3.4: The Transformer model settings for each dataset given the training data size. "N" is the number of layers in both encoder and decoder. Please see Section 3.2 for more information about model parameters.
*The maximum number of subword tokens per batch.

trained on part-of-speech to predict word shape). This shows that our training method fine-tunes each aspect vector to its linguistic task.

To validate the loss in Equation 3.3, we calculate the average euclidean distance of the aspect extractor reconstructed vectors and the original BERT embedding vectors[18] for M30k German to English dataset. We unit normalize each of the vectors for a score in $[0, 1]$. The average euclidean distance value of 0.1863 tells us that the reconstruction component of the aspect extractor is capable of reconstructing vectors that are close to the original embedding vectors.

a) M30k	German to English				German to French			
	val	**test2016**	**#param**	**runtime**	**val**	**test2016**	**#param**	**runtime**
Vaswani et al. (2017)	39.63	38.35	9.5 M	84 min	31.07	30.29	9.4 M	93 min
Sundararaman et al. (2019)	40.03	38.32	13.9 M	514 min	32.55	32.71	13.6 M	504 min
Clinchant et al. (2019) (BERT Freeze)	40.07	39.73	9.1 M	99 min	33.83	33.15	9.0 M	104 min
Shavarani and Sarkar (2021) +M30k asp. vectors	**40.47**	40.19	10.1 M	104 min	34.45	**34.42**	9.9 M	108 min
Shavarani and Sarkar (2021) +WMT asp. vectors	38.72	**41.53**	10.1 M	102 min	**34.73**	34.28	9.9 M	118 min

b) IWSLT	**dev2010**	**tst2010**	**tst2011**	**tst2012**	**tst2013**	**tst2014**	**tst2015**	**#param**	**runtime**
Vaswani et al. (2017)	27.69	27.93	31.88	28.15	29.59	25.66	26.76	18.4 M	172 min
Sundararaman et al. (2019)	29.53	29.67	33.11	29.42	30.89	27.09	27.78	28.9 M	1418 min
Clinchant et al. (2019) (BERT Freeze)	30.31	30.00	34.20	30.04	31.26	27.50	27.88	18.0 M	212 min
Shavarani and Sarkar (2021) +IWSLT asp. vectors	29.03	29.17	33.42	29.58	30.63	26.86	27.83	18.9 M	214 min
Shavarani and Sarkar (2021) +WMT asp. vectors	**31.22**	**30.82**	**34.79**	**30.29**	**32.34**	**27.71**	**28.40**	18.9 M	211 min

c) WMT	**wmt_val**	**nt2014**	**nt2015**	**nt2016**	**nt2017**	**nt2018**	**nt2019**	**#param**	**runtime**
Vaswani et al. (2017)	28.96	26.91	26.93	31.42	28.07	33.56	29.77	68.7 M	35 h
Sundararaman et al. (2019)	28.56	27.80	26.93	30.44	28.63	33.87	30.48	93.8 M	258 h
Clinchant et al. (2019) (BERT Freeze)	28.63	27.54	27.15	31.69	28.30	33.89	**31.48**	69.1 M	33 h
Shavarani and Sarkar (2021) +WMT asp. vectors	**28.98**	**28.05**	**27.58**	**32.29**	**29.07**	**34.74**	**31.48**	70.3 M	46 h

Table 3.5: Evaluated cased BLEU score (calculated using `mteval-v14.pl` script) results on M30k, IWSLT, and WMT datasets. `#param` represents the number of trainable parameters (size of BERT model parameters [110.5M] has not been added to the model size for the aspect augmented and bert-freeze models since BERT is not trained in these settings). `runtime` is the total time the training script has ran and includes time taken for reading the data and training the model from scratch (iterating over the instances for all the epochs).
All the baseline results are achieved using our re-implementation of the mentioned papers.

3.4.3 Linguistic Aspect Integrated Machine Translation

After confirming the adequacy and uniqueness of linguistic information in aspect vectors, we integrate the encoder part of aspect extractors into the translation model and perform translation experiments on M30k, IWSLT, and WMT datasets.

We implement our baseline Transformer model using the guidelines suggested by Rush (2018) in our translation toolkit SFUTranslate[19] and extend it for implementing the aspect-augmented model as well as the syntax-infused Transformer and Transformer with bert-freeze input setting. Table 3.4 provides the configuration settings for each of the datasets used in our experiments.

We use the pre-trained WordPiece[20] (Schuster and Nakajima, 2012) tokenizer packaged and shipped with BERT (containing 31,102 subword tokens for German language) to tokenize the source side data, and tokenize the target side data with MosesTokenizer[21] followed by the same WordPiece tokenizer model, trained on the target data, to split the target tokens into subword tokens. We set the target side WordPiece vocabulary size to 30,000 subwords for English and French. Our models share the vocabulary and embedding modules of both source and target (Press and Wolf, 2017) since both source and target are trained in subword space. The shared vocabulary sizes of M30k (German to English), M30k (German to French), IWSLT, and WMT are 16645, 16074, 40807, 47940, respectively.

We generate target sentences using beam search with beam size 4 and length normalization factor (Wu et al., 2016) of 0.6. We merge the WordPiece tokens in the generated sentences (a post-processing step to create words) and use MosesDetokenizer[22] to detokenize the generated outputs. We use Moses recaser[23] to produce cased translation outputs. We use a single GeForce GTX 1080 GPU for M30k experiments and a single Titan RTX GPU for IWSLT and WMT experiments.

For all models, we set positional encoding max length to 4096, dropout to 0.1, loss prediction smoothing to 0.1, and initialize the models using Xavier initialization. We train all models using `NoamOpt` optimizer (Rush, 2018) and perform the gradient accumulation trick (Ott et al., 2018) with one update per a number of batches (Table 3.4; `grad accumulation`) to simulate larger batch sizes on a single GPU.

[18] Average results of Equation 3.3 for all the tokens in the train set.

[19] `https://github.com/sfu-natlang/SFUTranslate`.

[20] `https://github.com/huggingface/tokenizers`.

[21] `https://github.com/alvations/sacremoses`.

[22] `https://github.com/alvations/sacremoses`.

[23] `https://github.com/moses-smt/mosesdecoder`.

a) M30k	German to English		German to French	
	val	test2016	val	test2016
Vaswani et al. 2017	37.20	36.56	53.22	52.58
Sundararaman et al. 2019	38.14	37.13	54.18	54.37
Clinchant et al. 2019 (BERT Freeze)	38.44	37.42	55.10	54.50
Aspect Augmented +M30k asp. vectors	**39.22**	38.17	**56.21**	**56.40**
Aspect Augmented +WMT asp. vectors	38.90	**38.57**	56.12	55.98

b) IWSLT	dev2010	tst2010	tst2011	tst2012	tst2013	tst2014	tst2015
Vaswani et al. 2017	31.82	31.99	34.57	32.65	32.49	30.65	31.13
Sundararaman et al. 2019	32.91	32.95	35.35	33.10	33.17	31.32	31.90
Clinchant et al. 2019 (BERT Freeze)	33.34	32.78	35.42	33.12	33.20	31.22	31.45
Aspect Augmented +IWSLT asp. vectors	32.86	32.86	35.38	33.43	33.23	31.37	31.87
Aspect Augmented +WMT asp. vectors	**33.78**	**33.56**	**36.14**	**33.51**	**33.98**	**31.86**	**32.37**

c) WMT	wmt_val	nt2014	nt2015	nt2016	nt2017	nt2018	nt2019
Vaswani et al. 2017	**30.65**	33.80	33.70	**37.10**	34.44	37.81	36.05
Sundararaman et al. 2019	29.23	31.57	31.61	34.05	31.87	35.18	33.60
Clinchant et al. 2019 (BERT Freeze)	30.39	33.46	33.20	36.13	33.73	37.24	35.68
Aspect Augmented +WMT asp. vectors	30.61	**33.97**	**33.99**	37.01	**34.71**	**38.17**	**36.48**

Table 3.6: Evaluated METEOR score results on M30k, IWSLT, and WMT datasets.

We compare our model to three baselines : (1) the vanilla Transformer model which does not use any external source of information, (2) the syntax-infused Transformer model of Sundararaman et al. (2019) which explicitly embeds linguistic aspect labels and concatenates their embedding to the token embedding, (3) the Transformer model with bert-freeze input setting (Clinchant et al., 2019) which replaces the source embedding tokens of the Transformer architecture with output embeddings of the pre-trained BERT model.

During each training trial, we perform 9 validation set evaluation steps, one after visiting each 10% of the data. In each step, the validation set is translated with the current state of the model (at the time of evaluation) and the generated sentences are detokenized and compared to the validation set reference data to produce sentence-level Bleu (Lin and Och, 2004) scores. The best scoring model throughout training is selected as the model with which the test set(s) are translated.

For M30k and IWSLT data sets, we train two separate models, one using the aspect vectors trained on the source side of its own training data (in-domain) and the other using the aspect vectors trained on the source side of WMT data (out-of-domain). We use cased

BLEU (evaluated with the standard `mteval-v14.pl` script) and METEOR (Denkowski and Lavie, 2014) to compare different models. Tables 3.5 and 3.6 show the results of evaluating the models trained with different mentioned settings.

The evaluation results show that taking advantage of aspect vectors improves the accuracy of translating German to both English and French in M30k as well as German to English in IWSLT and WMT. Also, in majority of the cases WMT-trained aspect vectors have pushed the model to produce more accurate results since they contain more generalized information. Based on these results, we conjecture that aspect vectors trained on large out-of-domain data can be helpful in low-resource settings but we leave the examination of this idea for future work.

Aside from performance, our model is approximately 5 times faster than syntax-infused translation model while demanding fewer trainable parameters. Although it is not as fast as bert-freeze model in large dataset settings (because of the size of computations required for calculating the linguistic embedding), it is comparable in speed to bert-freeze in medium and small data scale settings.

For smaller datasets (containing a few hundred thousand sentence pairs or less), the broader perspective of BERT knowledge is helpful in limiting the search space for the model. So using our technique, the translation model receives more information regarding the general use cases of (locally) rare words. Linguistic aspect vectors also help the model better understand less familiar (in comparison to what is frequent in its limited size training data) syntactic structures in input sentences. This is why we believe aspect vectors can be helpful in low-resource settings.

Improving models with large amounts of data (with several million sentence pairs) is a challenging task. The best practice in training neural translation models is to initialize the embedding module with small random values and let the model search through the parameter space to find the optimal parameter settings. Extracted aspect vectors, as an external source of monolingual knowledge on the source side, are a more reasonable starting point for large models than random initialization. Integrating aspect vectors thus helps these models find a better path towards the optimal point(s) and increases the chances of the model ending up in a more desirable point in search space.

Tables 3.7 and 3.8 demonstrate some examples of cases where aspect vectors have been useful in improving the translation quality.

3.5 Summary and Future Research

In this chapter, we proposed a simple method of employing structured prediction to extract linguistic information from BERT and integrate them into machine translation framework. We showed that the linguistic aspect vectors provide the translation models with out-of-domain knowledge, improving not only the translation quality but also the model's ability to handle out-of-vocabulary words.

Source	Ihm **werde weiterhin vorgeworfen**, unerlaubt geheime Informationen weitergegeben ...
Reference	He **is still accused of passing on** secret information without authorisation.
Vaswani et al. 2017	He has **also been accused of having** illegally **passed on** secret information.
Clinchant et al. 2019	He **continues to be accused of** fraudulently **passing on** secret information.
Sundararaman et al. 2019	He **is also accused of having pass** unauthorised secret information **on**.
Aspect Augmented NMT	He **is still accused of passing on** illegal secret information.
Source	Auto und Traktor krachen zusammen: Frau stirbt bei schrecklichem Unfall
Reference	Car and tractor **crash together**: woman **dies** in terrible accident
Vaswani et al. 2017	Car and traktor **cranes together**: women **die** in the event of a terrible accident.
Clinchant et al. 2019	Cars and tractors **are killing** women in the event of a terrible accident.
Sundararaman et al. 2019	Auto and tractor **are blowing together**: woman **dies** when the terrible accident occurs.
Aspect Augmented NMT	Car and tractor **crash together**: woman **dies** in terrible accidents.

Table 3.7: Examples of improved translation quality of WMT data where *part-of-speech* aspect vectors have helped the model choose better words both syntactically and semantically.

Source	Bucht die besten Hostels in **Ouarzazate** über Hostelsclub.
Reference	Book the best hostels in **Ouarzazate** with Hostelsclub.
Vaswani et al. 2017	Book the best hostels in **ouarzazazate** with Hostelsclub.
Clinchant et al. 2019	Book the best hostels in **Ouarzate** with Hostelsclub.
Sundararaman et al. 2019	Book the best hostels in **ouarzazazate** with Hostelsclub.
Aspect Augmented NMT	Book the best hostels in **Ouarzazate** with Hostelsclub.
Source	Die **Deutsche Bahn** will im kommenden Jahr die Kinzigtal-Bahnstrecke verbessern.
Reference	The **Deutsche Bahn** hopes to improve the Kinzigtal railway line in the coming year.
Vaswani et al. 2017	**The German Railway** wants to improve the Kinzig valley railway line next year.
Clinchant et al. 2019	**Christian Deutsche Bahn** intends to improve the Kinzig valley railway next year.
Sundararaman et al. 2019	**The German Railway** wants to improve the kinziggia railway line next year.
Aspect Augmented NMT	**Deutsche Bahn** wants to improve the Kinzig valley railway in the coming year.

Table 3.8: Examples of improved translation quality of WMT data where *word-shape* and *subword position* aspect vectors have helped the model choose a better sequence of subwords when it faces out-of-vocabulary tokens.

Future research may focus on reimagining the integration module as a multi-head attention module, attending on different linguistic aspects of the current subword or subword tokens of a single word. Expanding the range of linguistic aspects, particularly incorporating syntactic dependencies and morphology, and investigating how aspect vector size influences translation quality are promising avenues for exploration. Additionally, assessing the efficacy of aspect vectors trained on large out-of-domain data in low-resource settings and examining the applicability of linguistic aspect vectors in domains beyond machine translation warrant further investigation.

Chapter 4

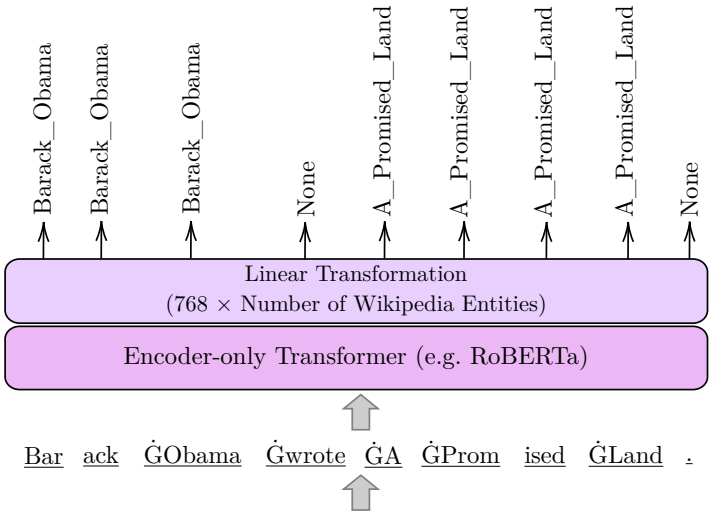
Structured Prediction for Entity Linking

In this chapter, we study the application of structured prediction (Section 2.4) to refine entity linking. Notably, this chapter reproduces results which we have originally published in (Shavarani and Sarkar, 2023).

4.1 Motivation

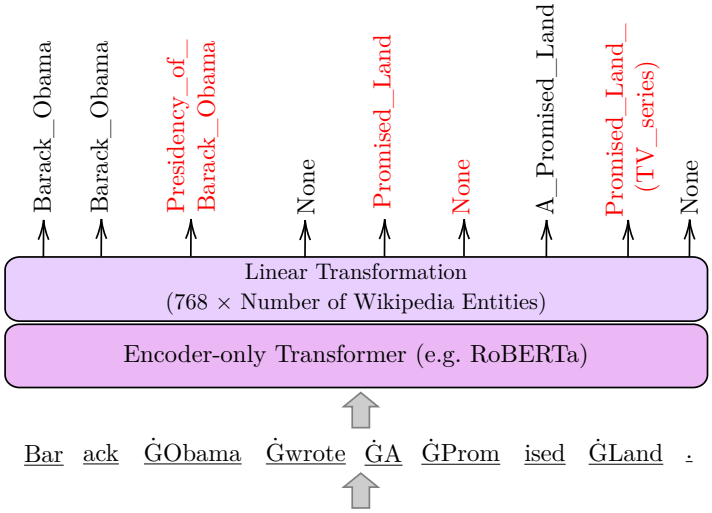
Entity linking can be modelled as sequence tagging using structured prediction (subword token multi-label classification), where a pre-trained Transformer model is utilized to encode each input subword token into a multi-layer context-aware dense vector representation. A classifier head fine-tunes each token representation to predict the entity for each subword token. The ideal performance of such an architecture would lead to the performance demonstrated in Figure 4.1.

Figure 4.1: The expected structured prediction based entity linking output for example sentence: "Barack Obama wrote A Promised Land."



However, adopting conventional approaches (e.g. Broscheit, 2019) for implementing the standard structured prediction-based entity linking architecture may yield suboptimal outcomes. Figure 4.2 illustrates a potential outcome that could arise from the implementation of the aforementioned architecture.

Figure 4.2: Example output of structured prediction-based entity linking in practice. The results may diverge significantly from the anticipated outcome.



Due to the large number of possible entities and issues with consistency of entity prediction across multiple subword tokens, structured prediction for entity linking (surprisingly) has not been studied in-depth. In this dissertation, we propose to re-examine the effectiveness of structured prediction for entity linking in the hopes of providing the models that are both more accurate and faster in inference.

In the rest of this chapter, we first delve into the factors contributing to poor performance in conventional structured prediction-based entity linking approaches. Subsequently, we propose our novel approach aimed at mitigating these issues. We conclude this chapter by presenting experimental results and a performance analysis of our proposed approach.

4.2 Entity Linking Literature

Knowledge bases, such as Wikipedia and Yago (Pellissier Tanon et al., 2020), are valuable resources that facilitate structured information extraction from textual data. Entity Linking (Shen et al., 2014) involves identifying text spans (mentions) and disambiguating the concept or knowledge base entry to which the mention is linked.
Entity linking can be viewed as three interlinked tasks (Broscheit, 2019; Poerner et al., 2020; van Hulst et al., 2020; Shavarani and Sarkar, 2023):

(1) *Mention Detection* (Nadeau and Sekine, 2007) to scan the raw text and identify the potential spans that may contain entity links.

(2) *Candidate Generation* (e.g. Fang et al., 2020) to match each potential span with a number of potential entity records in the knowledge base.

(3) *Mention Disambiguation*[1] (Ratinov et al., 2011; Yamada et al., 2022) to select one of the potential entity records for each detected mention.

An end-to-end entity linking system does all three tasks and links text spans to concepts. The system can either have independently modelled components (Piccinno and Ferragina, 2014; van Hulst et al., 2020) or jointly modelled components (Hoffart et al., 2011; Kolitsas et al., 2018; De Cao et al., 2021a).

Recent entity linking models use pre-trained representation learning methods that are based on Transformer architecture. These methods commonly utilize bidirectional language models like BERT (Section 2.2.1) or causal language models such as GPT (Section 2.2.2), which are then fine-tuned on specific entity linking training datasets. In a number of such techniques, entity linking is framed as another well-studied problem, and the best solution for that task is applied.

Autoregressive encoder-decoder sequence-to-sequence translation is one such approach. De Cao et al. (2021b) consider the input text as the source for translation and the text is annotated with Wikimedia markup containing the mention spans and the entity for each mention[2]. Instead of mapping the entity identifiers into a single id (this is the default in many techniques), their model generates the entity label in a token-by-token basis (it generates the Wikipedia URL one token at a time). The generation process follows a constrained decoding schema that prevents the model from producing invalid entity URLs.

De Cao et al. (2021a) use a BERT-style bidirectional model fine-tuned to identify potential spans (mention detection) by learning spans using a *begin* probability and an *end* probability for each subword in the input text. For each potential span, they use a generative LSTM-based language model to generate the entity identifiers (token-by-token), and limit the generation process to pre-defined candidate sets. We show that despite being interesting, token-by-token generation of entity identifiers is not necessary for the best performance in entity linking models.

Mrini et al. (2022) frame entity linking as a sequence-to-sequence translation task using BART (Lewis et al., 2020a). They duplicate the BART decoder three times to fine-tune the model in a multi-task setting. The two additional decoder modules are trained using auxiliary objectives of mention detection and re-ranking. While this training method increases the model size during training, they mitigate increased model size and speed at inference time by excluding the auxiliary decoder modules and employing sampling and re-ranking techniques on the generated target sequences.

[1]Until a few years ago (Shen et al., 2014), the task of Entity Linking was considered equivalent to *Named Entity Disambiguation* (NED) where it takes as input the identified named entities from the *Named Entity Recognition* task.

[2]The process of converting plain text into text containing Wiki markup is referred to as Text Wikification (Mihalcea and Csomai, 2007).

Zhang et al. (2022) use Question Answering as a way to frame the entity linking task. They suggest a two-step entity linking model in which they use a fine-tuned Transformer-based BLINK (Wu et al., 2020) model to find all the potential entity records that might exist in the text and then utilize a fine-tuned question answering ELECTRA model (Section 2.2.1) to identify the matching occurrences of the potential entities discovered in the first step. This approach obtains high accuracy, however it is very resource intensive and inference speed is slow.

Broscheit (2019) proposes a very simple entity linking model which places a classification head on top of BERT and directly classifies each token representation using a softmax over all the entities known to the model.

Other techniques focus on enhancing the entity linking knowledge in BERT (or one of its variants) and utilize one or more such *knowledge-enhanced* models to perform the task of entity linking. Peters et al. (2019) inject Wikipedia and Wordnet information into the last few layers of BERT, Poerner et al. (2020) inject pre-trained Wikipedia2Vec (Yamada et al., 2016) entity embeddings into the input layer of the language model while freezing the rest of its parameters, and Martins et al. (2019) leverage a Stack-LSTM (Dyer et al., 2015) NER model to enhance entity linking performance using multi-task learning to improve entity linking.

Kolitsas et al. (2018) jointly model mention detection and mention disambiguation using an LSTM-based architecture while reusing the candidate sets created by Ganea and Hofmann (2017) as a replacement for the candidate generation step, and Kannan Ravi et al. (2021) follow a similar framework while modeling each of mention detection and mention disambiguation using separate BERT models. Feng et al. (2022) compute entity embeddings (instead of using pre-trained ones) using the average of the subword embeddings of the candidates and compare them to the average of the subword embeddings for the potential span (training a Siamese network; Bromley et al., 1993). Févry et al. (2020) investigate pre-training strategies specifically tailored for Transformer models to perform entity linking, diverging from the conventional use of pre-trained models. And, van Hulst et al. (2020) propose a modular configuration that composes mention detection, candidate generation, and mention disambiguation in a pipeline approach, incorporating the most promising components from prior research.

Candidate Sets

Formally, entity linking receives a passage ($\mathbf{p}$) containing words w_1, ..., w_n and produces a list containing ℓ span annotations. Each span annotation is expected to be a triplet of the form (*span start*, *span end*, *entity identifier*). The *span start* and *span end* values are expected to be character positions on the original passage $\mathbf{p}$, and the *entity identifier* values are selected from a predefined vocabulary of entities (e.g. there would be approximately 6.5 million entities to choose from when entity linking to Wikipedia). The massive entity

vocabulary size increases the model's hardware requirements and in cases renders the task intractable.

To solve the entity vocabulary size problem, a common approach is to limit it to K most frequent entities in the knowledge base[3]. This vocabulary can be simply considered as the *fixed candidate set* for linking each mention to the knowledge base. Where no more information is available, the model will have to choose one entity from this set.

The selected *fixed candidate set* may lack many of the expected entity annotations at inference. Consequently, even if the model is highly capable, it may perform poorly during inference due to its inability to suggest the expected entities. Recognizing this challenge, there is a consensus among existing literature (Kolitsas et al., 2018; Broscheit, 2019; Peters et al., 2019; Poerner et al., 2020) to augment the *fixed candidate set* by including the expected entities necessary for inference.

An alternative is to use *mention-specific candidate sets* (Kolitsas et al., 2018; Peters et al., 2019; Kannan Ravi et al., 2021; De Cao et al., 2021b,a). *Mention-specific candidate sets* can be divided into two groups:

(1) *context-agnostic mention-specific sets* which are usually generated over large amounts of annotated text and try to model the probability of each mention span to all possible entity identifiers without assuming a specific context in which the mention would appear. KB+Yago (Ganea and Hofmann, 2017) contains candidate sets for approximately 200K mentions created over the entire English Wikipedia combined with the Yago dictionary of Hoffart et al. (2011).

(2) *context-aware mention-specific sets* can be constructed if there is a method for identifying mentions and a set of candidates for those mentions. For example, Pershina et al. (2015) have built such candidate sets, called PPRforNED. Such lists have been primarily suggested for the task of entity disambiguation where the mention is provided in the input. As gold mentions are not available for real-world use cases of entity linking, this type of candidate sets have fallen out of favor.

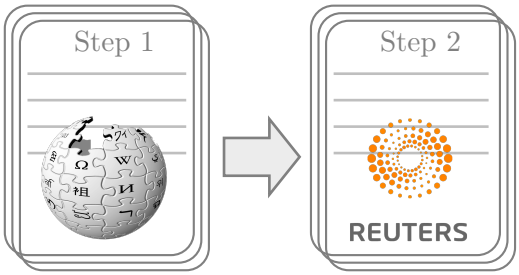
Mention-specific candidate sets consist of many entity identifiers and the correct entity identifier is not guaranteed to exist given the mention span.

4.3 Conventional Structured Prediction-based Entity Linking Challenges

In our preliminary experiments, we identified three primary factors that contribute to the poor performance of conventional structured prediction-based entity linking approaches. This section comprehensively examines each factor, and Section 4.4 outlines the solutions to these challenges, leading to the presentation of our state-of-the-art *Structured Prediction for Entity Linking* model.

[3]For Wikipedia, we can define an entity frequency as the number of times a title is hyperlinked in the other pages.

Figure 4.3: The two fine-tuning steps proposed in prior research to adjust the pre-trained language model for structured prediction-based entity linking. Step 1: General knowledge fine-tuning, Step 2: Domain specific fine-tuning.



1st Challenge. Conventional approaches presuppose knowledge of span boundaries during fine-tuning, which are absent during inference.

As depicted in Figure 4.3, prior research proposes a two-step fine-tuning process. In the first step, the model acquires a broad understanding of entity knowledge from a large corpus of entity-annotated text, such as Wikipedia. The second step involves adjusting the entity-knowledgeable language model to align with the in-domain data distribution, such as news articles from Reuters news agency[4].

To discuss this challenge, we consider the sentence "`On April 9, Tech hired coaching veteran and ESPN analyst Nell Fortner`" as an illustrative example, featuring four entities: `coaching` linking to `Coach_(basketball)`, `veteran` linking to `Veteran`, `ESPN` linking to `ESPN_Radio`, and `analyst` linking to `Sports_analyst`. It is important to note that while `Nell Fortner` is a person's name, it lacks a corresponding link in our hypothetical knowledge base, and thus remains unannotated with a knowledge base link.

In our examination of the two-step fine-tuning process, we discovered that during training, input sentence tokenization occurs in chunks, ensuring that each chunk either represents an entity or contains no entities. This approach facilitates the separate modeling of mention spans, aiding the model in recognizing span boundaries, albeit without explicitly modeling the probabilities of span beginnings or endings. This results in the following tokenized sentence, where a noticeable outcome is the distinct modeling of the space character (represented with Ġ in the example) preceding entity spans.

> *ĠOn ĠApril Ġ9 , ĠTech Ghired* **Ġ** *co aching* **Ġ** *ve ter an Ġand* **Ġ** *ESPN* **Ġ** *an alyst ĠN ell ĠFort ner*

However, during inference, this chunk-based tokenization approach is inapplicable, as entity spans for unseen sentences are unknown. Consequently, a tokenization mismatch arises, potentially compromising the model's performance during inference, as depicted in the

[4] `https://www.reuters.com/`.

following tokenized sentence where the same sentence is tokenized regardless of mention spans.

> *ĠOn ĠApril Ġ9 , ĠTech Ghired Ġcoaching Ġveteran Ġand ĠESPN Ġanalyst ĠN ell ĠFort ner*

The discrepancy in span tokenization, particularly regarding spans containing entity links, significantly undermines the entity linking performance of the structured prediction framework. Such observations might prompt the conclusion that structured prediction is ill-suited for this task. In Section 4.4, we will propose a mention-agnostic extension to step 1 fine-tuning to address this issue.

2nd Challenge. Structured prediction classifies tokens independently, disregarding neighboring predictions. A crucial observation is that structured prediction may overlook neighboring predictions when determining a label for the current token. Consequently, our prior assumption, that pre-trained language models inherently capture the inter-connectivity between output labels, proves inadequate for this task. To address this limitation, we require a more robust condition that reinforces inter-connectivity at both the word and phrase levels. In Section 4.4, we will propose a *context sensitive* prediction aggregation strategy to address this issue.

3rd Challenge. The output vocabulary is excessively large, occasionally encompassing redirect titles. The model encounters numerous entity annotations with specific surface form spans. For instance, in the example of Figure 4.2, we had `Barack Obama`, which could correspond to both `Barack_Obama` and `Presidency_of_Barack_Obama`, among other entities. While in-domain fine-tuning enables the neural representations of the model to align with the in-domain data, typically, the output space remains untuned. In Section 4.5, we will propose shrinking the in-domain output space to a desired *fixed candidate set* of entities to address this issue.

4.4 SpEL: State-of-the-art Structured Prediction for Entity Linking

In this section, we address the challenges outlined in Section 4.3, and develop our state-of-the-art *Structured Prediction for Entity Linking* model.

Formally, for a sequence of subwords $S = \{s_1, s_2, ..., s_n\}$[5], we employ RoBERTa (Section 2.2.1), in both `base` and `large` sizes, as our underlying model M to encode S into $H \in \mathbb{R}^{n \times d}$ where d is the hidden representation dimension of M. Each representation $h_i \in H$, $i \in 1, \ldots, n$ is then transformed into a distribution over the *fixed candidate set* (Section 4.2) of size KB using a transformation matrix $W \in \mathbb{R}^{d \times KB}$. This results in $P_i = h_i W$, where P_i represents the probability distribution for the i^{th} subword in the input sequence.

[5]When feeding a long text in training and inference, we split the text into smaller overlapping chunks.

When we set up fine-tuning for this task, we use hard negative mining (Gillick et al., 2019) to find the most probable incorrect predictions in the batch[6]. In each fine-tuning step, we update the network based on the subword classification probabilities of the hard negative examples as well as the expected prediction. To increase inference speed, the classification head does not normalize the predicted scores.

We employ binary cross-entropy with logits (Equation 4.1) as our loss function[7], which is calculated over many factors. Let N represent the total number of selected examples (ψ) comprising the one positive example corresponding to the expected prediction and the negative examples. Additionally, $a_{i,j}$ takes a value of 1 when the j^{th} member of ψ correctly points to the entity identifier for subword s_i, $p_{i,j}$ denotes model's predicted score for linking the j^{th} member of the selected examples to the i^{th} subword, and σ is the sigmoid function, which maps the scores to $[0, 1]$.

$$\mathcal{L}_i = -\frac{1}{N}\sum_{j=1}^{N}\left[a_{i,j}\cdot\log\Big(\sigma(p_{i,j})\Big) + (1-a_{i,j})\cdot\log\Big(1-\sigma(p_{i,j})\Big)\right] \tag{4.1}$$

During inference we collect the top k predictions for each subword i based on the predicted probabilities in P_i. We then collect subwords that belong to the same word into a single group, which we call the *word annotation*. For each word annotation, we generate an aggregated entity identifier prediction set by taking the union of the entity identifiers predicted for the subwords. We then compute the weighted average of the prediction probabilities for each entity identifier to obtain the word-level probability score over entities. Consecutive word-level entity labels when they refer to the same concept are joined into a single mention span over that phrase.

When a *mention-specific* candidate set is available, and the mention surface form matches one of the mentions in the candidate set, we filter out any predictions from the phrase annotation that are not present in the candidate set, regardless of their probability[8]. The final prediction for an entity span is generated based on the most probable prediction in the phrase annotations, excluding the ones annotated with `O` (which means the phrase is not an entity). As an additional post-processing cleanup step, we reject phrase annotations that span over a single punctuation subword (e.g. a single period or comma) or a single function (sub)word (e.g. `and`, `by`, ...). In such cases, we manually override the model's prediction to `O`.

This *context sensitive* prediction aggregation strategy leads to improved performance and enhances prediction results in inference. Our strategy ensures that annotation spans do

[6]We add random negative examples in addition to hard negatives to make sure we get to 5K negative examples for each batch when fine-tuning on CoNLL/AIDA and 10K negatives for general fine-tuning.

[7]We choose this loss function to ensure comparability with previous studies.

[8]The presence of a *mention-specific* candidate set is *not* a prerequisite for our model to be effective.

not begin or end inside a word[9], and the conflicts between the subword predictions within a word are resolved by the average prediction probability for each entity identifier.

This simpler method to ensure label consistency does better than using a CRF layer. Although our experiments show that a CRF layer does not improve our model, our readers can think of the suggested strategy as a domain-tailored, non-parametric, and rule-driven version of a CRF layer which guides the model to unify the predicted subword-level entity predictions considering the local context. Based on our experiments (Table 4.2), although we do not explicitly model Mention Detection (as predicting the *span start* and *span end* probability scores or separate `BIO` tags) for each subword in inference time, we observe a high in-domain accuracy in distinguishing `O` spans from non-`O` spans in predictions as a result of using the *context sensitive* prediction aggregation strategy.

Our modelling framework, SpEL (Figure 4.4), stands for *Structured Prediction for Entity Linking*[10].

Fine-tuning Procedure. Heinzerling and Inui (2021) argue that pre-trained language models can produce better representations when they are first fine-tuned on a much larger entity-linked training data (almost like a further pre-training step) and then subsequently fine-tuned for the entity-linking task. Following conventional methods (Figure 4.3), we perform such a multi-step fine-tuning procedure: first fine-tuning on a large dataset encompassing general knowledge on the set of linked concepts and then fine-tuning on an in-domain dataset specific to the target domain over which we aim to perform entity linking.

In the first step (general knowledge fine-tuning), we fine-tune the pre-trained language model using text that includes links to the knowledge base (in our experiments, we use a large subset of English Wikipedia[11]).

As discussed in Section 4.3, entity linking can benefit from tokenization that is aware of mentions, by using special space character subword tokens before and after each span linked to an entity. This approach aids the model in identifying the starting and ending subwords of entity mention spans (Broscheit, 2019). However, this imposes a mismatch in the distributions of the data in fine-tuning compared to inference, where the model does not have access to the entity mentions to perform the customized tokenization. To address this issue, as a subsequent fine-tuning step, we iterate again through the large entity-linked dataset which is re-tokenized *without* the knowledge of the mention spans.

In the third and last fine-tuning step (domain specific fine-tuning), we refocus the model's attention to the in-domain dataset annotated with a *fixed candidate set* which

[9]For instance, in the word `U.S.`, if in the `U.S` part, the subwords have high likelihood for the concept `The United States` and the ending `.` refers to an `O`, the conflict is resolved so that the entire word `U.S.` is linked to `The United States`.

[10]`https://github.com/shavarani/SpEL`.

[11]Limited to the articles that contain some presence of the entities in our selected *fixed candidate set*.

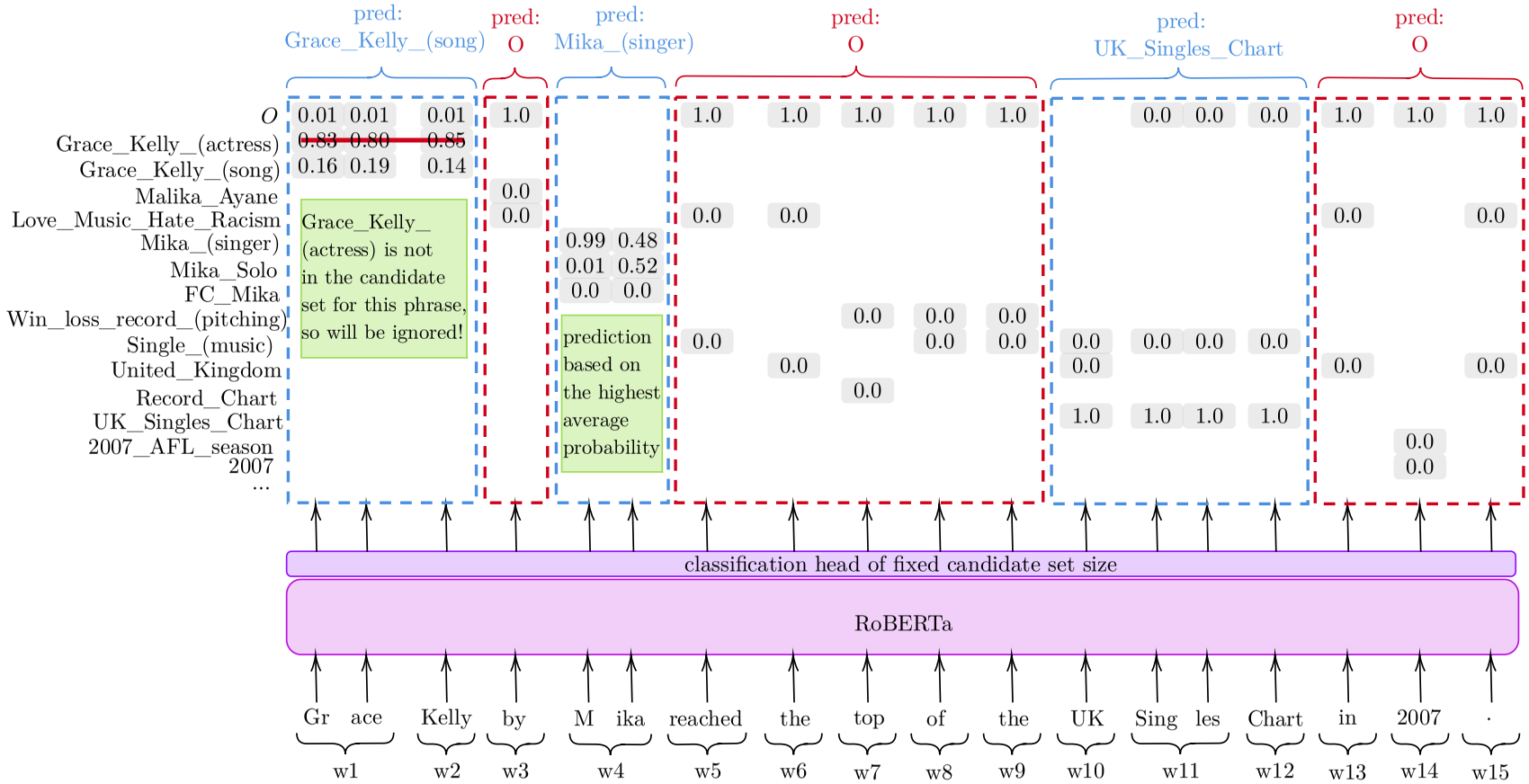
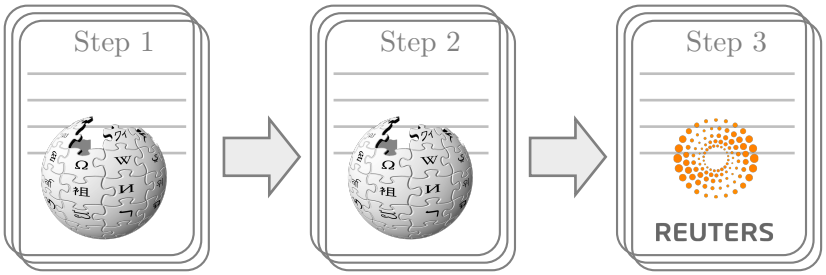


Figure 4.4: SpEL, a structured prediction modelling framework for entity linking. In this example, we demonstrate top 3 most probable entities (including `O`) for each tokenized subword.

Figure 4.5: The three fine-tuning steps proposed to tune the pre-trained language model for SpEL. Step 1: Mention-aware general knowledge fine-tuning, Step 2: Mention-agnostic general knowledge fine-tuning, Step 3: Domain specific fine-tuning.



usually is a subset of all the knowledge base entities that the model has observed in the previous two fine-tuning steps. Similar to the second fine-tuning step, we tokenize the in-domain dataset *without* the knowledge of the mention spans. Figure 4.5 demonstrates this three step fine-tuning procedure.

4.5 Experiments and Results

In this section, we discuss the data employed to fine-tune SpEL, our experimental setup, and SpEL performance evaluation experiments.

4.5.1 Data

For our experiments, we focus on Wikipedia as the knowledge base and we use the following datasets for the fine-tuning steps mentioned in Section 4.4.

Wikipedia. We use the August 20, 2023 dump of Wikipedia (with approximately 238K documents), and we use the script from Broscheit (2019) to handle incomplete annotations, perform mention-aware customized tokenization, and compute the average probability of linking to no entity (called the *Nil* probability) for the 1000 most frequent entities. The *Nil* probability is used to modify the Wikipedia training data annotations in such a way that the chance of linking a surface form referring to a frequent entity to `O` is almost 0. We construct the Wikipedia *fixed candidate set* using the union of the 500K most frequent mentions in the Wikipedia dump and the *fixed candidate set* of AIDA and the test datasets. We split the content of Wikipedia pages into chunks consisting of 254 subwords with a 20 subword overlap between consecutive chunks. After the split, our dataset contains 3,055,221 training instances with 1000 instances for validation. We also create a mention-agnostic re-tokenized version of this dataset with the same exact mentions to perform the second step of general knowledge fine-tuning as explained above.

AIDA (Hoffart et al., 2011) contains manual Wikipedia annotations for 1393 Reuters news stories originally published for the CoNLL-2003 Named Entity Recognition Shared Task (Tjong Kim Sang and De Meulder, 2003). The training and validation data is taken from the news articles published in the end of August 1996. The test data has been taken from the news stories published in December 1996. The `train`, `testa`, and `testb` splits of

AIDA contain 946, 216, and 231 documents, respectively. It has a *fixed candidate set* size of 5600 (including `O` tag) and for evaluation on the AIDA test sets, we shrink the classification head in the model to these 5600 candidates and disregard the rest of the 500K candidates[12].

4.5.2 Evaluation Using GERBIL

The GERBIL platform (Röder et al., 2018) is an evaluation toolkit (publicly available online) that eliminates any mistakes and allows for a fair comparison between methods. However, GERBIL is a Java toolkit, while most of modern entity linking work is done in Python. GERBIL developers recommend using `SpotWrapNifWS4Test`[13] (a middleware tool written in Java) to connect Python entity linkers to GERBIL. Because of the complexity of this setup, we have not been able to directly evaluate some of the earlier publications due to software version mismatches and communication errors between Python and Java. This is a drawback that discourages researchers from using GERBIL. To address this issue, we provide a Python equivalent of `SpotWrapNifWS4Test` to encourage entity linking researchers to use GERBIL for fair repeatable comparisons. We evaluate all SpEL models using GERBIL in the `A2KB` experiment type, and report InKB strong annotation matching scores for entity linking. Only five of the publications to which we compare use GERBIL, however, all report InKB strong Micro-F1 scores allowing a direct comparison to our work.

4.5.3 Setup

For the first general knowledge fine-tuning step (Section 4.4), as a warm-up to full fine-tuning, we freeze the entire RoBERTa model and only modify the classification head parameters on top of the encoder. We fine-tune with this configuration for 3 epochs and subsequently continue with fine-tuning all model parameters. We stop the fine-tuning process in this phase when the subword-level entity linking F1 score on the validation set shows no improvement for 2 consecutive epochs. Following this, we proceed to the second phase of full fine-tuning, where we adjust all model parameters using the mention-agnostic re-tokenized Wikipedia fine-tuning data. Just like phase one, we stop this phase based on the same criteria. We implement SpEL using `pytorch`, utilize `Adam` optimizer with a learning rate of $5e^{-5}$ to fine-tune the encoder parameters, and use `SparseAdam` optimizer with a learning rate of 0.01 to fine-tune the classification head. We run fine-tuning phases one and two on the large subset of Wikipedia using two Nvidia Titan RTX GPUs.

For the last phase of fine-tuning on the AIDA dataset (Section 4.4), we freeze the first four layers of the encoder (including the embedding layer) as well as the shrunk classification head parameters, and we fine-tune the rest of the model parameters for 30 epochs (over the

[12]Another implementation idea can revolve around multiplying the predicted output vector into a mask vector that masks all the candidates not in the expected 5600 entities.

[13]`https://github.com/dice-group/gerbil/tree/SpotWrapNifWS4Test/`.

`train` part of AIDA dataset). We run this step using one Nvidia 1060 with 6 GBs of GPU memory, and accumulate gradients for 4 batches before updating model parameters.

van Hulst et al. (2020) report better results using an older Wikipedia dump from 2014 compared to the dump from 2019. One possible explanation for this finding is that the 2014 dump contains Wikipedia entries with page identifiers that are more closely aligned with the annotated data. Over time, page identifiers in Wikipedia have undergone changes, and some of the older identifiers used in annotating test datasets may now function as redirect links. To tackle this issue, researchers such as Broscheit (2019) and Yamada et al. (2020, 2022) have considered redirect link normalization. We follow the same approach and use the collection of Wikipedia redirect links[14] to find all the redirect pairs (u, v) where u is not in our *fixed candidate set* and v is in the set. In inference, whenever SpEL predicts u, we automatically replace it with v.

4.6 SpEL Performance on AIDA

In this section, we conduct experiments to evaluate the performance of both SpEL-base and SpEL-large (referring to the size of the underlying RoBERTa model) in different configurations concerning the use of candidate sets (Section 4.2), and report our experimental results over the AIDA test datasets in Table 4.1.

In the first configuration, we examine our model without any *mention-specific* candidate sets. Our results show a minimum of 5.3 Micro-F1 score improvement in AIDA test sets compared to (Broscheit, 2019) while significantly reducing the required parameter size on GPU by fourfold, resulting in a 7.2 times increase in inference speed in `base` case.

Next, we run SpEL in three other configurations: (1) utilizing the KB+Yago (Ganea and Hofmann, 2017) context-agnostic candidate set, (2) employing the PPRforNED (Pershina et al., 2015) context-aware candidate set, and (3) adapting PPRforNED to aggregate the candidate information for each mention surface form, resulting in a context-agnostic candidate set, excluding context-specific information.

Candidate sets help reject many over-generated spans. If a mention's candidate set is empty, the mention span is excluded from further consideration. While this approach typically leads to improved precision and subsequent enhancement in F1 score, instances may arise where the model correctly predicts mentions that are not encompassed within the candidate sets. This can lead to lower recall in the evaluation. The observed Micro-F1 score drop when employing KB+Yago candidate sets compared to the scenario where no *mention-specific* candidate set is utilized, can be attributed to these cases.

SpEL-large using context-aware candidate sets achieves the highest boost, reporting 2.1 and 2.3 Micro-F1 scores improvement over `testa` and `testb` sets of AIDA, respectively, and establishes a new state-of-the-art for AIDA dataset. It is noteworthy to consider that the

[14] `http://downloads.dbpedia.org/2016-10/core-i18n/en/redirects_en.ttl.bz2`.

Approach		EL Micro-F1		#params on GPU	speed sec/doc
		testa	testb		
Hoffart et al. (2011)	(Linear)	72.4	72.8	-	-
Kolitsas et al. (2018)	(LSTM)	89.4	82.4	330.7M	0.097
Broscheit (2019)	(BERT)	86.0	79.3	495.1M	0.613
Peters et al. (2019)	(BERT)	82.1	73.7	-	-
Martins et al. (2019)	(Stack-LSTM)	85.2	81.9	-	-
van Hulst et al. (2020)	(LSTM)	83.3	82.4	19.0M	0.337
Févry et al. (2020)	(Transformer)	79.7	76.7	-	-
Poerner et al. (2020)	(BERT)	90.8	85.0	131.1M	-
Kannan Ravi et al. (2021)	(BERT)	-	83.1	-	-
De Cao et al. (2021b)	(BART)	90.1	83.7	406.3M	40.969
De Cao et al. (2021a)	(RoBERTa+LSTM)				
(no mention-specific candidate set)		61.9	49.4	124.8M	0.268
(using PPRforNED candidate set)		90.1	85.5	124.8M	0.194
Mrini et al. (2022)	(BART)	-	85.7	(train) 811.5M (test) 406.2M	-
Zhang et al. (2022)	(BLINK+ELECTRA)	86.8	85.8	1004.3M	-
Feng et al. (2022)	(BERT)	87.6	86.3	157.3M	-
Xiao et al. (2023)	(LLAMA-7B)	-	80.6	70000.0M	-
SpEL-base (no mention-specific candidate set)		91.3	85.5	128.9M	0.084
SpEL-base (KB+Yago candidate set)		90.6	85.7	128.9M	0.158
SpEL-base (PPRforNED candidate set)					
context-agnostic		91.7	86.8	128.9M	0.156
context-aware		92.7	88.1	128.9M	0.156
SpEL-large (no mention-specific candidate set)		91.6	85.8	361.1M	0.273
SpEL-large (KB+Yago candidate set)		90.8	85.7	361.1M	0.267
SpEL-large (PPRforNED candidate set)					
context-agnostic		92.0	87.3	361.1M	0.268
context-aware		**92.9**	**88.6**	361.1M	0.267

Table 4.1: Entity Linking evaluation results of SpEL compared to that of the literature over AIDA test sets.
#params on GPU only considers the total number of parameters that will directly effect the cost of GPU acquisition and does not reflect upon the total amount of data loaded into/from main memory and disk.

proposed model by Zhang et al. (2022) demands significant computational resources, including tens of gigabytes of RAM and over 7 and 2.7 times the number of parameters on GPU compared to SpEL-base and SpEL-large, respectively. Despite these resource-intensive requirements, SpEL outperforms (Zhang et al., 2022). The comparison between our results and that of De Cao et al. (2021a,b) demonstrates that generating entity descriptions (which can share lexical information with the mention text) is not necessary even for high accuracy Wikipedia entity linking. Our approach can be easily extended to ontologies without textual concept descriptions, while methods that generate entity descriptions cannot.

Approach	MD Micro Scores					
	testa			testb		
	P	R	F1	P	R	F1
De Cao et al., 2021a (using PPRforNED c. set)	93.9	96.7	95.2	92.2	94.8	93.5
SpEL-base (no mention-specific c. set)	94.6	94.4	94.5	92.5	90.1	91.2
SpEL-base (using PPRforNED c. set - context-agnostic)	98.3	91.6	94.8	98.3	86.4	92.0
SpEL-base (using PPRforNED c. set - context-aware)	99.4	90.9	95.0	99.4	84.9	91.6

Table 4.2: Mention Detection evaluation results of SpEL in comparison to the work of De Cao et al. (2021a) using their released evaluation code (from utils.py). As De Cao et al. (2021a) use PPRforNED candidate sets, we only compare the SpEL results calculated using these candidate sets.

Lastly, in Table 4.2, we compare SpEL-base, which utilizes the *context sensitive* prediction aggregation strategy to convert subword-level predicted entity identifiers into span-level predictions, to the model proposed by De Cao et al. (2021a). The latter model explicitly models the start and end positions of the spans for mention detection. We employ the evaluation script released by De Cao et al. (2021a) for our assessment. The results confirm that, despite not using `BIO` tags or explicitly modeling span boundaries, SpEL demonstrates strong performance in mention detection, with a high level of accuracy. Its near-perfect precision scores indicate its ability to minimize over-generated predictions, contributing to its state-of-the-art entity linking performance.

4.7 Comparison to Large Language Models

Large generative language models (Section 2.2.3) are effective zero shot and few shot learners (Brown et al., 2020) at many NLP tasks. We evaluate GPT-3.5 and GPT-4 for the task of entity linking using various prompts. For GPT-4, we consider both zero-shot and few-shot settings and we provide the few-shot prompts following the chain-of-thought (CoT; Wei et al., 2022) prompting technique. We frame the problem for the generative language model as in (De Cao et al., 2021b) to perform Wikification and produce markup around the mentions.

Table 4.3 compares the GPT evaluation results to that of SpEL. For a fair comparison, we consider the evaluation results without any *mention-specific* candidate sets. Currently the results are worse[15] than the state-of-the-art and using GPT-4 is more expensive. Further research into few-shot in-context learning on GPT-4 is likely to improve these results since LLMs have extensive knowledge about entities but cannot directly reason about specific Wikipedia URLs[16].

[15]Even considering the retrieval-augmented setting in (Xiao et al., 2023) which retrieves 100 related documents to each article and feeds them along with their annotations to the model when entity linking.

[16]Cho et al. (2022) employ GPT for entity linking by implementing a process that involves a sequence of summarization and multiple-choice queries to GPT. However, we have found this approach to be rather costly.

Approach	EL Micro-F1		US$ for 1000 docs
	testa	testb	
GPT-3.5 (zero-shot)	47.3	52.9	4.22
GPT-4.0 (zero-shot)	40.4	54.1	42.17
GPT-4.0 (few-shot w/ CoT)	62.4	66.2	59.37
Xiao et al. (2023, GPT-3-ICL)	-	60.7	-
Xiao et al. (2023, RAG-top100)	-	80.6	-
SpEL-base	91.3	85.5	2.28
SpEL-large	91.6	85.8	2.64

Table 4.3: Comparison of the performance of SpEL (in no *mention-specific* candidate set setting) to zero and few shot GPT-3.5-turbo-16k (accessed on June 16, 2023) and GPT-4-0613 (for the best performing prompts we attempted; accessed on August 24, 2023). For few-shot experiments we constructed the prompt using the chain-of-thought (CoT) method of Wei et al. (2022).

We emphasize the importance of the prompt in the performance of generative models. We examined multiple prompts with varying degrees of task explanation, both short and long. Our best-performing zero-shot prompt was:

```
You are a Wikipedia annotator. Annotate the Wikipedia entities in the
following paragraph, and produce the output in markup using the <mark>
element and the data-entity attribute:
```

In each query, we added the AIDA document received from GERBIL after the prompt and passed it to GPT. In the few-shot experiments, we followed the same procedure as zero-shot, testing various prompts. Our best-performing few-shot CoT prompt using the example document "`EU rejects German call to boycott British lamb`" follows.

```
Document: "EU rejects German call to boycott British lamb."
Answer: <p> <chain-of-thought> Considering EU, German, and British
are shown in the text together with the word boycott, this is a polit-
ical document. I should annotate EU with "European Union", German with
the country "Germany", and British with the country "United Kingdom".
I make sure I do not mistake Wikipedia identifiers with entity type
identifiers, for example I choose "United Kingdom" instead of the inc-
orrect general entity type "country". I make sure to annotate all ent-
ities even if there is a large number of entities. </chain-of-thought>
<result> <mark data-entity="European Union"> EU </mark> rejects <mark
data-entity="Germany"> German </mark> call to boycott <mark data-enti-
ty="United Kingdom"> British</mark> lamb.</result></p>
```

Furthermore, it necessitates prior knowledge of the target mention to condition the summary accordingly. Additionally, it relies on a set of candidates generated through heuristics which undermines the feasibility of utilizing GPT for end-to-end entity linking.

Adding more examples in this prompt did not significantly improve performance but substantially increased the prompting cost to GPT-4. We maintained the same configurations and setups for the few-shot experiments as in the zero-shot experiments.

We analyzed the validation set results and observed consistent patterns that shed light on the challenges posed by generative language models in entity linking. One notable observation was the presence of annotations from a mixture of knowledge bases and domains, indicating that the model possesses an excessive amount of knowledge, leading to *distractions* in the annotation process focused on entity linking over Wikipedia. With this regards, we observed a lack of stability in the model's output even when setting the `temperature` parameter to 0. Despite using the same prompt, the model occasionally confused entity linking with NER and reported mentions annotated with NER tags such as `Person` or `Location`. In our experiments, we removed all predicted spans with such tags and did not consider them in evaluation.

Furthermore, due to the nature of generative models, there were instances where the model failed to generate the complete entity, resulting in incomplete predictions (for example it generated `Leicestershire` instead of the full entity identifier `Leicestershire County Cricket Club` or `Pohang` instead of `Pohang Steelers`). In these instances, if an exact match to an entity in the knowledge base was not found, we collected all entities in the *fixed candidate set* that included the full prediction from the generative language model. In this collection, we randomly selected one of those mentions and reported it back to GERBIL instead of the original incomplete prediction generated by the model.

4.8 Practicality of the Fixed Candidate Sets

A valid concern regarding SpEL pertains to the construction of the *fixed candidate set* and its practicality in real-world scenarios, where the testing data may not be predetermined, making it challenging when attempting to assemble a subset of knowledge base entries for this purpose. As mentioned in Section 4.2, it is possible to construct this set based on the expected entities that SpEL should detect. In this section, we take a more flexible approach, and consider the entire set of 500K general fine-tuning entities as the *fixed candidate set*.

Furthermore, taking inspiration from Liu and Ritter (2023) regarding the extended existence of the CoNLL-2003 dataset, and consequently the AIDA dataset, for over two decades, we acknowledge the potential concern of adaptive overfitting. In response, we used their newly annotated NER test set of 131 Reuters news articles published between December 5th and 7th, 2020. We meticulously linked the named entity mentions in this test set to their corresponding Wikipedia pages, using the same linking procedure employed in the original AIDA dataset. Our new entity linking test set, `AIDA/testc`[17], has 1,160 unique

[17] `https://github.com/shavarani/SpEL/blob/main/resources/data/aida_testc.ttl`.

	Approach	EL Micro-F1		
		testa	testb	testc
SpEL-base	no mention-specific candidate set	89.6	82.3	73.7
	KB+Yago candidate set	89.5	83.2	57.2
	PPRforNED candidate set			
	context-agnostic	90.8	84.7	45.9
	context-aware	91.8	86.1	-
SpEL-large	no mention-specific candidate set	89.7	82.2	77.5
	KB+Yago candidate set	89.8	82.8	59.4
	PPRforNED candidate set			
	context-agnostic	91.5	85.2	46.9
	context-aware	92.0	86.3	-

Table 4.4: Entity Linking evaluation results of SpEL with a *fixed candidate set* size of 500K over AIDA test sets. Since the *context-aware* candidate sets require a mechanism for generating/looking up the candidate set during inference, we do not evaluate `testc` in this setting.

Wikipedia identifiers, spanning over 3,777 mentions and encompassing a total of 46,456 words.

We re-evaluate SpEL across all four settings outlined in Section 4.6 using the 500K entity output vocabulary and over all three AIDA test sets: `testa`, `testb`, and `testc`. We report our findings in Table 4.4. Examining the results shows that our newly created `testc` presents a new challenge for entity linking because the currently available candidate sets prove unhelpful and, in fact, detrimental to entity linking. The SpEL-large results for `testa` and `testb` show that SpEL with an unconstrained *fixed candidate set* size still matches the performance of the best model published before SpEL (with *fixed candidate set*s).

Section 6.2 will provide a unified examination of the recent entity linking methods on the newly annotated `testc`, and studies their performance in absence of candidate sets.

4.9 SpEL Performance on Out-of-domain Data

A few of the publications listed in Table 4.1 recommend assessing entity linking models on *out-of-domain* testing datasets. These datasets typically lack associated training sets and are often annotated with entity links to variations or subsets of the `DBpedia` (Auer et al., 2007) knowledge base. Out-of-domain annotation typically operates under the assumption that the knowledge base entry identifiers remain consistent between in-domain and out-of-domain scenarios. While this assumption may hold true to a certain extent, as DBpedia's primary focus has been on information extraction from Wikipedia, it's important to note that the temporal evolution of both knowledge bases has introduced discrepancies. These datasets, which are between 9 to 17 years old at the time of writing this dissertation, have been affected by temporal changes, and the two knowledge bases are not always perfectly

Approach	MSNBC	Derczynski	KORE	N^3 Reuters	N^3 RSS	OKE2015	OKE2016
Hoffart et al. (2011)†	65.1	32.6	55.4	46.4	<u>42.4</u>	<u>63.1</u>	0.0
Kolitsas et al. (2018)	72.4	34.1	35.2	<u>50.3</u>	38.2	61.9	52.7
van Hulst et al. (2020)	**74.4**	41.2	<u>61.6</u>	49.7	34.3	**64.8**	**58.8**
De Cao et al. (2021b)	<u>73.7</u>	<u>54.1</u>	60.7	46.7	40.3	56.1	50.0
Zhang et al. (2022)	72.1	52.9	**64.5**	**54.1**	41.9	61.1	51.3
SpEL-base	64.5	50.7	48.7	47.9	41.9	55.9	<u>57.4</u>
SpEL-large	63.1	**59.1**	53.7	47.1	**44.4**	59.5	56.6
Oracle‡	93.2	91.4	99.6	99.7	98.0	88.2	91.4

Table 4.5: Comparison of SpEL (with a *fixed candidate set* size of 500k) evaluation results with the literature on out-of-domain datasets. The best score is shown as bold and the second best is shown as underlined.
†Results from (Kolitsas et al., 2018 - Table 2).
‡The "Oracle" results are calculated through feeding the gold annotations of each dataset to GERBIL, and depict the In-KB annotation quality of each dataset.

aligned. The following offers a concise overview of some of the most commonly utilized out-of-domain datasets for evaluation:

MSNBC (Cucerzan, 2007) contains 20 MSNBC news stories (annotated with Wikipedia) from different categories including Health, Technology, Sports, etc.

KORE (Hoffart et al., 2012) contains 50 sentences annotated with DBpedia and chosen from five domains: celebrities, music, business, sports, and politics. It was created to examine the disambiguation functionality in the older entity disambiguation models.

N^3 Reuters and **N^3 RSS** (Röder et al., 2014) contain mentions referring to persons, places and organizations (DBpedia annotations). The Reuters dataset contains 128 news stories from Reuters news agency and the RSS dataset contains 500 RSS feed messages from worldwide newspapers (in English).

Derczynski (Derczynski et al., 2015) contains 182 tweets annotated with DBpedia knowledge base entities.

OKE challenge 2015 and 2016 evaluation sets (Nuzzolese et al., 2015) contain 101 and 55 sentences from Wikipedia articles (reporting biographies of scholars), respectively, annotated using a mixture of annotations from DBpedia and the OKE entity identifiers.

We provided the *out-of-domain* data sets to SpEL, using a *fixed candidate set* of 500K entities, and compared its performance against other methods that have reported results on these datasets. The comparative results can be found in Table 4.5.

Please note that SpEL's tokenization procedure does not allow the generation of annotations that start or end within a single word (separated by space characters). For instance, in SpEL, the token `washington-based` is considered a single word, whereas out-of-domain datasets contain several annotations that commence or conclude within a single word. Additionally, each dataset necessitates a specific redirect normalization schema; for example, `China` is annotated as `People's_Republic_of_China` in KORE, but in N^3 RSS, it is annotated as `China`.

Nevertheless, SpEL-large delivers the best results on two out of seven and the second-best result on one out of seven test sets. It doesn't significantly underperform the other models in terms of performance on the remaining four test sets.

4.10 Summary and Future Research

In this chapter, we introduced several improvements to a structured prediction approach for entity linking. Our experimental results on the AIDA dataset show that our proposed improvements to the structured prediction model for entity linking can achieve state-of-the-art results using a commonly used evaluation toolkit providing head to head numbers for competing methods on the same dataset. We showed that our approach has the best F1-score on this task compared to the state-of-the-art on this dataset. SpEL is more compute efficient with many fewer parameters and it is also much faster at inference time, providing faster throughput, compared to previous methods.

Future research may expand the scope of our investigations by exploring additional entity linking datasets spanning diverse domains, including medical NLP. Moreover, investigating the potential of multilingual applications of structured prediction for entity linking presents intriguing research pathways, examining the advantages of cross-lingual concept projection and leveraging multilingual representation learning for our entity knowledge tuned language models. Furthermore, the evolution of SpEL to accommodate zero-shot entity linking emerges as a promising domain for future exploration and advancement.

Chapter 5

SpEL for Answering Entity-Centric Questions

In this chapter, we further study the application of our proposed structured prediction based entity linking framework (Section 4.4) in the context of retrieval-augmented question answering. Notably, this chapter reproduces results which we have originally published in (Shavarani and Sarkar, 2024).

5.1 Motivation

Information retrieval has significantly enhanced the factual reliability of LLM generated responses (Shuster et al., 2021) in question answering (Zhu et al., 2021; Zhang et al., 2023). This improvement is particularly notable in a research area known as retrieval-augmented generation (RAG; Lewis et al., 2020b; Izacard and Grave, 2021a; Singh et al., 2021). RAG systems typically employ the *Retriever-Reader* architecture (Chen et al., 2017), with retrievers being either sparse (Peng et al., 2023), dense (Karpukhin et al., 2020), or a hybrid (Glass et al., 2022). The reader, which is a generative language model (e.g., GPT-3), conditions its generated answers on the documents deemed relevant by the retriever. Recent RAG methodologies exploit the in-context learning capabilities of LLMs to incorporate the retrieved documents into the prompt (Shi et al., 2023; Peng et al., 2023; Yu et al., 2023).

Kandpal et al. (2023) demonstrate that retrieval-augmentation improves LLMs' performance in answering entity-centric questions that seek factual information about real-world entities[1]. They show that this technique is particularly helpful for questions about rare entities, which appear infrequently in LLM training and fine-tuning data.

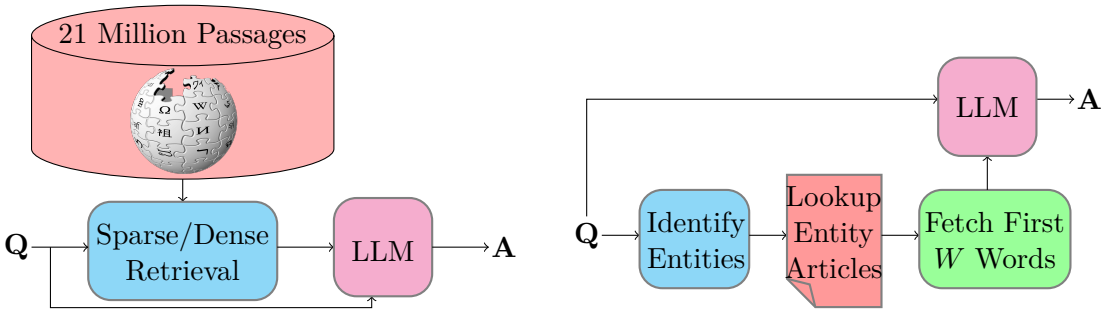
But is there a correlation between the quality of the retrieved documents and the generated response quality? Sciavolino et al. (2021) demonstrate that dense retrievers retrieve less relevant documents for answering entity-centric questions than simpler sparse retrievers. Additionally, Cuconasu et al. (2024) show that the presence of irrelevant documents leads to worse answers.

In the rest of this chapter, we introduce *Entity Retrieval* (Figure 5.1b). This method leverages salient entities in the question to lookup knowledge base (e.g., Wikipedia) articles

[1]Entity-centric questions typically have answers that are concise single words or short phrases. These answers often reference or directly stem from a knowledge base entity (Ranjan and Balabantaray, 2016).

Figure 5.1: *Entity Retrieval* simplifies the process of obtaining augmentation documents by replacing the need to search through large indexed passages with a straightforward lookup.

(a) Retrieval-augmented QA with Dense Retrieval (b) Retrieval-augmented QA with *Entity Retrieval*



corresponding to each entity and uses the first W words of these articles as augmentation documents for the question passed to the LLM. To evaluate the effectiveness of *Entity Retrieval*, we compare its retrieval performance against several passage retrieval techniques (both dense and sparse) using two entity-centric question-answering datasets. Additionally, we explore the application of entity linking, utilizing both the salient entity annotations of the questions and those identified using our proposed SpEL framework, for the *Entity Retrieval* method.

5.2 Retrieval for Retrieval-Augmentation

Retrieval-augmentation (Lewis et al., 2020b) is a method of converting Closed-book question answering[2] (Roberts et al., 2020) into extractive question answering (Abney et al., 2000; Rajpurkar et al., 2016), where the answers can be directly extracted from the retrieved documents. Even in cases that the documents do not contain the exact answer for extraction, they can function as a form of short-term memory recall and serve as potent indicators to help the model remember portions of its training data beneficial in answering the question[3]. Despite the abundance of effective retrieval techniques for retrieval-augmented question answering in existing literature (Zhan et al., 2020a,b; Yamada et al., 2021; Izacard et al., 2022; Santhanam et al., 2022; Ni et al., 2022, *inter alia.*), this section will concentrate on a select few methods[4] utilized to study answering entity-centric questions in this chapter.

[2]Closed-book QA focuses on answering questions without additional context during inference, while zero-shot QA targets answering questions on unseen topics without fine-tuning or access to examples from those topics during training.

[3]Although a careless selection of the documents may lead to distractions that worsen model performance (Cuconasu et al., 2024).

[4]We selected the methods supported by `pyserini.io` for the similarity between the underlying modules, minimizing discrepancies across different implementations.

BM25 (Robertson et al., 1994, 2009) is a probabilistic retrieval method that ranks documents based on the frequency of query terms appearing in each document, adjusted by the length of the document and overall term frequency in the collection. It operates in the sparse vector space, relying on precomputed term frequencies and inverse document frequencies to retrieve documents based on keyword matching.

DPR (Dense Passage Retrieval; Karpukhin et al., 2020) leverages a bi-encoder architecture, wherein the initial encoder processes the question and the subsequent encoder handles the passages to be retrieved. The similarity scores between the two encoded representations are computed using a dot product. Typically, the encoded representations of the second encoder are fixed and indexed in FAISS (Johnson et al., 2019; Douze et al., 2024), while the first encoder is optimized to maximize the dot product scores based on positive and negative examples.

ANCE (Xiong et al., 2021) is another dense retrieval technique similar to DPR[5]. It employs an encoder to transform both the questions and passages into dense representations. These representations are compared using dot product similarity. The key distinction from DPR is that ANCE uses hard negatives generated by periodically updating the passage embeddings during training, which helps the model learn more discriminative features, thereby enhancing retrieval performance over time.

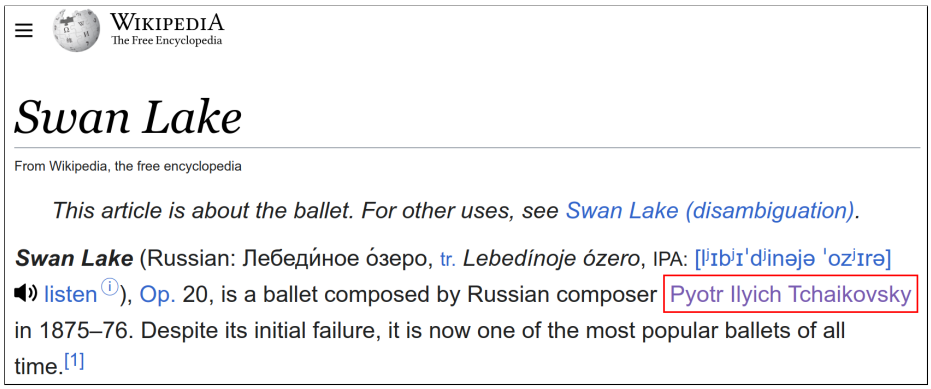
5.3 Entity Retrieval for Question Answering

While quite powerful, most retrieval-augmented systems are notably time and resource-intensive, necessitating the storage of extensive lookup indices and the need to attend to all retrieved documents to generate a response (see Section 5.4.7). This attribute renders such methods less desirable, particularly given the drive to run LLMs locally and on mobile phones (Alizadeh et al., 2023).

Entity recognition has been an integral component of statistical question answering systems (Aghaebrahimian and Jurčíček, 2016; Li et al., 2021; Adebisi et al., 2022). Additionally, the extensively studied field of Knowledge Base Question Answering (KBQA; Cui et al., 2017; Tan et al., 2023; Li et al., 2023) has underscored the significance of entity information from knowledge bases in question answering (Salnikov et al., 2023). A traditional neural question answering pipeline may contain entity detection, entity linking, relation prediction, and evidence integration (Mohammed et al., 2018; Lukovnikov et al., 2019), where entity detection can employ LSTM-based or BERT-based encoders. Inspired by this body of work, we investigate the relevance of retrieval based on entity information as an alternative strategy to the proposed retrieval methods of Section 5.2, especially for answering entity-centric questions with LLMs.

[5]We have also implemented DKRR (Izacard and Grave, 2021b), however, due to its significantly poorer performance compared to other methods, we exclude it from our analysis.

Figure 5.2: The answer to `Who is the composer of The Swan Lake ballet?` can be found in the first paragraph of `Swan Lake` Wikipedia article.



Our proposed method *Entity Retrieval*, leverages the salient entities within the questions to identify and retrieve their corresponding knowledge base articles. We will then use the first W words of these articles as the documents augmenting entity-centric questions when prompting LLMs. Figure 5.1 presents a schematic comparison between *Entity Retrieval* and dense retrieval methods in identifying retrieval documents to enhance question answering with LLMs, and Figure 5.2 provides an intuitive example to motivate the effectiveness of *Entity Retrieval*.

5.4 Experiments and Analysis

In this section, we provide our experimental setup followed by the presentation and analysis of our results.

5.4.1 Setup

We focus on Wikipedia as the knowledge base and utilize the pre-existing BM25, DPR, and ANCE retrieval indexes in Pyserini[6] (Lin et al., 2021). These indexes, follow established practices (Chen et al., 2017; Karpukhin et al., 2020) and segments the articles into non-overlapping text blocks of 100 words, resulting in 21,015,300 passages. For dense retrievers, the passages are processed with a pre-trained context encoder, generating fixed embedding vectors stored in a FAISS index (Douze et al., 2024). Our experimental entity-centric questions are encoded using the question encoder, and the top k relevant passages to the encoded question are retrieved from the FAISS index. For BM25 sparse retriever, the passages are stored in a Lucene index and the questions are keyword matched to this index.

As outlined in Section 5.3, the document retrieval process will require loading the entire index (as well as the question encoder for dense retrieval) into memory which entails significant time and memory consumption. To address this challenge, following Ram et al. (2023),

[6] `http://pyserini.io/`.

we treat document retrieval as a pre-processing step, caching the most relevant passages for each question before conducting the question answering experiments.

For *Entity Retrieval*, similar to BM25, DPR, and ANCE, we maintain document lengths at 100 words. However, our approach diverges in sourcing documents: rather than drawing from a large index of 21 million passages, we employ the salient entities within the question and retrieve their corresponding Wikipedia articles, which we then truncate to the initial 100 words[7]. Nonetheless, to explore the impact of document size, beyond the standard 100-word segment aligned with comparable methods, we investigate *Entity Retrieval* across varied lengths, including the first 50, 300, and 1000 words from the retrieved Wikipedia articles.

We conduct our retrieval-augmented question answering experiments using LLaMA-3 model, and in all such experiments[8], we prevent it from generating sequences longer than 10 subwords.

We do not use any training question-answer pairs in the prompts of our models[9]. In other words, aside from a simple instruction for answering the question, in the Closed-book setting, the prompt solely comprises the question, while in the retrieval-augmented settings using BM25, DPR, and ANCE, it includes the pre-fetched retrieved documents from the corresponding retrieval index along with the question. Similarly, for the *Entity Retrieval* settings, the prompt consists of the first W words of the Wikipedia pages corresponding to the salient entities in the question. We follow Ram et al. (2023) for question normalization and prompt formulation.

5.4.2 Data

We use the following datasets in our experiments:

EntityQuestions (Sciavolino et al., 2021) is created by collecting 24 common relations (e.g., 'author of' and 'located in') and transforming fact triples (subject, relation, object) that contain these relations, into natural language questions using predefined templates. The dataset comprises 176,560 train, 22,068 dev, and 22,075 test question-answer pairs. To expedite our analytical experiments in this chapter, given the extensive size of the dev and test sets, we constrain the question-answer pairs in these subsets to those featuring salient entities within the top 500K most linked Wikipedia pages, as suggested in Chapter 4. Thus, the dev and test subsets of EntityQuestions considered in our experiments consist of 4,710 and 4,741 questions, respectively.

[7]The first sentences of Wikipedia articles have been demonstrated to be effective for document classification (Section 6.1) as well as question answering (Choi et al., 2018).

[8]We run our experiments on one server containing 2 RTX A6000s with 49GB GPU memory each.

[9]Further exploration into few-shot experimental setups involving additional (context, question, answer) in-context examples is left for future investigation.

FactoidQA[10] (Smith et al., 2008) contains 2,203 hand crafted question-answer pairs derived from Wikipedia articles, with each pair accompanied by its corresponding Wikipedia source article included in the dataset.

StrategyQA[11] (Geva et al., 2021) is a complex boolean question answering dataset, constructed by presenting individual terms from Wikipedia to annotators. Its questions contain references to more than one Wikipedia entity, and necessitate implicit reasoning for binary (Yes/No) responses. The dataset comprises 5,111 answered questions initially intended for training question answering systems, with the system later tested on test set questions with unreleased answers. This training set is split into two subsets, based on the perceived challenge of questions by adversarial annotation models (Dua et al., 2019), resulting in `train` and `train_filtered` subsets containing 2,290 and 2,821 questions, respectively.

5.4.3 Evaluation

We evaluate the performance of the retrieval methods using the following metrics:

- nDCG@k (normalized Discounted Cumulative Gain at rank k; Järvelin and Kekäläinen, 2002) evaluates the quality of a ranking system by considering both the relevance and the position of documents in the top k results. Mathematically, it is represented as

$$\text{nDCG@}k = \frac{\sum_{i=1}^{k} \frac{2^{r_i}-1}{\log_2(i+1)}}{\sum_{i=1}^{|REL_k|} \frac{2^{r_i}-1}{\log_2(i+1)}}$$

  Where, r_i denotes the relevance score of a document for a question, with relevance score $r_i = 1$ if the document contains a normalized form of the expected answer to the question, and $r_i = 0$, otherwise. And, REL_k refers to a subset of the retrieved documents that contain a normalized form of the expected answer. nDCG@k scores range between 0 and 1, where a score of 1 signifies an optimal ranking with the most relevant documents positioned at the top.

- MRR (Mean Reciprocal Rank; Voorhees and Harman, 1999) is the average of the reciprocal ranks of the first relevant document for each question. Mathematically, it is represented as

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{r_i}$$

  where $|Q|$ represents the total number of questions and r_i denotes the rank of the first relevant document for the i-th question.

[10] `https://www.cs.cmu.edu/~ark/QA-data/data/Question_Answer_Dataset_v1.2.tar.gz`.

[11] `https://allenai.org/data/strategyqa`.

- Top-k Retrieval Accuracy, as reported by Sciavolino et al. (2021), is calculated as the number of questions with at least one relevant document in the top k retrieved documents divided by the total number of questions in the dataset.

We evaluate the performance of the retrieval-augmented question-answering models with each retrieval method as follows:

- For FactoidQA and EntityQuestions datasets, we use `OpenQA-eval` (Kamalloo et al., 2023) scripts to evaluate model performance, and report exact match (EM) and F1 scores by comparing expected answers to normalized model responses.

- For StrategyQA, we present accuracy scores by comparing model responses to the expected boolean answers in the dataset. As well, to assess model comprehension of the task, we count the number of answers that deviate from Yes or No and report this count in a distinct column labeled "`Inv #`" for each experiment.

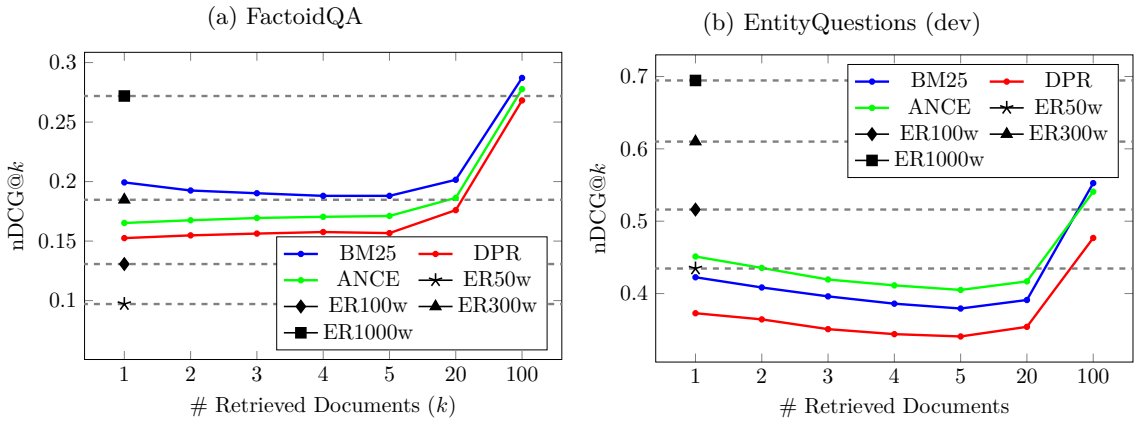
5.4.4 Entity Retrieval Performance using Question Entity Annotations

We begin our analysis by comparing *Entity Retrieval* performance to BM25, DPR, and ANCE. For this experiment, we calculate nDCG with various retrieved document sets of size $k = 1, 2, 3, 4, 5, 20$, and 100 documents. We use the entity annotations provided with the questions from FactoidQA and the dev set of EntityQuestions to fetch their corresponding Wikipedia articles, excluding StrategyQA from our analysis as it does not include entity annotations. On average, FactoidQA and EntityQuestions datasets contain one salient entity per question.

To evaluate the effect of document length, we compare *Entity Retrieval* with the first 100 words (equivalent to the size of documents returned by BM25, DPR, and ANCE; noted as *ER100w*) and also consider the first 50, 300, and 1000 words of the retrieved Wikipedia articles (noted as *ER50w*, *ER300w*, and *ER1000w*). An *Entity Retrieval* document with 300 words has the same word count as three documents returned by BM25 or DPR.

Figure 5.3 presents the computed nDCG@k scores across varying document sizes, highlighting the superior performance of *Entity Retrieval* over other retrieval methods in the context of the entity-centric datasets under study. Notably, *ER1000w*, which corresponds to ten BM25 retrieved passages in terms of word count, exhibits a retrieval performance on par with 100 retrieved documents in FactoidQA and surpasses BM25, the top-performing retriever on EntityQuestions, by 25%. This impressive performance by *Entity Retrieval* can be attributed to its ability to retrieve fewer, yet more relevant, documents. This observation aligns with the conclusion drawn by Cuconasu et al. (2024), which emphasizes that the retrieval of irrelevant documents can negatively impact performance. *Entity Retrieval* effectively minimizes the retrieval of such documents. Further insights can be gleaned from the comparison of nDCG scores along the x-axis of the plots in Figure 5.3. As the number of

Figure 5.3: nDCG@k scores comparing the quality of BM25, DPR, ANCE, and *Entity Retrieval* by considering both the relevance and the position of documents in the top k retrieved passages for each question.



retrieved documents increases, the likelihood of retrieving irrelevant documents also rises, leading to a decline in retrieval performance when moving from 1 to 5 retrieved documents.

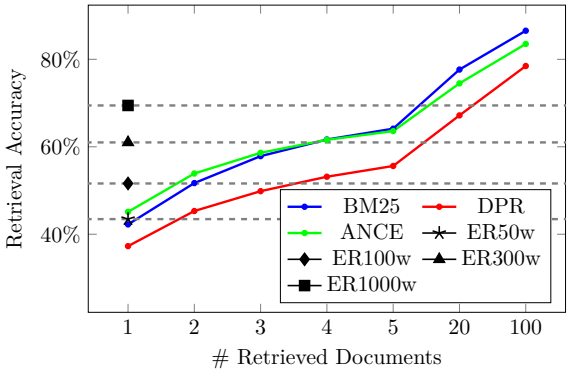
Table 5.1 showcases the calculated MRR scores, emphasizing the quicker attainment of relevant retrieval documents in *Entity Retrieval* compared to other retrieval methods. Concurrently, Figure 5.4 illustrates the impact of incrementing the number of retrieved documents on the expansion of the expected answers' coverage for the EntityQuestions dev subset.

	FactoidQA	**EntityQuestions (dev)**
BM25	0.245	0.522
DPR	0.209	0.456
ANCE	0.222	0.536
ER50w	0.097	0.435
ER100w	0.131	0.516
ER300w	0.185	0.610
ER1000w	**0.272**	**0.695**

Table 5.1: MRR scores comparing the retrieval quality of BM25, DPR, ANCE, and *Entity Retrieval* through the average of the reciprocal ranks of the first relevant document for each question.

While it may be appealing to consider 100 or more documents to simultaneously enhance both nDCG and Retrieval Accuracy, it is important to note that 100 retrieved documents would comprise 10,000 words. This could potentially overwhelm the model with excessive noise (irrelevant documents), and as well, could make it extremely costly to execute retrieval-augmented question answering, especially when the cost of API calls is calculated per token. We would need at least 10,000 tokens (optimistically, assuming each word equates to only one token) in addition to the tokens in the question. These factors suggest that retrieving a few documents for each question is more beneficial.

Figure 5.4: Retrieval Accuracy scores showcasing the correlation between the number of retrieved documents and the expected answers' coverage in EntityQuestions (dev) subset.



Taking these considerations into account, along with the nDCG@k, MRR, and Retrieval Accuracy results from this section, we gain a comprehensive understanding of the trade-off between the quality of the retrieved documents, which diminishes as we consider more documents, and the answer coverage, which increases as the model has a higher chance of encountering the right document with the correct hint for the answer. Consequently, we opt for $k = 4$ as a default, and we will always retrieve the top-4 documents in our retrieval-augmented question answering experiments.

5.4.5 Retrieval-Augmented Question Answering

Next, we shift our focus to study the effectiveness of our proposed *Entity Retrieval* method compared to other retrieval methods in enhancing the quality of responses to entity-centric questions. In this section, we examine three distinct scenarios: (1) the Closed-book setting, where we use "`Answer these questions:`" as the task instruction, followed by the question, (2) the Retrieval-Augmented setting, where we use retrieved documents as a basis, followed by "`Based on these texts, answer these questions:`", and then the question, and (3) the *Entity Retrieval* with question entity annotations, which uses the same prompt as the retrieval-augmented setting. The only difference lies in the documents retrieved, as we have previously discussed.

The initial eight rows of Table 5.2 present the results of our experiments using LLaMA-3 (8B) model. Upon examining these results, it is evident that *ER100w*, the most analogous *Entity Retrieval* setting to other retrieval methods, outperforms in terms of both EM and F1 scores. This setting returns 100-word documents as the other retrieval methods. Furthermore, our dense retrieval results align with the observations of Sciavolino et al. (2021), asserting that entity-centric questions indeed challenge dense retrievers. Although the BM25 method proves successful in enhancing the results compared to the Closed-book setting, it is noteworthy that even *Entity Retrieval* with the initial 50 words of the articles corresponding to the salient entities within questions yields superior results. This is particularly significant when compared to other retrieval methods which necessitate indexing the entire knowledge

LLaMA-3 (8B)*	FactoidQA		EntityQuestions			
			dev		test	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**
Closed-book	30.7±0.1	39.3±0.0	22.7±0.5	37.8±1.0	22.8±0.1	38.1±0.6
Retrieval-Augmented QA						
BM25	32.2±1.1	42.4±0.2	23.8±0.3	38.6±0.8	23.3±0.0	38.5±0.1
DPR	29.4±1.0	38.5±1.2	22.0±0.1	36.2±0.2	20.5±0.4	35.3±0.6
ANCE	30.5±0.4	40.0±0.4	23.1±0.7	37.9±0.6	22.7±0.7	37.9±0.9
Entity Retrieval w/ Question Entity Annotations						
ER50w	34.2±0.7	43.5±0.6	24.9±0.2	41.2±0.0	23.9±0.5	41.0±0.1
ER100w	33.6±0.5	42.8±0.5	**26.2**±0.0	**42.8**±0.1	**25.7**±0.1	**42.4**±0.0
ER300w	33.7±1.4	43.0±1.7	**26.2**±0.4	**42.8**±0.0	25.3±1.0	**42.4**±1.1
ER1000w	**35.1**±0.4	**44.9**±0.7	25.2±0.4	41.9±0.6	24.5±0.9	41.3±0.6
Entity Retrieval w/ SpEL Identified Entity Annotations						
ERSp50w	29.7±0.3	38.6±0.7	24.3±0.2	39.2±0.1	24.0±0.1	39.7±0.0
ERSp100w	28.3±0.9	37.4±1.2	25.0±0.4	40.1±0.3	24.2±0.2	39.8±0.1
ERSp300w	26.8±0.6	35.6±0.7	24.4±0.0	39.7±0.1	24.6±0.3	40.2±0.5
ERSp1000w	21.3±0.5	30.4±0.8	24.4±0.1	39.7±0.1	23.0±0.7	39.2±0.7

Table 5.2: Question answering efficacy comparison between Closed-book and Retrieval-augmentation using BM25, DPR, ANCE, and *Entity Retrieval*. EM refers to the exact match between predicted and expected answers, disregarding punctuation and articles (`a`, `an`, `the`).
* Results represent the average of two runs, accompanied by a margin of error based on a 99% confidence interval.

base on disk and loading the index into memory; a process required in inference time where caching is not an option.

5.4.6 Entity Retrieval in absence of Question Entity Annotations

In this section, we concentrate on the most crucial component of the *Entity Retrieval* method: the salient entities within entity-centric questions. We explore a scenario where the entities are not explicitly provided in the question, suggesting the use of an entity linking method to extract these entities. Ideally, we would like to evaluate all recent entity linking methods to identify the most effective one. However, due to time and budget limitations, we depend on the findings of Section 6.2 to choose an entity linking method. In this section, we examine the latest entity linking methods in terms of performance against unseen data and find SpEL as the top performer. Consequently, we investigate *Entity Retrieval* using entities identified with SpEL, while reserving the examination of other entity linking techniques for *Entity Retrieval* for future research.

We maintain the *Entity Retrieval* settings as before, defining *ERSp50w*, *ERSp100w*, *ERSp300w*, and *ERSp1000w* for performing entity linking with SpEL, then retrieving the

Question	Who performed Alexis Colby?	What is the capital of Seine-Saint-Denis?
Answer	Joan Collins	Bobigny
Closed-Book	Diana Ross	Paris
BM25	Linda Evans	Saint-Denis
DPR	Alexis Cohen	Saint-Denis
ANCE	Nicollette Sheridan performed Alexis Colby.	Saint-Denis
ERSp100w	Joan Collins	Bobigny
Question	Where did John Snetzler die?	Where was Brigita Bukovec born?
Answer	Schaffhausen	Ljubljana
Closed-Book	He died in London, England, in 178	Brigita Bukovec was born in Slovenia
BM25	John Snetzler died in London.	Slovenia
DPR	John Snetzler died in London	in Slovakia
ANCE	in England	Rîbnița
ERSp100w	Schaffhausen	Ljubljana

Table 5.3: Example questions from EntityQuestions (dev) to demonstrate the performance of *Entity Retrieval* in comparison to the other retrieval methods.

Wikipedia articles corresponding to the SpEL identified entities, and using the first 50, 100, 300, and 1000 words of these articles as documents to augment the question when prompting the LLM.

Passing the questions from our datasets to SpEL for analysis, we find that it generates a maximum of 8, 3, and 4 annotations for FactoidQA, EntityQuestions, and StrategyQA, respectively. On average, it produces 0.8, 0.7, and 1.1 annotations per question for these same datasets. SpEL successfully identifies and links entities in 56.5% of FactoidQA questions (1244/2203), 66.0% of EntityQuestions (dev) questions (3108/4710), 65.3% of EntityQuestions (test) questions (3095/4741), 75.8% of StrategyQA (train) questions (1735/2290), and 74.2% of StrategyQA (train_filtered) questions (2094/2821).

The final four rows of Table 5.2 showcase the comparative results of utilizing entities identified by SpEL for *Entity Retrieval*. Given that one-third of EntityQuestions and approximately half of FactoidQA lack identified annotations, the exact match scores reveal that *Entity Retrieval* performs robustly and surpasses BM25, the top-performing competitor retrieval method, for the entity-centric question-answering datasets under examination. This underscores the potential of *Entity Retrieval* within this paradigm. In addition, the disparity between the results with and without question entity annotations strongly indicates the necessity for further research in the Entity Linking domain, which could enhance entity-centric question answering as a downstream task. Table 5.3 provides some example questions where *Entity Retrieval* has led to better answers.

Table 5.4 presents a comparison of the performance of *Entity Retrieval* using SpEL identified entities against other retrieval methods on the StrategyQA dataset. The results clearly demonstrate the superior performance of *Entity Retrieval* over the top-performing retrieval methods as shown in Table 5.2. It is important to note that the 100-word setting (*ERSp100w*) is the most analogous to other retrieval methods, given that the size of their

LLaMA-3 (8B)*	train		train_filtered	
	Acc.	**Inv #**	**Acc.**	**Inv #**
BM25	43.8±0.1	601±4	49.1±1.0	679±7
ANCE	47.0±1.2	550±15	51.8±1.0	637±42
ERSp50w	49.7±1.2	378±34	56.2±1.3	417±31
ERSp100w	**50.5**±2.0	**367**±21	**56.6**±0.5	**389**±1
ERSp300w	46.2±1.9	508±22	53.9±1.9	538±14
ERSp1000w	40.2±0.4	778±3	43.2±0.3	924±13

Table 5.4: Comparison of *Entity Retrieval* using SpEL identified entities to the best-performing dense and sparse retrieval methods of Table 5.2 on the StrategyQA dataset. Given the expected boolean results for StrategyQA questions, we restricted LLaMA-3 to generate only one token. *Acc.* indicates the fraction of answers that correctly match the expected Yes or No responses in the dataset, while *Inv #* represents the count of labels that are neither Yes nor No, but another invalid answer.

* Results represent the average of two runs, accompanied by a margin of error based on a 99% confidence interval.

retrieved documents is also 100 words. Interestingly, the results from the 1000-word setting suggest that longer documents do not necessarily enhance the model's recall. In fact, beyond a certain length, the model may become overwhelmed by the sheer volume of noise, leading to confusion. Lastly, the invalid count values suggest that *Entity Retrieval* is more effective in assisting the model to comprehend the boolean nature of expected responses, eliminating the need to rely on retrieval from millions of passages.

5.4.7 Real-time Efficiency Analysis

Our analysis thus far has primarily focused on the retrieval performance, without consideration for the time and memory efficiency; crucial factors in retrieval method selection. In this section, we shift our focus to these aspects.

We begin by replacing the pre-built cache with the original retrieval modules that were used in creating the retrieval cache document sets. We load the indexes and the necessary models for fetching the retrieval documents. We then record the peak main memory requirement of each method during the experiment. It is important to note that all retrieval methods primarily rely on main memory, with minimal differences in GPU memory requirements. Therefore, we report an average GPU memory requirement of 35GB for the LLaMA-3 (8B) setting and exclude it from our results table. We then feed all 2,203 FactoidQA questions into the BM25, ANCE, and *Entity Retrieval* (using SpEL identified entities) to fetch the top-4 documents. We report the total time taken to generate answers to all the questions. Additionally, we keep track of all the pre-built models and indexes that each method requires for download and storage. We report the total size of all downloaded files to disk.

Table 5.5 presents our findings on time and memory requirements. It is evident that ANCE requires significantly more time to fetch and provide documents, six times more disk

	Total Time	Disk Storage	Main Memory
BM25	45min	11GB	2.3GB
ANCE	960min	61.5GB	64.2GB
ERSp100w	34min	9.4GB	6.3GB

Table 5.5: Comparison of the required resources for each retrieval method in real-time execution. The reported total time values exclude the time taken to load the indexes and models, focusing solely on the time used to answer the questions.

space to store its indexes, and over ten times higher main memory demands to load its dense representations[12]. In contrast, BM25 and *Entity Retrieval* are more resource-friendly. Notably, *Entity Retrieval* is 25% faster than BM25 in response generation while demanding the total memory and disk space of a standard personal computer. Future research can be directed towards reducing the memory requirements of *Entity Retrieval*; a direction which we find quit promising.

5.5 Related Studies

Similar to our work, Kandpal et al. (2023) investigate the impact of salient entities on question answering, and propose constructing oracle retrieval documents as the 300-word segment surrounding the ground-truth answer from the Wikipedia page that contains the answer (entity name). Our approach leverages salient entities from questions without directly involving answers. Additionally, they primarily use entities to classify questions into those concerning frequent knowledge base entries versus those about rare entries on the long-tail, whereas our approach assigns a more substantial role to entities, treating them as pointers guiding the retrieval of relevant documents to augment questions.

Sciavolino et al. (2021) compare DPR and BM25 retrievers for entity-centric questions, and demonstrate that DPR greatly underperforms BM25. They attribute this to dense retrievers' difficulty with infrequent entities, which are less represented in training data. In contrast, BM25's frequency-based retrieval is not sensitive to entity frequency. We take a parallel approach and propose a simple yet effective method that leverages salient entities in the question for identifying augmentation documents.

5.6 Summary and Future Research

In this chapter, we focused on retrieval-augmented question answering, and explored various retrieval methods that rely on the similarity between the question and the content of the passages to be retrieved. We introduced a novel approach, *Entity Retrieval*, which deviates from the conventional text similarity measure to identify relevant passages. Instead, it capitalizes on the salient entities within the question to identify retrieval documents. Our

[12]Our empirical results demonstrate that DPR follows the same trend.

findings indicate that our proposed method is not only more accurate but also faster in the context of entity-centric question answering.

Future research could delve into the application of *Entity Retrieval* in few-shot question answering, and examine the impact of different entity linking models on *Entity Retrieval*. Additionally, future studies could investigate the feasibility of considering all entities with a high degree of surface form overlap with linked entities to obtain augmentation documents. This could potentially address any ambiguities in identified entity links.

Chapter 6

Other Contributions

In this chapter, we discuss other contributions that either diverge from the primary focus of the dissertation or involve shared authorship, even if they pertain to the central theme of the dissertation. Each section will isolate the topic for discussion and provide references to their original publications.

6.1 Multi-class Multilingual Classification of Wikipedia Articles Using Extended Named Entity Tag Set

This section reproduces results which we have originally published in (Shavarani and Sekine, 2020).

Wikipedia serves as a valuable repository of global knowledge. Establishing an interconnected taxonomy within Wikipedia entities requires significant organizational efforts. Sekine et al. (2018b) propose structuring Wikipedia articles to include recognized entities and associated attributes, facilitating interlinking between attributes. The initial step involves categorizing entities into predefined classes and validating the results through human annotation, a crucial aspect in ensuring the accuracy of the knowledge base.

Over the years, numerous attempts have been made to categorize Wikipedia articles into various sets typically comprising 3 to 15 class types (Toral and Munoz, 2006; Watanabe et al., 2007; Dakka and Cucerzan, 2008; Chang et al., 2009; Tardif et al., 2009). However, such classification schemes offer limited utility when utilized as training data for question answering systems due to the lack of detailed information within the extracted knowledge base. Conversely, broader categorization sets like Cyc-Taxonomy (Lenat, 1995), Yago-Taxonomy (Suchanek et al., 2007), or Wikipedia's taxonomy of categories (Schönhofen, 2009) present challenges for classifying Wikipedia articles as the tags lack verifiability for annotators. Moreover, these taxonomies, often lacking a hierarchical tree structure, complicate the verification process, especially for articles covering multiple topics.

Addressing these challenges, the *Extended Named Entities Hierarchy* (ENE; Sekine et al., 2002) emerges as a promising tag set, offering 200 fine-grained categories tailored for Wikipedia articles. Higashinaka et al. (2012) pioneer the utilization of this extended tag set as output labels for categorizing Wikipedia pages, employing a hand-extracted feature set to convert pages into model-compatible input vectors. Building upon this, Suzuki

et al. (2016) augment the input features with trained vectors representing links between Wikipedia pages, proposing a more intricate model for mapping articles to labels, albeit without exploring the multilingual aspect of Wikipedia articles.

The work described in this section builds upon Sekine et al. (2018a)'s efforts, where linguists were employed as annotators and trained on the ENE tag set to annotate each article with up to 6 different ENE classes. We leverage Wikipedia language links to create a multi-lingual Wikipedia classification dataset. Subsequent subsections detail the dataset creation process, multi-lingual feature selection methods, model descriptions, experimental setup, and classification results, benchmarked against existing methodologies proposed by Higashinaka et al. (2012) and Suzuki et al. (2016).

6.1.1 Dataset Collection and Annotation

Sekine et al. (2018a) curated an annotated dataset containing 782,517 Japanese Wikipedia articles spanning various domains, encompassing 175 out of 200 ENE labels. The selection criteria omitted certain categories due to a lack of qualifying articles at the time. Articles were sourced from Japanese Wikipedia, requiring a minimum of 5 hyperlinks from other Wikipedia articles. Annotators, predominantly possessing post-secondary degrees in linguistics, were tasked with assigning up to 6 labels from the suggested 200 ENE labels. Although inter-annotator agreement data is unavailable, the quality of annotations was validated through random sampling and assessment by proficient annotators. This dataset is accessible for the SHINRA2020-ML classification task[1].

We restricted our analysis to a subset of annotated articles, aligning with recommendation from Suzuki et al. (2016), which stipulated a minimum of 100 hyperlinks per article. This criterion yielded 118,635 Japanese Wikipedia articles, annotated with 164 out of 200 ENE labels, with a maximum of 5 annotations per article. To expand our dataset, we gathered the content corresponding to the same article titles from English, French, German, and Farsi Wikipedia. Leveraging Wikipedia language links, which connect articles on identical entities across languages, we accessed the May 20, 2018 snapshot of Wikipedia in all five languages. We applied the labels assigned to the Japanese articles to their counterparts in the other languages, capitalizing on the language-agnostic nature of ENEs and the consistent content across pages.

To initiate our language link exploration, we constructed a comprehensive graph of language links encompassing all (`wikipedia_id`, `language`) pairs, facilitating connections between articles across the five languages. Additionally, we accounted for Wikipedia redirect links in our exploration to accommodate instances where language links direct to redirect pages in other languages. Utilizing this language links graph, we organized *Entities* by group-

[1] `http://shinra-project.info/shinra2020ml/`.

Language	Documents	Classes	Average		Max Ann.
			Articles/Class	Ann./Article	Count
Japanese	118,635	164	742.5	1.0357	5
English	52,445	159	339.9	1.0357	5
French	34,432	156	227.2	1.0346	5
German	29,808	154	198.6	1.0306	5
Farsi	14,058	148	97.7	1.0335	5

Table 6.1: Statistics about the collected *Shinra 5-Language Categorization Dataset*.

ing various (`wikipedia_id`, `language`) pairs that denote the same subject. Subsequently, we assigned the ENE labels to articles across different languages, ensuring consistent labeling.

We created the *Shinra 5-Language Categorization Dataset* (SHINRA-5LDS[2]) as a comprehensive collection of multilingual, multi-labeled Wikipedia articles. This dataset facilitates benchmarking on multi-labeled Japanese, English, French, German, and Farsi Wikipedia categorization using various methodologies proposed by researchers. Table 6.1 presents statistics detailing the total number of annotated articles in each language, the total count of ENE classes with at least one annotated article, the average number of articles per class, and the average number of annotations per article provided by annotators.

6.1.2 Feature Selection and Models

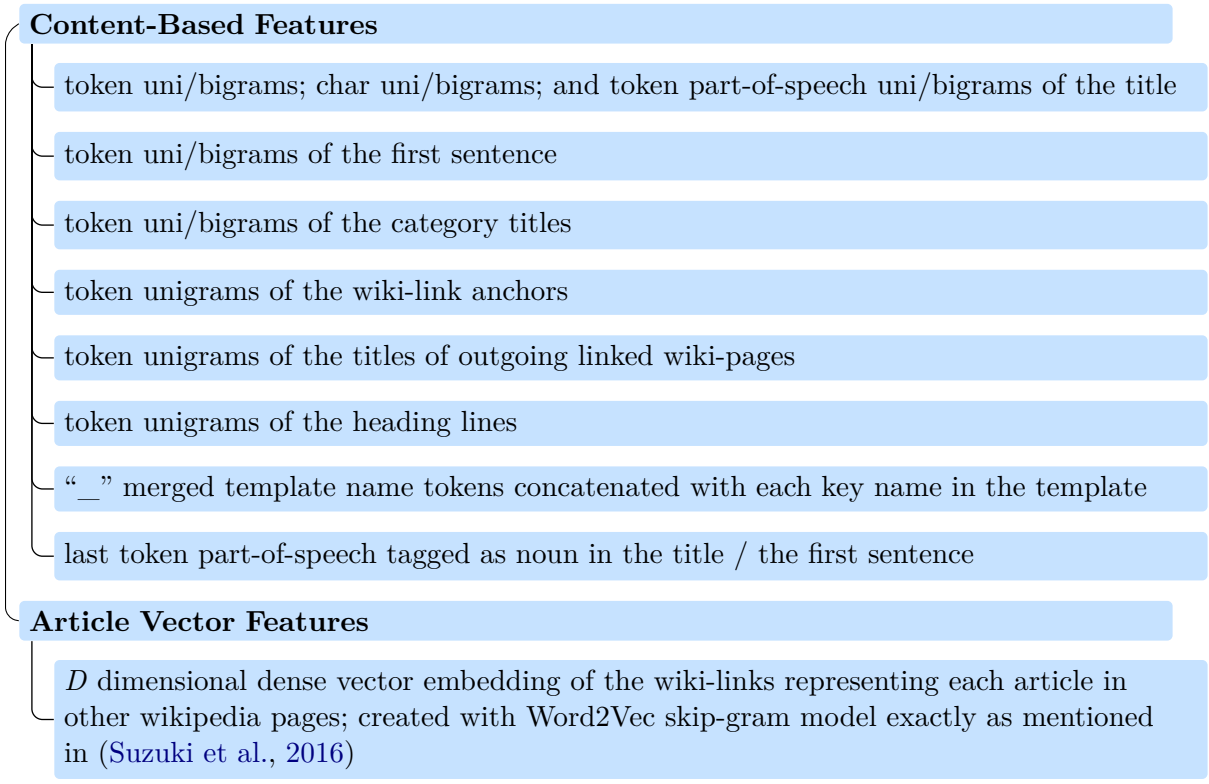
For benchmarking purposes, we reviewed existing models proposed for multi-class categorization of Wikipedia articles and opted for methodologies suggested by Higashinaka et al. (2012) and Suzuki et al. (2016), both of which advocate for classifying articles using ENE tag set. Additionally, to assess the efficacy of hierarchical structures in classifier training with ENEs, we included models proposed by Wehrmann et al. (2018) in our analysis.

Feature Selection. Ensuring a fair comparison among the models necessitates standardized inputs across all methodologies. To achieve this, we amalgamated feature selection methods advocated in previous studies (Wang and Manning, 2012; Higashinaka et al., 2012; Suzuki et al., 2016), forming a unified set of features. However, due to the multilingual context of our task, certain features such as *Last one/two/three characters in the headings or titles* and *Last character type (Hiragana/Katakana/Kanji/Other)* had to be excluded from the union. Figure 6.1 provides an overview of the final unified schema delineating the categorization features for Wikipedia articles in the SHINRA-5LDS.

Binary Logistic Regression. Higashinaka et al. (2012) proposed using a set of distinct *Binary Logistic Regression Classifier* models to distinguish the contribution of extracted fea-

[2] `https://huggingface.co/datasets/shavarani/SHINRA-5LDS`; the articles in this repository are updated to contain the Wikipedia content from April 2024, and Table 6.1 reflects on the statistics on this updated version.

Figure 6.1: Unified categorization feature extraction schema from Wikipedia articles.



tures to the final selected class. We adopt this approach to assess the classification difficulty level of our dataset using a simple model.

Multi-task Regression. Suzuki et al. (2016) proposed aggregating separate Logistic Regression Classifier models into a *2-Layer Perceptron Neural Network* (referred to as *Multi-task Regression*), aiming to enhance information capture for more confident assignment of ENE classes to articles. Their study concludes that *Multi-task Regression* effectively learns feature-label correlations compared to separate logistic regression models or isolated 2-Layer Perceptron Networks. We implement their model and explore its performance further by augmenting it with an additional layer in our benchmark experiments.

Hierarchical Multi-Label Classification Networks. To explore the information contained within the Hierarchy of ENEs, we propose utilizing *Hierarchical Multi-Label Classification Networks* (HMCN). Wehrmann et al. (2018) outline two distinct configurations for HMCNs, both employing a top-down approach for predicting the label hierarchy. The first configuration, *HMCN Feed-forward (HMCN-F)*, utilizes dedicated segments of the network for predicting each hierarchy level, while *HMCN Recurrent (HMCN-R)* iteratively incorporates previous top layer predictions into subsequent lower-level predictions. We employ both *HMCN-R* and *HMCN-F* to investigate the impact of model compression on hierarchy prediction during testing.

Model	Japanese	English	German	French	Farsi
Binary Logistic Regression	71.2	74.5	69.4	67.8	73.1
Multi-task Regression (2L)†	**78.8**	78.3	**81.5**	80.0	78.0
Multi-task Regression (3L)	77.6	**81.0**	79.9	**83.5**	**82.5**
HMCN-F	71.7	73.5	70.6	71.9	76.0
HMCN-R	61.5	63.7	63.2	64.7	70.3

Table 6.2: 5-fold cross validation classification accuracy of the predicted labels for the fine-grained labels in SHINRA-5LDS dataset.
† While we aimed to maintain settings comparable to their model, a fair comparison between our results and theirs is unfair due to disparities in dataset size and class numbers between our experiments and theirs.

Training and Evaluation. For multi-label classification, we pass the predicted membership distributions through a Sigmoid layer and assign a label to the article if the resulting probability, post-Sigmoid transformation, exceeds 0.5.

Evaluation is based on the micro-averaged precision (Sorower, 2010) of the predicted labels in the last level of hierarchy. To mitigate the influence of more frequent classes during training, we use weighted gradient back-propagation (He and Garcia, 2009). The weight for each article is calculated as $w = \frac{N}{\sum_{n=1}^{N} c(l_n)}$, where N represents the number of labels assigned to the article (capped at 6), and $c(l_n)$ counts the total number of training set articles associated with label l_n. The loss function employed for training all models is the *Binary Cross Entropy* loss, averaged across all possible classes, to ensure comparability with previous work.

6.1.3 Experiments and Results

We implemented all models outlined in Section 6.1.2 using `pytorch`. For part-of-speech tagging, as well as article normalization and tokenization, we used the Hazm[3] Toolkit for Farsi, the Mecab Toolkit (Kudo, 2006) for Japanese, and the TreeTagger[4] Toolkit for English, French, and German.

Across all experiments, we employed the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $1e^{-3}$ and applied gradient clipping (Pascanu et al., 2013) at 5.0. Network parameters were initialized randomly between $(-0.1, 0.1)$, and training utilized mini-batches of size 32 over 30,000 steps (batches) of randomly shuffled training instances. The hidden layer size for all models was set to 384[5]. Evaluation was conducted via 5-fold cross-validation,

[3] `https://github.com/sobhe/hazm`.

[4] `https://github.com/miotto/treetagger-python`.

[5] Larger hidden layer sizes were explored but yielded marginal differences in results, thus not impacting our experiment analysis.

allocating 80% of the data for training, and 20% for testing. Additionally, classes with a frequency less than 20 in the dataset were excluded from training/testing.

Table 6.2 illustrates the micro-averaged precision of article classification in SHINRA-5LDS dataset. Initial results indicate the dataset's complexity, as evidenced by the Binary Logistic Regression model's modest accuracy. Notably, Japanese exhibits lower scores, suggesting heightened classification difficulty across all models.

Furthermore, the consistent outperformance of non-hierarchical models over hierarchical ones suggests that leaf-node ENEs contain sufficient information for classification, and hierarchy may introduce ambiguity. The overall precision scores highlight current model limitations with larger, more complex annotated article sets.

6.2 Unified Examination of Entity Linking in Absence of Candidate Sets

This section reproduces results which we have originally published in (Ong, Shavarani, and Sarkar, 2024).

Unified evaluation of the different entity linking systems with respect to the application of candidate sets (Section 4.2) should play a crucial role in a better understanding of the strengths and weaknesses of each system. This will give the research community and commercial deployments better ways to select the most suitable system based on their needs while providing them a platform to identify avenues for enhancement. In this section, we unify the evaluation setup for the systems using GERBIL (Röder et al., 2018) and `gerbil_connect` (Section 4.5.2), and black-box evaluate the systems over the same benchmark dataset CoNLL/AIDA (Hoffart et al., 2011) which allows us to abstract away their internal model structure and decoding algorithms. Next, we discuss the importance of the pre-built candidate sets (Section 4.2) for obtaining good results on benchmarks in entity linking. However, candidate sets are not always available, and the literature lacks a systematic evaluation of the entity linking systems in absence of the candidate sets. To fill this gap, we suggest an experimental setup to replace them with a feasible set; the entire in-domain vocabulary of the benchmark dataset. Please note that our focus in these experiments is not to re-implement each technique, but rather to evaluate the *resilience* of the entity linking systems in absence of the carefully hand-crafted candidate sets. Lastly, we examine the *adaptability* of the entity linking systems in the literature to unseen test data using the novel `AIDA/testc` dataset (Section 4.8) which contains new annotations on news stories in 2020 with 924 novel entities.

6.2.1 Unified Black-Box Evaluation

We benchmark the recent entity linking systems, unchanged and as provided originally by their authors. In these experiments, we intend to examine the suitability of these systems as off-the-shelf systems which can be integrated in future downstream applications.

In the evaluation procedure, GERBIL will provide the testing documents to `gerbil_connect` and receives the entity annotations in the format of (*begin character*, *end character*,

	Micro-F1			Difference	
	testa	testb	testc	testa	testb
Kolitsas et al. (2018)	89.50	82.44	65.75	+0.10	+0.04
Peters et al. (2019)					
KnowBert-Wiki	76.74	71.68	54.12	-3.46	-2.72
KnowBert-W+W	77.19	71.69	53.92	-4.91	-2.01
Poerner et al. (2020)	89.40	84.83	65.93	-1.40	-0.17
van Hulst et al. (2020)					
Wiki 2014	83.30	82.53	71.69	-	-0.77
Wiki 2019	79.64	80.10	73.54	-	-0.40
De Cao et al. (2021b)	90.09	82.78	75.60	-	-0.92
De Cao et al. (2021a)	87.29	85.65	47.54	-	+0.15
Zhang et al. (2022)	86.81	84.30	72.55	-	-1.50
Feng et al. (2022)	87.64	86.49	65.05	-	+0.19
SpEL-large-500K (no cnds.)	89.72	82.25	77.54	+0.02	+0.05
SpEL-large-500K (Kb+Yago)	89.89	82.88	59.50	+0.09	+0.08
SpEL-large-500K (PPRforNED)	91.58	85.22	46.98	+0.08	+0.02

Table 6.3: Comparison of recent entity linking systems within the unified black-box testing framework of GERBIL + `gerbil_connect`. Difference column reports the difference between our unified evaluation environment and the originally reported numbers. We have assessed all models twice for consistency. Except for (De Cao et al., 2021b), all models yielded identical scores, while De Cao et al. (2021b) showed a low variance of 0.08 in the results. Thus, the results mirror those reported by GERBIL, with the exception of (De Cao et al., 2021b), which is averaged over two runs.

entity annotation) from `gerbil_connect`. We implement `gerbil_connect` tailored to each entity linking system so that it can transform the evaluation documents to readable inputs for each system. Specifically, we (1) utilize NLTK's word tokenizer[6] to transform raw non-tokenized evaluation sets into their expected CoNLL tokenized format for the models that depend on reading from AIDA test files (Peters et al., 2019; Poerner et al., 2020; Feng et al., 2022), (2) simulate long text splitting and result merging strategies for the models with input length constraints (Peters et al., 2019; Poerner et al., 2020; Feng et al., 2022; De Cao et al., 2021b), (3) implement a subword token id to character id conversion for the models that output annotations as tokenized subword ids (Peters et al., 2019; Poerner et al., 2020; Feng et al., 2022), and (4) provide the external data sources such as the pre-built candidate sets to the model initializers where necessary (De Cao et al., 2021b,a). Empirically, running the models without adding these techniques significantly hurts performance. Removing the tokenization step alone can drop the model performance by up to 20 Micro-F1 points.

At the end, `gerbil_connect` translates the produced annotations in each system back to the unified annotation format, understandable for GERBIL. We train the models that are not released by the authors (Poerner et al., 2020; Feng et al., 2022), using their own

[6] `https://www.nltk.org/api/nltk.tokenize.html`.

released source code, and do not consider the models which we were not able to acquire their training source code or were not able to get their training scripts to converge (Martins et al., 2019; Févry et al., 2020; Mrini et al., 2022; Kannan Ravi et al., 2021; Broscheit, 2019; Xiao et al., 2023). We use CoNLL/AIDA evaluation sets `testa` and `testb` - reported by all entity linking systems tested in different evaluation frameworks - as well as the newly annotated AIDA/`testc` evaluation set. The results tables show the GERBIL InKB Micro-F1 evaluation results.

Table 6.3 presents the unified black-box evaluation results. The necessary unification adjustments mentioned above and the evaluation format has caused some evaluation scores to deviate from their original reported results. However, we have tried to control for this as much as possible. The `Difference` columns in Table 6.3 reflects on the mentioned score deviations.

In our experiments, we found that (Peters et al., 2019) suffered the most, with an approximate loss of 5% when comparing our results to the originally reported scores. The rest of the models were hit by at most 2%, confirming the reliability of our framework for further analysis. `testc` is a more challenging evaluation set which contains novel entities that typically hurt model recall. Our experimental results confirm that the structured prediction-based SpEL model is the best performing entity linker. Additionally, we observed that following SpEL, generative entity linking models outperformed other non-generative models on `testc`.

6.2.2 Candidate Set Ablations

Candidate sets are an integral part of entity linking systems, many of which assume the presence of good quality sets to perform well. Although this assumption holds when linking to English Wikipedia, it does not necessarily hold when considering other ontologies (e.g. UMLS; Bodenreider, 2004) and languages other than English[7].

We ablate the mention-specific candidate sets from the entity linking systems to study their performance in absence of the hand-crafted candidate sets. For our experiments, we select the candidate-set-independent setting of the models in any system that provides such a setting. For the other systems that require a candidate set, and we cannot remove the candidate set dependence, we return the entire in-domain mention vocabulary of AIDA (the in-domain fixed candidate set in Section 4.5) as the replacement for the required candidate sets (5600 entities including the '`O`' entity). Where applicable, we add priors such that each candidate has an equal probability.

Table 6.4 demonstrates the evaluation results of the models after considering the candida te-independent version of the models, or the candidate set expansion. We also experimented with removing candidate sets altogether, but the models that appear in Table 6.3, and

[7]See Botha et al. (2020) for more discussion.

		Micro-F1		
		testa	testb	testc
a)	De Cao et al. (2021b)	85.15	78.98	75.62
	De Cao et al. (2021a)	62.00	49.51	37.05
	Zhang et al. (2022)	86.81	84.30	72.55
	SpEL-large-500K	89.72	82.25	77.54
b)	Poerner et al. (2020)	22.81	18.81	17.56
	Feng et al. (2022)	35.00	32.58	27.48

Table 6.4: Comparison of entity linking systems after a) running the model with no access to hand-crafted candidate sets b) modifying the model to consider the entire AIDA in-domain vocabulary as the candidate set.

do not appear in Table 6.4 failed without candidate sets. These results demonstrate that most entity linking systems are too intertwined with their candidate sets and without this additional data resource, the systems do not produce useful results and are too brittle to be used in real-world production deployments.

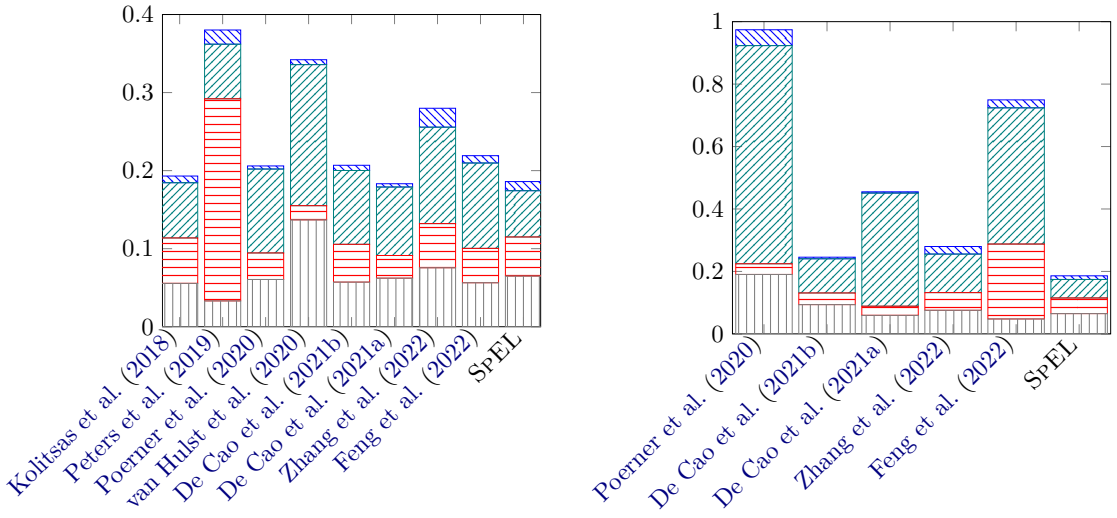
Table 6.4 results prove that generation-based systems are more *resilient* against candidate sets. Nonetheless, without given candidate sets, (De Cao et al., 2021b) and (De Cao et al., 2021a) lose approximately 5% and 20-30% of their best performance, respectively. SpEL - a non-generation-based system, designed without dependence on candidate sets and only using these resources to improve performance - suffers the least and loses only 2% of its best performance without candidate sets.

The largest performance drop in our experiments correlates with using mention-entity similarity methods for entity disambiguation, where a representation of the mention and entity are compared to determine the most relevant entity. In these systems, models that generate mention representations by combining candidate entity representations see their performance decreased to 20%-35%, while models that generate mention representations by combining the word or token representations within or surrounding the mention perform too poorly to be present in Table 6.4. SpEL and (De Cao et al., 2021b) only show an approximate 2% drop in performance, showing that they can easily handle a larger set of candidate entities.

The larger candidate sets lead to longer inference times. The run time for (Feng et al., 2022; Kolitsas et al., 2018; Poerner et al., 2020; Peters et al., 2019) that compare the mentions to each entity in the candidate set increases by 90x, 50x, 25x, and 10x, respectively. (van Hulst et al., 2020) does not follow this trend since it selects the 30 candidate entities with the highest prior before performing entity disambiguation.

Error Analysis. We store the produced annotations from each system reported in Table 6.3 (w/ candidate sets) and Table 6.4 (w/o candidate sets), and compare their produced annotations with the expected annotations of AIDA/`testa` (4791 annotations). For models with multiple reported settings, we select the setting correlated to best performance on

Figure 6.2: Entity linking error distribution in four categories of over-generated (gray, vertical), under-generated (red, horizontal), incorrect entity (teal, north east) and incorrect mention (blue, north west) before candidate set ablations (left) and after the ablations (right). The y-axis is the error analysis ratio as described below.



AIDA/`testc` as it represents the most generalization-capable setting for unseen in-domain documents.

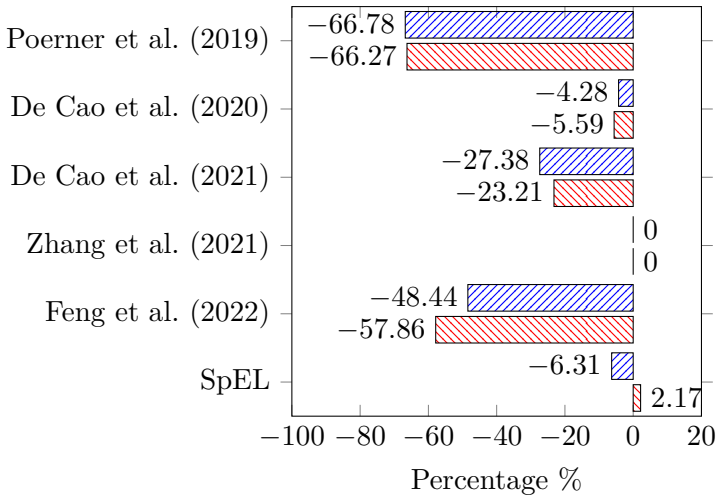
We count the number of annotations in four error categories of over-generated, under-generated, incorrect mention and incorrect entity, and divide each by the total number of gold annotations. Figure 6.2 presents the calculated error analysis ratios. Over-generation refers to annotations predicted by the model and not in the gold set. Under-generation refers to annotations in the gold set but not predicted by the model. Incorrect entity refers to annotations where the model linked to the wrong entity. Incorrect mention refers to annotations where the span's start or end is incorrect.

Before ablation of candidate sets (Figure 6.2-left), (van Hulst et al., 2020) has the highest rate of over-generation followed by (Zhang et al., 2022), while (Peters et al., 2019) shows the lowest over-generation rate. On the other hand, (Peters et al., 2019) has the highest under-generation ratio establishing itself as a conservative entity linking system.

Comparing the performance of entity systems with and without candidate sets, the biggest increase is seen in incorrect entity prediction ratios, confirming the dependence of entity linking systems to candidate sets. (Poerner et al., 2020) sees the biggest increase in incorrect entity predictions followed by (Feng et al., 2022). While (Zhang et al., 2022) and SpEL report the smallest rate increase in this category as these methods are less dependent on candidate sets. (Feng et al., 2022) on the other hand shows an increase in under-generation signaling the effect of candidate sets in prediction confidence for this system.

Figure 6.3 illustrates the disparities in precision and recall pre- and post-ablation of candidate sets for the models outlined in Table 6.4. Our findings reveal that candidate sets

Figure 6.3: Entity linking micro precision (blue, north east) and recall (red, north west) score differences over `testa` between model's original configuration and candidate set ablation configuration.



significantly enhance precision and recall. With the exception of (Zhang et al., 2022), which generates candidates in real-time, the other systems show that without candidate sets there is a substantial decrease in precision and recall, exceeding 60% for (Poerner et al., 2020).

6.3 Additional Co-authored Contributions in Machine Translation

In addition to the five publications discussed in earlier chapters and sections of the dissertation, this section discusses three other co-authored peer-reviewed publications in the course of the PhD program. The following outlines each publication and reviews their respective contributions.

- **Translation-based Supervision for Policy Generation in Simultaneous Neural Machine Translation** (Alinejad, Shavarani, and Sarkar, 2021) where we introduced a novel supervised learning approach for optimizing simultaneous machine translation, aiming to reduce average lagging in target token production while maintaining translation quality. By comparing the translations in simultaneous setting with full-sentence translations during training to generate oracle action sequences, the proposed method offered a more trainable alternative to heuristic methods, leading to improved translation quality and reduced average lagging in simultaneous translation.

- **Top-down Tree Structured Decoding with Syntactic Connections for Neural Machine Translation and Parsing** (Gū, Shavarani, and Sarkar, 2018) where we presented *Seq2DRNN*, a novel translation model with syntax-aware decoding through a top-down tree-structured approach. This model enhanced translation quality by leveraging constituency parsing instead of dependency parsing, and demonstrated improved fluency and reordering in translations compared to sequential and other con-

temporary syntax-based translation models, while also exhibiting competitive parsing accuracy simultaneously.

- **Simultaneous Translation using Optimized Segmentation** (Siahbani, Shavarani, Alinejad, and Sarkar, 2018) where we introduced an automatic simultaneous translation framework by integrating a segmentation model with an incremental decoding algorithm, resulting in improved translation accuracy and reduced delay. Leveraging annotated data for sentence segmentation proposed by Oda et al. (2014), our approach achieved high translation quality close to offline systems while minimizing production delay, surpassing other simultaneous translation systems in both translation quality and latency.

Part III

Summary

Chapter 7

Conclusion and Future Directions

In this dissertation, we presented three contributions focused on using structured prediction to achieve computationally efficient and highly accurate NLP solutions. While targeting diverse complex tasks, our primary aim was to demonstrate the potential for performance enhancement and accuracy improvement, with emphasis on simplification. We highlighted the utilization of pre-trained models and the development of reusable models when task-specific options are lacking. Our approach offers a pathway to simplification but is not comprehensive; rather, it is a direction for the community to integrate simplification and efficiency considerations into their designs.

We summarize this dissertation as the following:

- In our first contribution, we demonstrated the relevance of structured prediction in extracting useful linguistic knowledge from BERT and integrating them into neural machine translation framework. We showed that the extracted information provide the translation models with out-of-domain knowledge which not only improves the translation quality but also helps the model to better deal with out-of-vocabulary words. While fine-tuning was common during the project's execution, a key insight from this contribution was that in cases, simpler information extraction techniques may yield superior results, surpassing fine-tuning all model parameters.

- In our next contribution, we introduced several improvements to a structured prediction approach for entity linking leading to SpEL, our proposed entity linking framework. Our experiments on the AIDA dataset demonstrated that SpEL yield state-of-the-art performance, as evidenced by head-to-head comparisons with competing methods using a commonly used evaluation toolkit. A key insight from our approach is the feasibility of designing models with superior performance in entity linking while prioritizing computational efficiency and reducing model parameter count to enhance throughput.

- In our last contribution, we proposed *Entity Retrieval*, an application of our structured prediction-based entity linking framework aimed at enhancing retrieval-augmented question answering systems. Our results demonstrate that *Entity Retrieval* offers a

promising alternative to dense retrieval for augmenting entity-centric questions in prompting LLMs. A key insight from this contribution is that a simplified and efficient design not only benefits the primary task but also extends advantages to downstream tasks by improving both speed and accuracy.

Lastly, we discuss potential future directions for the work presented in this dissertation:

- Structured prediction holds significant untapped potential in the domain of LLMs. As highlighted by Liu et al. (2023), LLMs may struggle with ambiguity, prioritizing immediate token prediction tasks over contextual nuances. While acknowledging the remarkable language understanding abilities of LLMs, research suggests that their effectiveness can be substantially enhanced with additional support in navigating ambiguity, as exemplified by our proposed *Entity Retrieval* method. Conceptually akin to operating systems, LLMs require complementary software to fully leverage their capabilities. Through structured prediction, cost-effective and efficient software solutions can be devised to empower these potent *operating systems*, enabling them to integrate effortlessly with various language processing models such as named entity recognition, part-of-speech tagging, entity linking and alike, thereby enhancing language comprehension in response generation.

- Structured prediction offers the potential to cultivate cost-effective yet robust models tailored to specialized domains such as healthcare. For instance, entity-centric question answering, as explored in Chapter 5, has played a crucial role in disseminating public knowledge during the COVID-19 pandemic, where human resources were limited (Kumar et al., 2023; Indriati et al., 2024). A promising avenue for future research within this dissertation could involve adapting the proposed methodologies to the field of medical NLP (e.g. Sezgin et al., 2023).

- Structured prediction-based modeling offers particular benefits in low-resource settings and languages lacking ample training data. Such an approach holds promise for future exploration, as demonstrated by our linguistic information augmentation method outlined in Chapter 3.

Bibliography

Steven Abney, Michael Collins, and Amit Singhal. Answer extraction. In *Sixth Applied Natural Language Processing Conference*, pages 296–301, Seattle, Washington, USA, April 2000. Association for Computational Linguistics. doi: 10.3115/974147.974188. URL https://aclanthology.org/A00-1041.

Emmanuel Adebisi, Bolanle Adefowoke Ojokoh, and Folasade Olubusola Isinkaye. An open domain factoid qa framework with improved validation techniques. *International Journal of Information Science and Management (IJISM)*, 20(1), 2022. URL https://ijism.isc.ac/article_698358_5bbe2ff065b2c5c80582fa168e9cc58c.pdf.

Ahmad Aghaebrahimian and Filip Jurčíček. Open-domain factoid question answering via knowledge graph search. In Mohit Iyyer, He He, Jordan Boyd-Graber, and Hal Daumé III, editors, *Proceedings of the Workshop on Human-Computer Question Answering*, pages 22–28, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-0104. URL https://aclanthology.org/W16-0104.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298.

Wazir Ali, Rajesh Kumar, Yong Dai, Jay Kumar, and Saifullah Tumrani. Neural joint model for part-of-speech tagging and entity extraction. In *Proceedings of the 2021 13th International Conference on Machine Learning and Computing*, pages 239–245, 2021. URL https://drive.google.com/file/d/1zjY4B7iMEKmmEzOknVQPpTZ2JGXCl-vs.

Ashkan Alinejad, Hassan S. Shavarani, and Anoop Sarkar. Translation-based supervision for policy generation in simultaneous neural machine translation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1734–1744, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.130. URL https://aclanthology.org/2021.emnlp-main.130.

Keivan Alizadeh, Iman Mirzadeh, Dmitry Belenko, Karen Khatamifard, Minsik Cho, Carlo C Del Mundo, Mohammad Rastegari, and Mehrdad Farajtabar. Llm in a flash: Efficient large language model inference with limited memory. *arXiv preprint arXiv:2312.11514*, 2023. URL https://arxiv.org/pdf/2312.11514.pdf.

Tariq Alqahtani, Hisham A Badreldin, Mohammed Alrashed, Abdulrahman I Alshaya, Sahar S Alghamdi, Khalid bin Saleh, Shuroug A Alowais, Omar A Alshaya, Ishrat Rahman, Majed S Al Yami, et al. The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research. *Research in Social and Administrative Pharmacy*, 2023. URL https://www.sciencedirect.com/science/article/pii/S1551741123002802.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer, 2007. URL https://link.springer.com/chapter/10.1007/978-3-540-76298-0_52.

Damaris Ayuso, Sean Boisen, Heidi Fox, Herb Gish, Robert Ingria, and Ralph Weischedel. BBN: Description of the PLUM system as used for MUC-4. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*, 1992. URL https://aclanthology.org/M92-1024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. URL https://arxiv.org/pdf/2204.05862.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf.

Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004. URL https://pubmed.ncbi.nlm.nih.gov/14681409/.

Jan A. Botha, Zifei Shan, and Daniel Gillick. Entity Linking in 100 Languages. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.630. URL https://aclanthology.org/2020.emnlp-main.630.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993. URL https://proceedings.neurips.cc/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf.

Samuel Broscheit. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1063. URL https://aclanthology.org/K19-1063.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

David Carter. *Interpreting anaphors in natural language texts*. Halsted Press, 1987. URL https://aclanthology.org/J90-1006.pdf.

Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, may 2012. URL https://cris.fbk.eu/retrieve/handle/11582/104409/4358/WIT3-EAMT2012.pdf.

Joseph Chang, Richard Tzong-Han Tsai, and Jason S Chang. Wikisense: Supersense tagging of wikipedia named entities based wordnet. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, volume 1, 2009.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171.

Young Min Cho, Li Zhang, and Chris Callison-Burch. Unsupervised entity linking with guided summarization and multiple-choice selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9394–9401, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.638. URL https://aclanthology.org/2022.emnlp-main.638.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. QuAC: Question answering in context. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1241. URL https://aclanthology.org/D18-1241.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB.

Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. On the use of BERT for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 108–117, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5611. URL https://www.aclweb.org/anthology/D19-5611.

Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118694. URL https://www.aclweb.org/anthology/W02-1001.

Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39 (1):80–91, 1996. URL https://dl.acm.org/doi/pdf/10.1145/234173.234209.

Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/D07-1074.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. *arXiv preprint arXiv:2401.14887*, 2024. URL https://arxiv.org/pdf/2401.14887.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. Kbqa: Learning question answering over qa corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5), 2017. URL https://www.vldb.org/pvldb/vol10/p565-cui.pdf.

Wisam Dakka and Silviu Cucerzan. Augmenting wikipedia with named entity tags. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Highly parallel autoregressive entity linking with discriminative correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7662–7669, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.604. URL https://aclanthology.org/2021.emnlp-main.604.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 01 2021b. URL https://openreview.net/forum?id=5k8F6UU39V.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke Van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49, 2015. URL https://giusepperizzo.github.io/publications/Derczynski_Manyard_Rizzo-IPM2014.pdf.

Chauhan Dev, Naman Biyani, Nirmal P Suthar, Prashant Kumar, and Priyanshu Agarwal. Structured prediction in nlp–a survey. *arXiv preprint arXiv:2110.02057*, 2021. URL https://arxiv.org/pdf/2110.02057v1.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024. URL https://arxiv.org/pdf/2401.08281.pdf.

Xinya Du, Alexander Rush, and Claire Cardie. Template filling with generative transformers. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.70. URL https://aclanthology.org/2021.naacl-main.70.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1033. URL https://aclanthology.org/P15-1033.

Zheng Fang, Yanan Cao, Ren Li, Zhenyu Zhang, Yanbing Liu, and Shi Wang. High quality candidate generation and sequential graph attention network for entity linking. In *Proceedings of The Web Conference 2020*, pages 640–650, 2020.

Yukun Feng, Amir Fayazi, Abhinav Rastogi, and Manabu Okumura. Efficient entity embedding construction from type knowledge for BERT. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 1–10, Online only, November 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-aacl.1.

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. Empirical evaluation of pretraining strategies for supervised entity linking. In *Automated Knowledge Base Construction*, 2020. doi: 10.24432/C59G6S. URL https://openreview.net/forum?id=iHXV8UGYyL.

Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1277. URL https://aclanthology.org/D17-1277.

Mercedes Garcia-Martinez, Loic Barrault, and Fethi Bougares. Factored neural machine translation architectures. In *HAL archives ouvertes*, 2016. URL https://hal.archives-ouvertes.fr/hal-01433161/document.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1049. URL https://aclanthology.org/K19-1049.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2G: Retrieve, rerank, generate. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.194. URL https://aclanthology.org/2022.naacl-main.194.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. URL http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf.

Yoav Goldberg. Assessing bert's syntactic abilities. *Computation and Language Research Repository*, arXiv:1901.05287, 2019. URL http://arxiv.org/abs/1901.05287. version 1.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052. IEEE, 2005. URL https://mediatum.ub.tum.de/doc/1290193/document.pdf.

Ralph Grishman. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692, 2019. doi: 10.1017/S1351324919000512. URL https://doi.org/10.1017/S1351324919000512.

Ralph Grishman, David Westbrook, and Adam Meyers. Nyu's english ace 2005 system description. *Ace*, 5(2), 2005.

Jetic Gū, Hassan S. Shavarani, and Anoop Sarkar. Top-down tree structured decoding with syntactic connections for neural machine translation and parsing. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 401–413, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1037. URL https://aclanthology.org/D18-1037.

Rujun Han, Qiang Ning, and Nanyun Peng. Joint event and temporal relation extraction with shared representations and structured prediction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 434–444, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1041. URL https://aclanthology.org/D19-1041.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Benjamin Heinzerling and Kentaro Inui. 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*, pages 1772–1791, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.153. URL https://aclanthology.org/2021.eacl-main.153.

Ryuichiro Higashinaka, Kugatsu Sadamitsu, Kuniko Saito, Toshiro Makino, and Yoshihiro Matsuo. Creating an extended named entity dictionary from Wikipedia. In *Proceedings of COLING 2012*, pages 1163–1178, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-1071.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/HintonDengYuEtAl-SPM2012.pdf.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL https://www.cs.toronto.edu/~hinton/absps/distillation.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1072.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554, 2012. URL https://dl.acm.org/doi/abs/10.1145/2396761.2396832.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/pdf/2106.09685.

Shijie Hu, Xiaoyu Li, Jiayu Bai, Hang Lei, Weizhong Qian, Sunqiang Hu, Cong Zhang, Akpatsa Samuel Kofi, Qian Qiu, Yong Zhou, et al. Neural machine translation by fusing key information of text. *CMC Comput. Mater. Contin*, 74:2803–2815, 2023. URL https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC-74-2/TSP_CMC_32732/TSP_CMC_32732.pdf.

Guanhua Huang, Runxin Xu, Ying Zeng, Jiaze Chen, Zhouwang Yang, and Weinan E. An iteratively parallel generation method with the pre-filling strategy for document-level event extraction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10834–10852, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.668. URL https://aclanthology.org/2023.emnlp-main.668.

Kenji Imamura and Eiichiro Sumita. Recycling a pre-trained BERT encoder for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 23–31, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5603. URL https://www.aclweb.org/anthology/D19-5603.

Indriati Indriati, Randy Cahya Wihandika, Putra Pandu Adikara, Barlian Henryranu Prasetio, and Lailil Muflikhah. Question answering system for factoid questions about COVID-19 with natural language processing approach. *AIP Conference Proceedings*, 3026(1):050012, 03 2024. ISSN 0094-243X. doi: 10.1063/5.0199740. URL https://doi.org/10.1063/5.0199740.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online, April 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74. URL https://aclanthology.org/2021.eacl-main.74.

Gautier Izacard and Edouard Grave. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=NTEz-6wysdb.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jKN1pXi7b0.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. URL https://faculty.cc.gatech.edu/~zha/CS8803WST/dcg.pdf.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://www.aclweb.org/anthology/P19-1356.

Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. In Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://aclanthology.org/P08-1030.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/pdf/2401.04088.pdf.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. URL https://arxiv.org/pdf/1702.08734.pdf.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.307. URL https://aclanthology.org/2023.acl-long.307.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR, 2023. URL https://proceedings.mlr.press/v202/kandpal23a/kandpal23a.pdf.

Yue Kang, Zhao Cai, Chee-Wee Tan, Qian Huang, and Hefu Liu. Natural language processing (nlp) in management research: A literature review. *Journal of Management Analytics*, 7(2):139–172, 2020. URL https://www.tandfonline.com/doi/abs/10.1080/23270012.2020.1756939.

Manoj Prabhakar Kannan Ravi, Kuldeep Singh, Isaiah Onando Mulang', Saeedeh Shekarpour, Johannes Hoffart, and Jens Lehmann. CHOLAN: A modular approach for neural entity linking on Wikipedia and Wikidata. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 504–514, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.40.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020a. URL https://arxiv.org/pdf/2001.08361.pdf.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020b.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Diederik P Kingma and Lei Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language*

Learning, pages 519–529, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1050. URL https://aclanthology.org/K18-1050.

Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.jp*, 2006.

Amar Kumar, Rupal Bhargava, and Manoj Jayabalan. Covid qa network: A specific case of biomedical question answering. In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, pages 333–338. IEEE, 2023. URL https://ieeexplore.ieee.org/document/10099510.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1132. URL https://www.aclweb.org/anthology/P18-1132.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL https://aclanthology.org/N16-1030.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=H1eA7AEtvS.

Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020b. URL https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Lin Li, Mengjing Zhang, Zhaohui Chao, and Jianwen Xiang. Using context information to enhance simple question answering. *World Wide Web*, 24:249–277, 2021. URL https://arxiv.org/pdf/1905.01995.pdf.

Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In Hinrich Schuetze, Pascale Fung, and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1008.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. Few-shot in-context learning on knowledge base question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6980, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.385. URL https://aclanthology.org/2023.acl-long.385.

Chin-Yew Lin and Franz Josef Och. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507, Geneva, Switzerland, August 2004. COLING. URL https://www.aclweb.org/anthology/C04-1072.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362, 2021. URL https://dl.acm.org/doi/10.1145/3404835.3463238.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016. doi: 10.1162/tacl_a_00115. URL https://www.aclweb.org/anthology/Q16-1037.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. We're afraid language models aren't modeling ambiguity. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.51. URL https://aclanthology.org/2023.emnlp-main.51.

Shuheng Liu and Alan Ritter. Do CoNLL-2003 named entity taggers still work well in 2023? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8254–8271, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.459. URL https://aclanthology.org/2023.acl-long.459.

Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. Autoregressive structured prediction with language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational*

Linguistics: EMNLP 2022, pages 993–1005, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.70. URL https://aclanthology.org/2022.findings-emnlp.70.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. URL https://arxiv.org/pdf/1907.11692.pdf.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. Pretrained transformers for simple question answering over knowledge graphs. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*, pages 470–486. Springer, 2019. URL https://arxiv.org/pdf/2001.11985.pdf.

Youmi Ma, An Wang, and Naoaki Okazaki. DREEAM: Guiding attention with evidence for improving document-level relation extraction. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1971–1983, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.145. URL https://aclanthology.org/2023.eacl-main.145.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-2026. URL https://aclanthology.org/P19-2026.

Malak Mashaabi, Areej Alotaibi, Hala Qudaih, Raghad Alnashwan, and Hend Al-Khalifa. Natural language processing in customer service: A systematic review. *arXiv preprint arXiv:2212.09523*, 2022. URL https://arxiv.org/pdf/2212.09523.pdf.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/20c86a628232a67e7bd46f76fba7ce12-Paper.pdf.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024, 2019. URL https://papers.nips.cc/paper/9551-are-sixteen-heads-really-better-than-one.pdf.

Rada Mihalcea and Andras Csomai. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, 2007.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010. URL https://www.isca-speech.org/archive_v0/archive_papers/interspeech_2010/i10_1045.pdf.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. URL https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Scott Miller, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. Algorithms that learn to extract information BBN: TIPSTER phase III. In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998*, pages 75–89, Baltimore, Maryland, USA, October 1998. Association for Computational Linguistics. doi: 10.3115/1119089.1119107. URL https://aclanthology.org/X98-1014.

Makoto Miwa and Mohit Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1105. URL https://aclanthology.org/P16-1105.

Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2047. URL https://aclanthology.org/N18-2047.

Khalil Mrini, Shaoliang Nie, Jiatao Gu, Sinong Wang, Maziar Sanjabi, and Hamed Firooz. Detection, disambiguation, re-ranking: Autoregressive entity linking as a multi-task problem. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1972–1983, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.156. URL https://aclanthology.org/2022.findings-acl.156.

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

Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre, editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1142.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.669. URL https://aclanthology.org/2022.emnlp-main.669.

Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. Open knowledge extraction challenge. In *Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31-June 4, 2015, Revised Selected Papers*, pages 3–15. Springer, 2015. URL https://dariogarigliotti.github.io/assets/pdf/pubs/2015-ESWC-OKE2015_challenge.pdf.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Optimizing segmentation strategies for simultaneous speech translation. In Kristina Toutanova and Hua Wu, editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 551–556, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2090. URL https://aclanthology.org/P14-2090.

Nicolas Ong, Hassan Shavarani, and Anoop Sarkar. Unified examination of entity linking in absence of candidate sets. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 113–123, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-short.11.

OpenAI. Gpt-4 technical report, 2023. URL https://arxiv.org/pdf/2303.08774.pdf.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6301. URL https://www.aclweb.org/anthology/W18-6301.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021. URL https://arxiv.org/pdf/2104.10350.pdf.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. Yago 4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*, pages 583–596. Springer, 2020.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023. URL https://arxiv.org/pdf/2302.12813.pdf.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Maria Pershina, Yifan He, and Ralph Grishman. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243, Denver, Colorado, May 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1026. URL https://aclanthology.org/N15-1026.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 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)*, pages 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1005. URL https://aclanthology.org/D19-1005.

Francesco Piccinno and Paolo Ferragina. From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62, 2014. URL https://dl.acm.org/doi/pdf/10.1145/2633211.2634350.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.71. URL https://aclanthology.org/2020.findings-emnlp.71.

Massimo Poesio, Juntao Yu, Silviu Paun, Abdulrahman Aloraini, Pengcheng Lu, Janosch Haber, and Derya Cokal. Computational models of anaphora. *Annual Review of Linguistics*, 9:561–587, 2023. URL https://doi.org/10.1146/annurev-linguistics-031120-111653.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain, April 2017.

Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-2025.

L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi: 10.1109/5.18626. URL https://ieeexplore.ieee.org/document/18626.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. In , 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8): 9, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023. doi: 10.1162/tacl_a_00605. URL https://aclanthology.org/2023.tacl-1.75.

Prakash Ranjan and Rakesh Chandra Balabantaray. Question answering system for factoid based question. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 221–224. IEEE, 2016. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7917964.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1138.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. URL https://www.nowpublishers.com/article/Details/INR-019.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In *Text Retrieval Conference*, 1994. URL https://api.semanticscholar.org/CorpusID:3946054.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N^3-a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *LREC*, pages 3529–3533, 2014.

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625, 2018. URL http://www.semantic-web-journal.net/system/files/swj1671.pdf.

Alexander Rush. The annotated transformer. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 52–60, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2509. URL https://www.aclweb.org/anthology/W18-2509.

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. Large language models meet knowledge graphs to answer factoid questions. In Chu-Ren Huang, Yasunari Harada, Jong-Bok Kim, Si Chen, Yu-Yin Hsu, Emmanuele Chersoni, Pranav A, Winnie Huiheng Zeng, Bo Peng, Yuxi Li, and Junlin Li, editors, *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 635–644, Hong Kong, China, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.paclic-1.63.

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

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.272. URL https://aclanthology.org/2022.naacl-main.272.

Alexander Schlaubitz. Natural language processing in finance: analysis of sentiment and complexity of news and earnings reports of swiss smes and their relevance for stock returns. , 2021.

Robin Schmidt, Telmo Pires, Stephan Peitz, and Jonas Lööf. Non-autoregressive neural machine translation: A call for clarity. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2785–2799, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.179. URL https://aclanthology.org/2022.emnlp-main.179.

Peter Schönhofen. Identifying document topics using the wikipedia category network. *Web Intelligence and Agent Systems: An International Journal*, 7(2):195–207, 2009.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012. URL https://storage.googleapis.com/pub-tools-public-publication-data/pdf/37842.pdf.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. Simple entity-centric questions challenge dense retrievers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.496. URL https://aclanthology.org/2021.emnlp-main.496.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended named entity hierarchy. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May 2002. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2002/pdf/120.pdf.

Satoshi Sekine, Maya Ando, Akio Kobayashi, Koji Matsuda, Masatoshi Suzuki, Duc Nguyen, and Kentaro Inui. Wikipedia categorization data based on extended named entity (in japanese). *The 24th Annual conference of Association for Natural Language Processing, Japan*, pages 504–507, 2018a.

Satoshi Sekine, Akio Kobayashi, and Kouta Nakayama. Shinra: Structuring wikipedia by collaborative contribution. *Automated Knowledge Base Construction*, 2018b.

Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2209. URL https://www.aclweb.org/anthology/W16-2209.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Emre Sezgin, Syed-Amad Hussain, Steve Rust, and Yungui Huang. Extracting medical information from free-text and unstructured patient-generated health data using natural language processing methods: feasibility study with real-world data. *JMIR Formative Research*, 7:e43014, 2023. URL https://formative.jmir.org/2023/1/e43014.

Shariq Shah, Hossein Ghomeshi, Edlira Vakaj, Emmett Cooper, and Shereen Fouad. A review of natural language processing in contact centre automation. *Pattern Analysis and Applications*, 26(3):823–846, 2023. URL https://link.springer.com/article/10.1007/s10044-023-01182-8.

Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951. URL https://languagelog.ldc.upenn.edu/myl/Shannon1950.pdf.

Hassan S. Shavarani and Anoop Sarkar. Better neural machine translation by extracting linguistic information from BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2772–2783, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.241.

Hassan S. Shavarani and Anoop Sarkar. SpEL: Structured prediction for entity linking. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11123–11137, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.686. URL https://aclanthology.org/2023.emnlp-main.686.

Hassan S. Shavarani and Anoop Sarkar. Entity retrieval for answering entity-centric questions, 2024.

Hassan S. Shavarani and Satoshi Sekine. Multi-class multilingual classification of Wikipedia articles using extended named entity tag set. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1197–1201, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.150.

Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27 (2):443–460, 2014.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023. URL https://arxiv.org/pdf/2301.12652.pdf.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.320. URL https://aclanthology.org/2021.findings-emnlp.320.

Maryam Siahbani, Hassan S. Shavarani, Ashkan Alinejad, and Anoop Sarkar. Simultaneous translation using optimized segmentation. In Colin Cherry and Graham Neubig, editors, *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 154–167, Boston, MA, March 2018. Association for Machine Translation in the Americas. URL https://aclanthology.org/W18-1815.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34:25968–25981, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/da3fde159d754a2555eaa198d2d105b2-Paper.pdf.

George Smith et al. A brief introduction to the tiger treebank, version 1. *Potsdam Universität*, 2003. URL https://www.ims.uni-stuttgart.de/documents/ressourcen/korpora/tiger-corpus/annotation/tiger_introduction.pdf.

Noah A Smith, Michael Heilman, and Rebecca Hwa. Question generation as a competitive undergraduate course project. In *Proceedings of the NSF Workshop on the Question Generation Shared Task and Evaluation Challenge*, volume 9, 2008. URL https://www.cs.cmu.edu/~nasmith/papers/smith+heilman+hwa.nsf08.pdf.

Mohammad S Sorower. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 18, 2010.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *16th International Conference on the World Wide Web*, pages 697–706, 2007.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. Syntax-infused transformer and bert models for machine translation and natural language understanding. *Computation and Language Research Repository*, arXiv:1911.06156, 2019. URL http://arxiv.org/abs/1911.06156. version 1.

Beth M. Sundheim. Overview of the third Message Understanding Evaluation and Conference. In *Third Message Understanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*, 1991. URL https://aclanthology.org/M91-1001.

Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. Neural joint learning for classifying wikipedia articles into fine-grained named entity types. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation*, pages 535–544, Seoul, South Korea, October 2016. URL http://www.aclweb.org/anthology/Y16-3027.

Masaya Suzuki, Kanako Komiya, Minoru Sasaki, and Hiroyuki Shinnou. Fine-tuning for named entity recognition using part-of-speech tagging. In Stephen Politzer-Ahles, Yu-Yin Hsu, Chu-Ren Huang, and Yao Yao, editors, *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, 1–3 December 2018. Association for Computational Linguistics. URL https://aclanthology.org/Y18-1072.

Chuanyuan Tan, Yuehe Chen, Wenbiao Shao, and Wenliang Chen. Make a choice! knowledge base question answering with in-context learning. *arXiv preprint arXiv:2305.13972*, 2023. URL https://arxiv.org/pdf/2305.13972.pdf.

Sam Tardif, James R Curran, and Tara Murphy. Improved text categorisation for wikipedia named entities. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 104–108, 2009.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=SJzSgnRcKX.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015. URL https://ieeexplore.ieee.org/document/7133169.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://aclanthology.org/W03-0419.

Antonio Toral and Rafael Munoz. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources*, 2006.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a. URL https://arxiv.org/pdf/2302.13971.pdf.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b. URL https://arxiv.org/pdf/2307.09288.pdf.

Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3397271.3401416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Ellen M Voorhees and Donna Harman. Overview of the eighth text retrieval conference (trec-8). In *Proceedings of the Eighth Text REtrieval Conference (TREC-8), NIST Special Publication*, 1999.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. DeepStruct: Pretraining of language models for structure prediction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for*

Computational Linguistics: ACL 2022, pages 803–823, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.67. URL https://aclanthology.org/2022.findings-acl.67.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*, 2023. URL https://arxiv.org/pdf/2304.10428.

Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics, 2012.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Automated concatenation of embeddings for structured prediction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 2643–2660, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.206. URL https://aclanthology.org/2021.acl-long.206.

Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. A graph-based approach to named entity categorization in wikipedia using conditional random fields. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*, pages 5225–5234, 2018.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. Acquiring knowledge from pre-trained model to neural machine translation. *AAAI*, 2019. URL https://aaai.org/Papers/AAAI/2020GB/AAAI-WengR.7823.pdf.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.519. URL https://aclanthology.org/2020.emnlp-main.519.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *Computation and Language Research Repository*, arXiv:1609.08144, 2016. URL https://arxiv.org/abs/1609.08144. version 2.

Chunyang Xiao, Marc Dymetman, and Claire Gardent. Sequence-based structured prediction for semantic parsing. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1341–1350, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1127. URL https://aclanthology.org/P16-1127.

Zilin Xiao, Ming Gong, Jie Wu, Xingyao Zhang, Linjun Shou, and Daxin Jiang. Instructed language models with retrievers are powerful entity linkers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2267–2282, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.139. URL https://aclanthology.org/2023.emnlp-main.139.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1025. URL https://aclanthology.org/K16-1025.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.523. URL https://aclanthology.org/2020.emnlp-main.523.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. Efficient passage retrieval with hashing for open-domain question answering. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.123. URL https://aclanthology.org/2021.acl-short.123.

Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. Global entity disambiguation with BERT. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3264–3271, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.238. URL https://aclanthology.org/2022.naacl-main.238.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. Towards making the most of bert in neural machine translation. *AAAI*, 2020. URL https://aaai.org/Papers/AAAI/2020GB/AAAI-YangJ.7695.pdf.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*, 2023. URL https://arxiv.org/pdf/2305.14002.pdf.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106, 2003. URL https://www.jmlr.org/papers/volume3/zelenko03a/zelenko03a.pdf.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Learning to retrieve: How to train a dense retrieval model effectively and efficiently. *arXiv preprint arXiv:2010.10469*, 2020a. URL https://arxiv.org/pdf/2010.10469.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*, 2020b. URL https://arxiv.org/pdf/2006.15498.

Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. A survey for efficient open domain question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14447–14465, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.808. URL https://aclanthology.org/2023.acl-long.808.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. Entqa: Entity linking as question answering. *International Conference on Learning Representations*, 2022. URL https://openreview.net/pdf?id=US2rTP5nm_.

Yuan Zhang and David Weiss. Stack-propagation: Improved representation learning for syntax. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1566, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1147. URL https://www.aclweb.org/anthology/P16-1147.

Zhebin Zhang, Sai Wu, Dawei Jiang, and Gang Chen. Bert-jam: Maximizing the utilization of bert for neural machine translation. *Neurocomputing*, 460:84–94, 2021. URL https://www.sciencedirect.com/science/article/pii/S0925231221010365.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://aclanthology.org/P19-1139.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. URL https://arxiv.org/pdf/2303.18223.pdf.

Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. A neural probabilistic structured-prediction model for transition-based dependency parsing. In Chengqing Zong and

Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1213–1222, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1117. URL https://aclanthology.org/P15-1117.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*, 2021. URL https://arxiv.org/pdf/2101.00774.pdf.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation. *ICLR 2020*, 2020. URL https://openreview.net/pdf?id=Hyl7ygStwB.