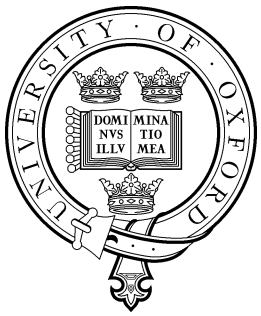
# Impact of Pre-training on Background Knowledge and Societal Bias




Vid Kocijan
St Hugh's College
University of Oxford

A thesis submitted for the degree of
*DPhil of Computer Science*
Michaelmas 2021

Dediju, ki bi ga gotovo
zanimalo moje delo.

To my grandfather
who would surely have loved to
hear about my work.

# Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisors, Thomas Lukasiewicz and Oana-Maria Camburu, who guided me through the project and encouraged my ideas, letting me explore them on my own.

Moreover, I would like to thank everybody who helped me with my research work, either by co-authoring, reviewing, or just discussing my work. I would particularly like to thank (in alphabetical order) Ralph Abboud, Phil Blunsom, İsmail İlkan Ceylan, Ana-Maria Creţu, Eleonora Giunchiglia, and Yordan Yordanov. I would also like to thank Samuel Bowman for accepting me into his research lab for a summer, even if my visit was made virtual by the pandemic. Special thanks go to my examiners, Kyunghyun Cho and Emanuel Salinger, whose helpful suggestions helped me improve the thesis.

My DPhil work was also made possible by staff at St Hugh's College and the Department of Computer Science. I would especially like to thank Julie Sheppard and Sarah Retz-Jones for making administrative work at the department easy with their incredible patience and help.

Finally, I would like to thank everyone that supported me outside of my research work, making my DPhil easier. Ana Štuhec for her never-ending support through my countless failures and successes. My family for always being happy to welcome me at home. Evelina and Marcus Ferrar for helping me out of many seemingly trivial situations that can be hard for a student to resolve, be with transport, storage, food, or a lawn mower. Everyone from Oxford University Mountaineering Club and Taruithorn for countless adventures, some in the mountains, some

in the world of fantasy. Amy, Anna, Odhran, and Robin for making me feel like we can achieve anything we set our minds on.

This work has been supported by the EPSRC Studentship OUCS/ EPSRC-NPIF/VK/1123106. The experiments were conducted on the UK ESPRC funded Tier 2 facility JADE (EP/P020275/1), GPU computing support by Scan Computers International Ltd, and platform provided by Run:AI Labs Ltd. Without these resources, my work could not be completed.

# Abstract

With appropriate pre-training on unstructured text, larger and more accurate neural network models can be trained. Unfortunately, unstructured pre-training data may contain undesired societal biases, which a model may mimic and amplify. This thesis focuses on both improving unsupervised pre-training and developing diagnostics of obtained pre-trained models for potential undesired behaviour.

Pre-training and diagnostics are done on two tasks: coreference resolution and knowledge base completion. For both of them, a novel task-specific method for unsupervised pre-training is introduced. Then, the obtained models are analysed for potential undesired behaviour by evaluating them on relevant datasets, focusing on gender bias in particular.

Two novel pre-training datasets for coreference resolution are introduced, MaskedWiki and WikiCREM. By fine-tuning on these datasets, state-of-the-art performance on multiple benchmarks is achieved, including on the Winograd Schema Challenge, a commonsense reasoning benchmark that requires a lot of background knowledge. The obtained pre-trained models are then evaluated on the Gap benchmark. On this benchmark, potentially problematic patterns in the test set are demonstrated. To remove these undesired patterns, a novel test sample weighting method and a proof of its correctness are introduced.

A method for pre-training in knowledge base completion is introduced, the first of its kind, significantly improving the results on multiple smaller datasets. The obtained models outperform much larger and highly trained models, which are trained on more general language-modelling tasks. To better understand the behaviour of the obtained

models for knowledge base completion, the first diagnostic dataset for pre-trained knowledge base completion models is introduced, demonstrating how stereotypes in the pre-training data can affect the predictions of a model on the target knowledge base.

The future developments of both task-specific pre-training and bias detection are discussed, motivating future research directions in the field.

# Publications

This thesis is based on the following publications:

- Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019b. A surprisingly robust trick for the Winograd schema challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4837–4842, Florence, Italy. Association for Computational Linguistics.
- Vid Kocijan, Oana-Maria Camburu, Ana-Maria Cretu, Yordan Yordanov, Phil Blunsom, and Thomas Lukasiewicz. 2019a. WikiCREM: A large unsupervised corpus for coreference resolution. 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 4303–4312, Hong Kong, China. Association for Computational Linguistics.
- .
- Vid Kocijan, Oana-Maria Camburu, and Thomas Lukasiewicz. 2021. The gap on gap: Tackling the problem of differing data distributions in bias-measuring datasets. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, February 2–9, 2021*. AAAI Press.
- .

During my DPhil, I have also co-authored the following publications, not part of my thesis:

- Yordan Yordanov, Oana-Maria Camburu, Vid Kocijan, and Thomas Lukasiewicz. 2020. Does the Objective Matter? Comparing Training Objectives for Pronoun Resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4963–4969, Online. Association for Computational Linguistics.
- Patrick Hohenecker, Frank Mtumbuka, Vid Kocijan, and Thomas Lukasiewicz. 2020. Systematic comparison of neural architectures and training approaches for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8554–8565, Online. Association for Computational Linguistics.

# Abbreviations

| abbreviation | full name |
|---|---|
| AI | artificial intelligence |
| CNN | convolutional neural network |
| GPU | graphics processing unit |
| GRU | gated recurrent unit |
| H@N | Hits at N |
| KB | knowledge base |
| KBC | knowledge base completion |
| LM | language model |
| LSTM | long-short term memory |
| MR | mean rank |
| MRR | mean reciprocal rank |
| NER | named entity recognition |
| NLP | natural language processing |
| OIE | open information extraction |
| OKBC | open knowledge base completion |
| RNN | recurrent neural network |
| SOTA | state-of-the-art |
| WSC | Winograd Schema Challenge |

# Contents

# Chapter 1

# Introduction

Human language is a complex system of communication whose ambiguity and complexity make it challenging to process automatically. Some of the most successful approaches to natural language processing (NLP) involve the analysis of large corpora of text to detect subtle patterns and correlations between the input and output data. The employment of artificial neural networks with many layers, so-called *deep learning*, has become the most common approach to problems involving automatic processing of human language. However, the more parameters such a model has, the more annotated data samples it requires to be trained (**?**).

Collecting labelled data to train a neural network is usually expensive and can require a lot of manual labour. To improve the performance of a model without collecting more task-specific data, one can first pre-train the model on a related task with an abundance of data. If the two tasks share enough common properties, the final model can perform better than if it were trained on the data from the target task only. This process is called *transfer learning* and can be incredibly useful in NLP, where all tasks share common properties — grammar and vocabulary.

Pre-training models on related unsupervised tasks, such as language modelling or masked language modelling, has already shown to be incredibly successful across various NLP tasks where large-scale pre-trained models achieve better overall performance than specific architectures built for one target task (Peters et al., 2018; Devlin et al., 2019). However, pre-training a model on unstructured text, usually collected from the internet, comes with drawbacks. Pre-training data can contain undesired societal biases that the model can mimic and amplify (Bolukbasi et al.,

2016; Zhao et al., 2018; Rudinger et al., 2018; Webster et al., 2018). This is an undesired side effect that can result in unfair treatment and propagation of existing divisions within the society when used at scale. As such, it is important to investigate its sources and reduce its influence on the final predictions. In this thesis, I investigate the impact that pre-training on unstructured text data has on the behaviour of the models. I develop task-specific pre-training approaches to coreference resolution and knowledge base completion and demonstrate that they produce better result than *general purpose* pre-training objectives. Finally, I develop several diagnostic tests to analyse the obtained models, detecting societal bias and other properties.

## 1.1 The Role of Pre-training in Natural Language Processing

The common property of most NLP tasks is the input structure, motivating the use of multi-task or transfer learning. An approach that takes advantage of this property was first implemented by Ando and Zhang (2005) by training a model on multiple tasks simultaneously. It later became common to use a separate task to train parts of a model, most notably, vector embeddings of words (**??**Pennington et al., 2014). Training parts of a model on a separate task was taken to the extreme with large-scale pre-trained language models such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2020), and GPT-3 (Brown et al., 2020).[1] Pre-trained in an unsupervised manner on colossal corpora of text scraped from the internet, these models require only a few training examples to match the performance of fully-trained models designed and trained for the target task only. The trends in their results are clear: the more pre-training we do with a model, the better its fine-tuned performance is.

The implications of these results are far-reaching as pre-trained models quickly made their way from the academic environment into the industry. The BERT lan-

[1]Not all of these are language models in the strict sense, because not all of them can be used to assign a probability to the input text, however, I choose to follow related literature and use *language model* as an umbrella term for brevity.

guage model, for example, is now used to re-word Google search queries.[2] The best-performing approaches across most NLP benchmarks usually make use of one of the pre-trained language models rather than training from scratch. The majority of such pre-trained models are trained on next-word-prediction (Howard and Ruder, 2018), masked-word-prediction (Devlin et al., 2019), or a similar variation of language modelling (Joshi et al., 2020).

Using general-purpose pre-training tasks such as next-word-prediction may be a good choice when creating a universal pre-trained model that we aim to use for multiple tasks. However, knowing that more similarity between the pre-training and fine-tuning task results in better transfer of knowledge, we may choose to design the pre-training task accordingly. Even large-scale pre-trained language models, such as BERT, can benefit from additional target-specific pre-training (Phang et al., 2018). This demonstrates the importance of the pre-training task choice and motivates the development of task-specific pre-training approaches, preferably unsupervised. In this work, I introduce two such approaches, one in Chapter 3, and one in Chapter 5.

## 1.2 Societal Bias and the Importance of Fairness in Natural Language Processing

One of the main reasons for the use of machine learning in NLP is the lack of need for explicit rules, as the model automatically detects relevant statistical correlations and patterns from the training data instead. Unfortunately, not all statistical patterns in the training data are beneficial. Any sort of training data collected from the real world reflects the habits and biases of the society, even undesired ones (Bolukbasi et al., 2016). An algorithm that is designed to pick up and exploit statistical patterns in data can base its predictions on detrimental societal stereotypes if they turn out to be good predictors. The mass use of such systems in practice can perpetuate these stereotypes and result in unfair/unjust treatment of marginalized groups.

[2] `https://blog.google/products/search/search-language-understanding-bert/`

However, reliance on damaging stereotypes is not the only harmful manifestation of societal bias. Data randomly sampled from English Wikipedia is prone to underrepresent women as 8 out of 9 Wikipedia biographies are about men (Webster et al., 2018). A model trained on an imbalanced dataset is unlikely to perform equally well on the overrepresented and underrepresented type of data. Even if the reader is not convinced by political motivations of fairness in machine learning, an imbalanced performance across the userbase is an undesired trait that can curb its widespread use.

While anecdotal evidence and post-hoc analysis are often enough to demonstrate that a model is biased, more systematic and well-defined methods for comparison are required for the development of bias-neutering methods (Gonen and Goldberg, 2019). To this end, all tasks in this work are accompanied by a theoretically-grounded metric of bias, as well as other informative diagnostics that can help by providing a better understanding of the model behaviour. In Chapter 4, I address the problem of an imbalanced test set in the GAP dataset for coreference resolution (Webster et al., 2018), providing an alternative metric that alleviates the issue. Finally, I design a completely new benchmark, called DOGE, to investigate the role of historical gender bias in pre-trained knowledge base completion models in Chapter 6.

## 1.3 Research Focus and Outline

Both transfer learning and bias in natural language processing are broad topics that have and continue to be active areas of research. In this work, I narrow my focus on two specific tasks: coreference resolution (Levesque et al., 2011; Webster et al., 2018) and knowledge base completion (KBC) (Dettmers et al., 2018; Balažević et al., 2019; Abboud et al., 2020), observing the behaviour of pre-trained models on these tasks. There are multiple reasons for this choice. Firstly, both tasks are considered unsolved and are active areas of research. Secondly, both can benefit from external background or commonsense knowledge, making pre-training on unstructured data beneficial. Finally, both allow for simple investigation into potential bias, e.g., by observing the changes of the coreference model behaviour depending on the gender of the pronoun (Rudinger et al., 2018; Zhao et al., 2018;

Webster et al., 2018). In this work, I introduce a novel pre-training approach for both of the tasks and then analyse the obtained models, introducing novel metrics and datasets where necessary.

**Outline of Coreference Resolution Work** Coreference resolution is an instance of a task where humans use background knowledge without explicitly thinking about it. In a sentence "*A student fell asleep during a professor's lecture because* ***he*** *was so bored.*", the human reader automatically understands that the pronoun *he* refers to the student. Changing the word *bored* into *boring*, however, creates a sentence where the same pronoun refers to the professor, indicating that the knowledge of grammar is insufficient for the correct understanding of the sentence. Instead, the human reader implicitly knows that the *bored* person is the one falling asleep and not the *boring* one. Winograd Schemas, as such examples are called (Levesque et al., 2011), are challenging cases of coreference resolution that require the human reader to use their *commonsense* or *background knowledge* to resolve them. A program designed to automatically resolve such cases should in principle have to do this as well.

To introduce background knowledge through pre-training, I create a pre-training protocol designed specifically for coreference resolution. Looking for short text passages where the same noun or personal name appears multiple times, I mask the non-first occurrence of a repeated entity and train the model to predict which of the candidates from the text was masked out. Using this pre-processing procedure, I automatically collect two large datasets for coreference resolution pre-training, MaskedWiki and WikiCREM. Additional pre-training of an already pre-trained model Bert (Devlin et al., 2019) on these datasets boosts the performance of trained models and leads to improvements across different benchmarks — including the Winograd Schema Challenge, a challenging collection of Winograd Schemas, where previous attempts obtained only a chance-level performance (Levesque, 2011). This line of work is described in Chapter 3.

As highlighted in the previous section, such pre-training usually does not come without negative side-effects and undesired behaviour. To investigate how additional pre-training impacts a model, I evaluate the obtained pre-trained models on the Gap dataset for coreference resolution (Webster et al., 2018), which can

additionally be used to assign a *gender bias* score to each model. However, I show the benchmark to have a potentially problematic imbalance in its test data distribution. Feminine examples on average contain more potential candidates which are also often further away from the pronoun in question compared to the masculine examples. These are at least two properties that can make examples *easier* or *harder*, making the models seem more or less biased than they actually are. In Chapter 4, I introduce a method for weighting test instances that removes the impact of such undesired imbalances on the result. Upon evaluation of models on the balanced benchmark, I find that additional pre-training on the introduced data improves (rather than worsens) the gender-bias score of the models.

**Outline of Knowledge Base Completion Work** Knowledge base completion, on the other hand, is an area that has seen few attempts at transfer learning, and no attempts to analyse the existence of undesired societal biases in trained models. With the goal to deduce missing facts in the knowledge base from existing ones, the motivation for the use of transfer learning might be natural. However, the structured nature of knowledge bases makes it less obvious how to make use of unstructured text, as phrases in text can be hard to connect to entities and relations in the target knowledge base. In Chapter 5, I introduce a modification to the architecture of existing models for knowledge base completion (Dettmers et al., 2018; Balažević et al., 2019) that allows them to be used on non-canonicalized knowledge bases — e.g., collections of facts extracted automatically from unstructured text. Replacing fixed entity and relation embeddings with encoders allows us to pre-train these models on data collected in an unsupervised manner. I show that this improves the performance of models on canonicalized target knowledge bases, providing a practical way of incorporating external knowledge from unstructured data into structured knowledge bases.

In Chapter 6, I investigate what type of knowledge pre-trained models for knowledge base completion contain and how they behave. To do so, I introduce a novel diagnostic dataset for knowledge base completion that tests different types of behaviour — gender bias being one of them. Each model is investigated for coverage of different areas of knowledge, its ability to perform deductive reasoning, consistency in the presence of synonyms and inverse relations, and susceptibility to

gender stereotypes. I find that all systems associate feminine names with stereotypically feminine occupations and masculine names with stereotypically masculine occupations, even in the presence of facts that contradict this. However, I demonstrate that the inability of a model to change its biased predictions in spite of contrary facts stems from the general inability to perform deductive reasoning rather than deeply ingrained biases.

Any background knowledge or related work, necessary to understand the proposed and developed ideas is discussed in Chapter 2. Each chapter finishes with a summary of work, main conclusions and open questions, with more overarching conclusions and discussion provided in Chapter 7.

# Chapter 2

# Preliminaries

This chapter covers the relevant background, necessary for the understanding of the proposed methods and performed analyses. Its purpose is to define the terminology, introduce the necessary background, and review recent advances in the field. For certain problems, the field has already advanced beyond the solutions that I have introduced in this thesis. Methods that were introduced afterwards are only briefly mentioned in this chapter as they are not crucial to the understanding of the thesis. Instead, I discuss them at the end of each chapter that introduces my contribution to those problems, where I also analyse the impact of my work on the development of the field.

The rest of this chapter is split into five sections. The first one introduces a brief and non-exhaustive background on neural networks and gradient optimisation. The second section introduces the notion of pre-training in natural language processing. The third section focuses on the problem of coreference resolution and the existing approaches, with a lot of focus on the Winograd Schema Challenge. The third section of this chapter introduces the problem of knowledge base completion, common datasets, and common approaches to it. The fourth section defines the problem of societal bias in natural language processing, putting special focus on gender bias and grammatical gender.

## 2.1 Neural Networks and Gradient-based Optimisation

The majority of work in this thesis focuses on data and metrics, treating the models as black boxes. However, the reader will probably find the work more intuitive if they are already familiar with neural networks, gradient-based optimization, as well as specific neural architectures such as recurrent neural networks, and transformers. This section provides a brief and high-level introduction of these terms. Any readers who would like more a more detailed introduction are directed to Goodfellow et al. (2016) and Vaswani et al. (2017). Unless otherwise specified, this section is taken from Goodfellow et al. (2016).

Neural networks are a class of models for machine learning. They usually consist of stacked matrix multiplications with differentiable non-linearities between them, which gives them the expressive power to approximate any smooth function. To achieve useful task-specific inductive bias, they are often combined with components such as convolutions, gating mechanisms, and other operations with desirable mathematical properties. A neural network thus contains many parameters, such as matrix elements and vectors, whose values are tuned to obtain the desired behaviour of the network.

Under the supervised learning paradigm, used in this work, parameters of a neural network are usually tuned by optimizing the correctness of its predictions on training examples. The most common approach to this optimisation problem are gradient-based methods. At every step of the training, a gradient of the function w.r.t. a set of training examples is computed. This gradient is then used to estimate how a change of parameters will affect the correctness of the model predictions and update their values accordingly. The exact details of these steps differ between different gradient-based optimisation methods and are omitted for brevity.

Two classes of neural networks are particularly relevant to work in this thesis, recurrent neural networks (RNNs) and transformers, both used to process sequential data. A recurrent neural network processes an input sequence (e.g., text) one input element at the time, resembling an automaton. In each step, an RNN takes its previous "hidden state" vector and the new element as the input, and computes

an output and the next hidden state. For a more detailed discussion of different types of RNNs, the reader is directed to (Goodfellow et al., 2016).

Transformers are an alternative approach to sequence processing. Unlike RNNs, transformers process the entire input sequence at the same time through the "attention mechanism" — a neural network component that relies on repeated computing of weighted sums of its inputs. Removing the need for sequential processing allows easier distributed learning and thus simplifies the training at scale, allowing the training of larger and more powerful models. For a more detailed description of transformers, the reader is directed to Vaswani et al. (2017).

## 2.2 Pre-training in Natural Language Processing

Any model claiming the ability to process human language requires an immense amount of background knowledge. The Oxford English Dictionary alone records over $600,000$ different English words[1] and to handle even a small fraction of these, a machine learning model would require an enormous and diverse training set. All models operating on the English language share this vocabulary, prompting the development of common resources such as pre-trained models or word embeddings trained on large unsupervised corpora. In this chapter, I will introduce the relevant methods for the unsupervised pre-training of models. While pre-training on supervised tasks is possible and sometimes even more sample-efficient (Conneau et al., 2017), the accessibility of raw text makes unsupervised approaches the more scalable option, typically resulting in better performance overall. The methods introduced in this chapter are a selection of the most relevant and commonly used methods for pre-training, and tend to be based on language modelling. The list is in no way exhaustive as only the methods relevant to the rest of the thesis are covered.

As an example of a pre-training method that is not based on language modelling, I highlight the unsupervised multi-lingual paraphrasing approach by **?**. They show that one can obtain a model for machine translation in an unsupervised manner by training to paraphrase articles about the same news. This is

[1] Taken from `https://www.oed.com/` on 23 June 2021

just one example of the usefulness and importance of task-specific unsupervised pre-training, with two more introduced in later chapters of this thesis.

**Language Modelling** A *Language Model* is a computational model that assigns a probability score to the input sequence of words (Bengio et al., 2003). Given an input sequence $w_1, w_2, \dots w_n$, the probability $\mathbb{P}(w_1, w_2, \dots, w_n)$ can be computed as the product of conditional probabilities:

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

A language model is thus usually trained to predict the probability of the next word by conditioning on the preceding words. This definition is agnostic to the architecture of the model, but as a pre-training objective it is mostly used on recurrent neural networks (RNNs) (Howard and Ruder, 2018) and transformers (Brown et al., 2020). The above definition describes the input as a sequence of words, however, this formulation also works if the input is split into characters, subword tokens, morphemes, n-grams, or other substrings. In the rest of the thesis such an elementary block of the input sequence is denoted as a *token*.

A language model is usually obtained through statistical analysis of large corpora of text. In the case of neural networks, this is achieved through minimization of a loss function with stochastic gradient optimization methods (Goodfellow et al., 2016). The natural product of such training is a model that can predict the most probable next token or the probability distribution over the possible next tokens given the sequence so far. Such a model can be used for natural language generation by generating one token of text at the time, however, it lacks bidirectionality. Suppose we wanted to use such a model to predict a missing token in the middle of a sentence. For the model to make a prediction that additionally considers tokens following this token, the model has to be modified, or ran multiple times, once for each possible missing token. The latter option can be computationally expensive. A simple trick to provide such bidirectional context is to train a separate language model that processes the sentence from the opposite direction (Peters et al., 2018).

It is up to the implementation of a specific model to determine how the probability $\mathbb{P}(w_i | w_1, w_2, \dots, w_{i-1})$ is computed and how the model is trained. The

obtained trained model is usually referred to as the *pre-trained model* and is typically used as the starting point, i.e. the initialization for any subsequent training, which is also called *fine-tuning*. It is worth noting that this is not the only way in which a pre-trained language model can be used. ELMO (Peters et al., 2018), for example, is not fine-tuned, but is used instead to compute *contextualized word embeddings*, which are then used as an input to the target model. However, uses of pre-trained models, other than the initialization of the target models, do not appear in this thesis and their description will be omitted for brevity.

**Masked Language Modelling** A *Masked Language Model* is obtained with a modified language modelling objective, designed to capture bidirectionality better. Instead of predicting the next token in the sequence, the model is designed and trained to predict one or more missing tokens in the input sequence (Devlin et al., 2019). Masked language modelling, sometimes also referred to as *Cloze task* (Taylor, 1953), is not language modelling in the strict sense, because a masked language model cannot be naturally used to assign a probability to the entire input sequence. As already noted in the introduction, I use the term *language model* as an umbrella term for any type of model pre-trained on unstructured text. This choice of terminology is used for brevity as the distinction between left-to-right and masked language models is not required most of the time.

Unlike left-to-right language modelling, masking text passages does not have a unique way of selecting masked tokens and or choosing in which order to predict them. Devlin et al. (2019), when training BERT, choose 15% of the input positions and replace 80% of them with a `[MASK]` token, 10% with a random token from the vocabulary, and 10% with the original token, all chosen uniformly at random. For comparison, Joshi et al. (2020) only mask contiguous spans, while Yang et al. (2019) never replace inputs with random or original tokens but instead train the model to predict the masked tokens in a random order, one-by-one. These are not the only strategies employed in masked language modelling, however, their result is usually the same – a model that can fill in a small fraction of missing tokens in the input sequence.

In addition to the masked language modelling objective, these models are often further trained on an additional objective of *next sentence prediction* (Devlin et al.,

2019) or *sentence order prediction* (Lan et al., 2020), which aims to capture high-level semantics. In these objectives, the models are trained to detect whether two pieces of input text come one after another or if they are given in the correct order. Their aim is to capture some level of *reading comprehension*, however, experiments from different authors give somewhat conflicting results on their usefulness (Devlin et al., 2019; Lan et al., 2020; Yang et al., 2019; Joshi et al., 2020; Liu et al., 2019c). This does not affect any research conducted in this work, as pre-trained masked language models are only used for their ability to fill in missing tokens.

## 2.3 Coreference Resolution and the Winograd Schema Challenge

The aim of coreference resolution is to identify which phrases within the text corefer to each other, i.e. refer to the same entity. As can be seen from Figure 2.1, coreference plays a big role in English language and is non-trivial. Different entities may overlap, the resolution of pronouns can be ambiguous, and lexical similarity does not always entail coreference, e.g. *Tina* and Tina Turner are not the same entity.

> Myra had **her second daughter** with Billy a year later and named **her Mercedes**. Billy left Myra and **his daughters** soon after **Mercedes** was born. At age 19, Myra gave birth to *Tina* following a one-night stand with a married man named Marvin Bassman who refused to be involved in *his daughter's* life. She named *her daughter* after Tina Turner.

Figure 2.1: An example of coreference resolution, taken from the GAP dataset, with all coreferences labelled (Webster et al., 2018). Substrings written in the same colour or with the same font refer to the same entity. In the dataset, only the underlined pronoun is annotated.

Determining which words co-refer is usually a necessary step in tasks that require any sort of language understanding. It has to be handled either implicitly by the model or explicitly with an external tool. In this thesis, I focus mainly on pronoun resolution, a specific subfield of coreference resolution, which focuses

only on pronouns rather than any noun phrase. The main reason for this choice is the availability of performance-measuring and diagnostic benchmarks for the task. In English, pronouns are gendered, which allows for a controlled investigation of the impact of grammatical gender on model behaviour. While the role of gender and bias in coreference resolution is an important motivation for the choice of this task, its discussion is postponed until Section 2.4, where both grammatical gender and bias are discussed more extensively. Instead, this section centers on the introduction of pronoun resolution and the existing attempts at solving it.

Particular attention is drawn to the Winograd Schema Challenge (WSC) (Levesque, 2011), a particularly difficult case of pronoun resolution, which doubles as a commonsense reasoning benchmark. Unlike benchmarks for pronoun resolution collected from real-world documents, WSC is manually designed to be challenging and to require a significant amount of background knowledge. Moreover, its specific structure allows for a controlled environment to test certain types of gender bias, resulting in multiple bias-measuring WSC datasets being used in my work. In the first part of this section, I give an overview of general methods and relevant datasets for pronoun disambiguation. The Winograd Schema Challenge and relevant attempts at it are presented in the second part.

### 2.3.1 Pronoun Disambiguation in Natural Language Processing

Pronouns in English are most commonly, though not always, anaphoric, i.e. a pronoun generally refers to some entity mentioned by a noun phrase earlier in the text. The problem is thus a subset of the broader problem of anaphoric resolution. There are exceptions to this rule, where the pronoun precedes the entity it refers to. Such examples were, for example, specifically targeted by the GAP dataset collection (Webster et al., 2018). The rest of the chapter focuses specifically on pronouns, but for a more extensive survey of anaphoric reference, I direct the reader to the works of Poesio et al. (2011), Poesio et al. (2016), and Kehler et al. (2008).

Early attempts at pronoun resolution mostly revolved around the role of commonsense knowledge in disambiguation, e.g. (Charniak, 1972; Wilks, 1975; Schank

and Abelson, 1977; Hobbs, 1979; Hobbs et al., 1993), or the structural characteristics of the text, such as parallelism and focus (Grosz, 1977; Sidner, 1979; Kameyama, 1986). Implementations were almost always proof-of-concept and their development was guided by intuition rather than by evaluation, the latter being rare (McDermott, 1976). Both commonsense knowledge and the structural characteristics of the text are used in the interpretation of language. When different constraints conflict, the rules for resolving the conflict are complex: pragmatic preferences can overrule formal preferences, but sometimes it is the other way around. The conflict can also lead to unintelligible sentences (Kehler et al., 2008).

The MUC-6 and MUC-7 conferences (Message Understanding Conferences) (Grishman and Sundheim, 1996; Chinchor, 1998) included evaluation of coreference resolution systems. Both conferences involved the construction of large text corpora, drawn from natural sources, which were subsequently carefully hand-annotated. These corpora were later replaced by OntoNotes as the most popular dataset for coreference resolution (Hovy et al., 2006), which is still the de-facto standard for document-level coreference resolution to date.

Systems developed for these datasets were initially rule-based or used the machine learning technology available at the time (Lee et al., 2013). Any introduced systems for coreference resolution were both trained and evaluated on these corpora. These systems are slowly being replaced by deep-learning-based approaches since 2015 (Clark and Manning, 2015; Wiseman et al., 2016; Lee et al., 2017, 2018); and, since 2019, becoming particularly reliant on pre-trained transformers (Liu et al., 2019a; Joshi et al., 2019, 2020). In both statistical and deep learning-based approaches commonsense knowledge has been incorporated into systems only to the extent where the learning procedure implicitly abstracts it from text corpora.

Since general-purpose datasets for coreference resolution such as OntoNotes contain difficult or otherwise interesting samples only in small quantities, their influence is generally obscured in the overall score. To this end, multiple smaller, usually *snippet-level* datasets were released, often targeting specific linguistic phenomena or sentence structures (Webster et al., 2018; Emami et al., 2019; Levesque, 2011). These datasets tend to be small, often manually created or manually modified, and are not annotated with all coreferences in the text, but rather only with coreferences of a specific type. Out of all snippet-level datasets, the GAP dataset

and Winograd Schema Challenge are particularly relevant to this work. A more detailed description of GAP is given in the following paragraph and Winograd Schema Challenge is presented in the following section.

**Gendered Ambiguous Pronouns (GAP) Dataset** GAP is a corpus of challenging examples of pronouns from the English Wikipedia. It was introduced as a gender-balanced dataset, so that exactly half of the pronouns are masculine and half are feminine (Webster et al., 2018). For each text span, one pronoun has to be resolved. Pronoun resolution is treated as a binary classification task, with the goal to determine whether a single candidate is the referent of the pronoun or not. Candidates are not given as input and the model is expected to find them on its own. However, it is guaranteed that the candidates are always personal names from the input text and that at most one of them is the correct referent. An example from GAP can be found in Figure 2.1. Only the underlined pronoun is labelled and evaluated in the dataset. The rest were annotated only for the illustration.

To obtain a more challenging dataset, several measures were taken during the data-collecting procedure. During the collection time the authors automatically detected sentences from a specified list of grammatical patterns that were estimated to contain more challenging examples. Moreover, instances where the pronoun appears before or in between the two candidates in question were upsampled to increase their representation. Finally, to ensure equal representation of both genders, the more common masculine sentences were downsampled. The dataset also comes with a bias score, defined to compare the performance of a model on the feminine and masculine subset of the dataset. The discussion of the GAP dataset usage for bias detection is postponed until later and can be found in Section 2.4.

### 2.3.2 Winograd Schema Challenge

The Winograd Schema Challenge was proposed by Levesque (2011), inspired by an example in Terry Winograd's 1972 doctoral thesis, *Understanding Natural Language*. The example consists of a pair of sentences:

> The city councilmen refused the demonstrators a permit because they feared violence.
> The city councilmen refused the demonstrators a permit because they advocated violence.

In the first sentence, the pronoun *they* is naturally interpreted as referring to the city councilmen; in the second it is naturally interpreted as referring to the demonstrators. The only difference between the two sentences is that the first one contains the word *feared*, whereas the second one contains *advocated*. Hence the different referents chosen for *they* must somehow reflect the different choice of word.

Levesque (2011) therefore proposed to use these kinds of sentences as a test for the depth of understanding of AI natural language programs. In particular, he defined a *Winograd schema* as a pair of sentences,[2] comparable to Winograd's example above, with the following features.

1. The two sentences are identical except for one or two words: *feared* vs. *advocated* in Winograd's example.

2. The two sentences both contain two noun phrases and a preposition: *the city council*, *the demonstrators*, and *they* in Winograd's example.

3. The natural readings of the two sentences in isolation would assign different choices of referents for the prepositions.

4. Simple feature matching, known as *selectional restrictions*, will not suffice to do the disambiguation. For instance, the pair of sentences

   > The women stopped taking the pills because they were [pregnant/carcinogenic].

   is disallowed, because pills cannot be pregnant and women cannot be carcinogenic.

[2] In later work, this was often relaxed so that each element of the pair could be a two-sentence text.

5. Matching based on simple frequency of co-occurrence will not suffice to do the disambiguation. For instance, the pair of sentences

   > The racecar zoomed by the school bus because it was going so [fast/slow].

   would be disallowed, because the words "racecar" and "fast" tend to appear together in text.

6. Both sentences must seem natural and must be easily understood by a human listener or reader; ideally, so much so that, coming across the sentence in some context, the reader would not even notice the potential ambiguity.

Since the two sentences only differ in one *special word*, any system that fails to consider it a clue is bound to resolve both cases in the same way — achieving only a 50% (chance-level) performance. Levesque (2011) argued that this test would be a better test of language understanding than the Turing test—trivial to evaluate, difficult to game.

#### 2.3.2.1 Winograd Schema Challenge Datasets

Throughout the years, multiple datasets inspired by this idea were created. In this section, I cover all the datasets that were built to evaluate the performance of a model. The two bias-measuring WSC datasets, WinoGender and WinoBias (Rudinger et al., 2018; Zhao et al., 2018), are presented in Section 2.4. The WinoGrande and WinoFlexi datasets, discussed in this section, were released only after my work on WSC had been already concluded and are thus not present in the experimental chapters.

**Original Collection of Winograd Schemas** The first collection of 100 Winograd schemas was published together with the introduction of the Winograd Schema Challenge (Levesque, 2011).[3] Examples are constructed manually by AI experts, with the exact source for each example available. At the time of writing, there are 285 examples available; however, the last 12 examples were only added recently.

[3] `https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html`

To ensure consistency with earlier models, several authors often prefer to report the performance on the first 273 examples only. These datasets are usually referred to as WSC285 and WSC273, respectively.

Trichelair et al. (2018) have observed that 37 sentences in the WSC273 dataset (13.6%) are conceptually easier than the rest. The correct candidate is commonly associated with the rest of the sentence, while the incorrect candidate is not. An example of such a sentence is

*In the storm, the tree fell down and crashed through the roof of my house. Now, I have to get* ***it*** *[repaired/removed].*

*The roof* is commonly associated with *being repaired*, while *the tree* is not. They call these examples *associative* and name the rest *non-associative*. Moreover, they find that models often perform much better on the associative subsets.

Additionally, 131 sentences (48% of WSC273) were found to form meaningful examples if the candidates in the sentence are switched. An example of such a sentence is

*Bob collapsed on the sidewalk. Soon he saw Carl coming to help.* ***He*** *was very [ill/concerned].*

In this sentence, *Bob* and *Carl* can be switched to obtain an equivalent example with the opposite answers. Such sentences were named *switchable*. Trichelair et al. (2018) encourage future researchers to additionally report the consistency on the *switchable* dataset, when the candidates are switched, and when they are not.

**Winograd Schema Challenge in other languages** While the inspiration and original design of the challenge was in English, translations into other languages exist. Amsili and Seminck (2017) translated the collection of 144 Winograd schemas into French, and 285 original Winograd schemas were translated into Portuguese by Melo et al. (2020). The authors of both French and Portuguese translations report the need to make some changes to the content to avoid unintended cues, such as grammatical gender. In the case of Portuguese, 8 sentences had to be dropped as no appropriate translation could be found.

Translations to Japanese[4] and Chinese[5] are available on the official web page

[4] `http://arakilab.media.eng.hokudai.ac.jp/~kabura/collection_katakana.html`

[5] `https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSChinese.html`

of the challenge. A second translation into Chinese, *Mandarinograd*[6] is reported by Bernard and Han (2020) together with an account of the difficulties involved in the translation process. The Chinese Winograd Schema Collection (CLUEWSC2020) is an anaphora/coreference resolution task where the model is asked to decide whether a pronoun and a noun (phrase) in a sentence co-refer (binary classification). The collection contains 1838 questions hand-selected from thirty-six contemporary literary works in Chinese. The anaphora relations have been hand-annotated by linguists. The dataset is part of CLUE (Chinese Language Understanding Evaluation), a collection of Chinese language benchmarks analogous to GLUE (Xu et al., 2020)

Slovene translation of SuperGLUE benchmark includes the translation of WNLI into Slovene (?). Finally, Indic General Language Understanding Evaluation (IndicGLUE) (Kakwani et al., 2020) includes translations of GLUE's WNLI into Hindi, Marathi, and Gujarathi.

**Definite Pronoun Resolution Dataset** The Definite Pronoun Resolution (DPR) dataset is an easier variation of the Winograd Schema Challenge (Rahman and Ng, 2012). The constraints on the Winograd schemas have been relaxed, and several examples in the dataset can be resolved by observing word co-occurrence. The dataset consists of 1322 training examples and 564 test examples constructed manually. 6 examples in the training set reappear in WSC273 in a very similar form. These should be removed when training on DPR and evaluating on WSC273. This dataset is also referred to as WSCR, as named by Opitz and Frank (2018).

An expanded version of this dataset, called WINOCOREF, has been released by Peng et al. (2015), who further annotate all previously ignored mentions (in their work, a mention can be either a pronoun or an entity) in the sentences that were not annotated in the original work. In this way, they add 746 mentions to the dataset, 709 of which are pronouns.

**Pronoun Disambiguation Problem Dataset** The Pronoun Disambiguation Problem (PDP) dataset consists of 122 problems of pronoun disambiguation collected from classic and popular literature, newspapers, and magazines. Because

[6] `https://gitlab.com/vanTot/mandarinograd/`

constructing Winograd schemas according to the original guidelines was a difficult manual process, PDPs, which were collected and vetted rather than constructed, were intended to be used as a gateway set before administration of the Winograd Schema Challenge (Davis et al., 2017). Each PDP, once collected, was vetted (and sometimes modified) to ensure that (like Winograd schemas) the problem was of the sort that humans use commonsense knowledge to disambiguate and not resolvable through analysis of word co-occurrence. Although each PDP was vetted in order to remove examples where sentence structure would help find the answer, there was no *special* word, and thus, unlike Winograd schemas, no guarantee that sentence structure could not be exploited. PDPs are therefore expected to be easier than Winograd schemas.

Example of PDP from the PDP dataset: *Do you suppose that Peter is responsible for the captain's illness? Maybe he bribed the cook to put something in* ***his*** *food.*
The referent of **his** is: (a) Peter or (b) the captain.

Before the Winograd Schema Challenge was administered, 62 examples of PDPs were published [7] and 60 PDPs were included in the Winograd Schema Challenge that was administered at IJCAI 2016. [8] (Davis et al., 2017) A separate corpus of 400 sentences was collected semi-automatically from online text, with less vetting, by Davis and Pan (2015).[9]

**Winograd Natural Language Inference Dataset** The Winograd Natural Language Inference (WNLI) dataset is part of the GLUE benchmark (Wang et al., 2018) and is a textual entailment variation of the Winograd Schema Challenge. An example from WNLI is given below with the aim to determine whether the hypothesis follows from the premise.

**Premise:** *The city councilmen refused the demonstrators a permit because they feared violence.*
**Hypothesis:** *The demonstrators feared violence.*
**Answer:** true / **false**

[7] `http://commonsensereasoning.org/disambiguation.html`
[8] `https://cs.nyu.edu/faculty/davise/papers/PDPChallenge.xml`
[9] `https://cs.nyu.edu/faculty/davise/annotate/corpus.xml`

The dataset consists of 634 training examples, 70 validation examples, and 145 test examples. Training and validation sets contain a major overlap with the WSC273 dataset, while test samples come from a previously unreleased collection of Winograd schemas. Not all examples in this dataset contain the *special* word and therefore do not come in pairs. In Section 3.1.1 I note that examples are much easier to approach if the Winograd schemas are transformed from the textual entailment back into the pronoun resolution problem and approached as such.

The same collection of examples as for WNLI is also used as part of the SuperGLUE benchmark (Wang et al., 2019a). There, the examples are phrased as a pronoun resolution task and not as a natural language inference problem. In this work, WNLI and SuperGlue WSC are considered the same dataset. Not only do they consist of the same examples, but all successful approaches to WNLI follow my introduced transformation mentioned in the previous paragraph.

**WinoGrande Dataset** The WINOGRANDE dataset is a large-scale Winograd Schema Challenge dataset ($44k$ examples) (Sakaguchi et al., 2020) collected via crowdsourcing on Amazon Mechanical Turk. To prevent the crowd from creating lexically and stylistically repetitive examples, the workers are primed by a randomly chosen topic from a WikiHow article as a suggestive context. Finally, the authors use an additional crowd of workers to ensure that the sentences are difficult but not ambiguous to humans. These measures were taken to ensure that there is no instance-level bias that models could exploit.

However, checking for instance-level cues is often not enough as models tend to pick on dataset-level biases. The authors additionally introduce the AFLITE adversarial filtering algorithm. They use a fine-tuned ROBERTA language model (Liu et al., 2019c) to gain contextualized embeddings for each instance. Using these embeddings, they iteratively train an ensemble of linear classifiers, trained on random subsets of the data and discard top-$k$ instances that were correctly resolved by more than 75% of the classifiers. By iteratively applying this algorithm, the authors identify a subset ($12,282$ instances), called $\text{WINOGRANDE}_{\text{debiased}}$. Finally, they split this dataset into training ($9,248$), development ($1,267$), and test ($1,767$) sets. They also released the unfiltered training set $\text{WINOGRANDE}_{\text{all}}$ with $40,938$ examples.

**WinoFlexi Dataset** Similarly to Sakaguchi et al. (2020), Isaak and Michael (2019) aim to construct a dataset through crowdsourcing. They build their own system and collect 135 pairs of Winograd schemas (270 examples). Unlike workers on WinoGrande, workers on WinoFlexi are not presented with any particular topic and are free to pick it on their own. Despite this, the authors find the collected schemas to have decent quality achieved through manual supervision between workers.

#### 2.3.2.2 Review of approaches to Winograd Schema Challenge

At least three different groups of approaches have been used to try to solve the Winograd Schema Challenge. One group consists of feature-based approaches, typically extracting information such as semantic relations. Additional *commonsense knowledge* is usually included in the form of explicitly written rules from knowledge bases, web searches, or word co-occurrences. The collected information is then used to make a decision using rule-based systems, various types of logics, or discrete optimization algorithms. Extraction of relevant information from the sentence is usually the bottleneck for these approaches. Given the nature of the challenge, even the slightest noise in the feature collection can render the problem unsolvable.

The second group are neural approaches, excluding language-model-based approaches, which I consider as a separate group due to the different nature of their development. Neural-network-based approaches usually read the sentence as a whole, removing the bottleneck of information extraction. To incorporate background information, these networks or their components are typically pre-trained on unstructured data, usually unstructured text or other datasets for coreference resolution. Common approaches to the tasks in this group take advantage of semantic similarities between word embeddings or use recurrent neural networks to encode the local context. These approaches tend to lack reasoning capabilities, as semantic similarity or local context usually do not contain sufficient information to solve Winograd schemas.

The third group includes approaches that make use of large-scale pre-trained language models, trained with deep neural networks, extensively pre-trained on large corpora of text. Some of the approaches then additionally fine-tune the

model on Winograd-Schema-Challenge-style data to maximize their performance. Approaches in this group achieve visibly better performance than approaches from the first two groups. Most of these approaches were introduced after, or were even heavily influenced by my work in Chapter 3. As such, they will only be briefly mentioned here, but further discussed in the end of that chapter.

**Feature-based Approaches to the Winograd Schema Challenge** This section covers the approaches that collect knowledge in the form of explicit rules from knowledge bases or internet search queries, and use logic-based systems or optimization techniques to deduce the answer. Results of methods that rely on search engines, such as Google, can be irreproducible as they strongly depend on the search results.

The majority of the early approaches to the Winograd Schema Challenge were feature-based or logic-based. While powerful in theory, their main bottleneck was usually the processing of the input data or relevant background knowledge. This is best demonstrated by Sharma (2019), who design a reasoning algorithm that solves 84.2% of examples in WSC285, if the input processing and background knowledge acquisition are done manually. The same algorithm solves fewer than 50% of examples when data processing is done with K-Parser (Sharma et al., 2015).

At the same time, methods in this group are the hardest to compare one to another. The models are often evaluated on specific subsets of WSC273 or other datasets, making direct comparison difficult. I thus chose not to summarize the results in a table like I do for neural and language-model-based approaches.

Approaches in this work usually work in three steps: firstly, the input sentence is processed to extract keywords and semantics. In the second step, the relevant background knowledge is extracted from various sources, most commonly knowledge bases and internet search queries. Finally, all this information combined is used as the input to the *reasoning algorithm*, which gives the final answer. The third step is where approaches conceptually differ the most one from another. The approaches employed were, for example, SVM-rankers (Rahman and Ng, 2012), integer linear programming (Peng et al., 2015), answer set programming (Sharma et al., 2015), message passing on a graph (Fähndrich et al., 2018), and formal logic (Isaak and Michael, 2016).

While many of these models achieve decent results on the data they have been evaluated on, their ability to generalize can be contested. Sharma et al. (2015), for example, report achieving 69% on a manually selected subset of WSC285, but when re-evaluated on a different subset of the data, the same model only achieved chance-level performance (Zhang and Song, 2018).

I highlight the Knowledge Hunter by Emami et al. (2018) as the first model to achieve a better-than-chance accuracy (57.1%) performance on the entire WSC273. Their system is completely rule-based and focuses on high-quality knowledge hunting, rather than reasoning, showing the importance of the former. Unlike neural approaches from later sections, this model is not negatively affected by switching candidates.

With the increase in popularity of large language models, interest in feature and reasoning-based methods decreased. However, Hong and Bennett (2020) note that hybrid approaches can be useful when solving schemas of a very specific type, using a language model when the symbolic approach does not yield an answer.

**Neural Approaches** This section contains approaches that rely on neural networks and deep learning, but do not use pre-trained language models. Models in this section are usually designed, built, and trained from scratch in contrast to models that use language models and are built on top of an off-the-shelf pre-trained neural network. Several ideas introduced in this section are later adjusted and scaled to language models; see next paragraph. Note that each work comes with a collection of model-specific architecture designs that are not covered in detail. The resources and approaches of neural methods are given in Table 2.1 and the main observations are summarized below.

Most of the methods in this section attempt to incorporate *semantic features* into the unsupervised training of the models. The listed pieces of work usually introduce tricks that aim to force a model to pick up the relevant training signals, either through specific architectures or through appropriate formulation of the training data. Success is often limited and many models are thus only evaluated on subsets of the dataset that suits the model better, making a direct comparison of the results non-trivial. One evident trend through time is the shift from methods

| | training resource | WSC273 | PDP |
|---|---|---|---|
| (Liu et al., 2017b) | cause-effect pairs | 70%† | – |
| (Liu et al., 2017a) | cause-effect pairs, Ontonotes | 52.8% | 58.3% |
| (Zhang and Song, 2018) | Wikipedia, dependency parser | 60.33%† | – |
| (Opitz and Frank, 2018) | DPR, InferSent▲ | 56% | – |
| (Wang et al., 2019b) | Gutenberg, 1 Billion Word | 62.4% | 78.3% |

Table 2.1: Resources, both data and tools, used by different neural approaches, ordered by the time of publication.
† denotes results obtained on modified version of the WSC273 dataset, usually a hand-picked subset. Such results are not guaranteed to generalize to full dataset.
▲InferSent refers to the pre-trained sentence embeddings by Conneau et al. (2017).

based on *(contextual) word embeddings*, e.g. (Zhang and Song, 2018) to end-to-end systems with a more complex set of objectives (Wang et al., 2019b).

**Language Model Approaches** This section covers the approaches that use neural language models to tackle the Winograd Schema Challenge. They use one or more language models that were trained on a large corpus of text. Several authors use large pre-trained language models, such as BERT (Devlin et al., 2019), and have to tailor their approach accordingly. Such works thus focus on better fine-tuning of such language models instead of inventing new architectures. The resources and results of all language-model based approaches are summarized in Tables 2.2 and 2.3, respectively.

Entries that correspond to (Kocijan et al., 2019b,a) are results of experiments introduced in this thesis, described in Chapter 3. A careful comparison of approaches in the listed works reveals multiple changes in trends over time. I list and discuss them below.

- The employed language models consistently increase in complexity. The first approaches use custom-made LSTMs trained specifically for the task, which are later replaced with large-scale language models such as BERT, which are in turn replaced with even larger T5 and GPT-3. This trend is not exclusive to Winograd Schema Challenge, and is a general ongoing trend in NLP.

- Appropriate fine-tuning grows in importance as it turns out to be the most efficient way of improving performance. Approaches that use language mod-

| | Language Model | fine-tuning or external data |
|---|---|---|
| (Trinh and Le, 2018) | custom LSTM | – |
| (Radford et al., 2019) | GPT-2 | – |
| (Klein and Nabi, 2019) | BERT | – |
| Prakash et al. (2019) | custom LSTM | internet querying |
| (Kocijan et al., 2019b) | BERT | MaskedWiki, DPR |
| (Kocijan et al., 2019a) | BERT | WikiCREM, DPR, GAP |
| (Ruan et al., 2019) | BERT | DPR |
| (He et al., 2019) | BERT | DPR |
| (Ye et al., 2019) | BERT | ConceptNet, DPR |
| (Sakaguchi et al., 2020) | RoBERTa | WinoGrande |
| (Melo et al., 2020) | custom LSTM | – |
| (Brown et al., 2020) | GPT-3 | – |
| (Yang et al., 2020) | RoBERTa | WinoGrande and generated data |
| (Lin et al., 2020) | T5 (3B) | WinoGrande |
| (Khashabi et al., 2020) | T5 | WinoGrande and QA tasks |
| (Lourie et al., 2021) | T5 | WinoGrande and Rainbow |

Table 2.2: Resources used by different language-model based approaches, ordered by the time of publication. With time, ever larger language models and more additional fine-tuning data were used.

els without fine-tuning, e.g. Trinh and Le (2018) and Klein and Nabi (2019), are outperformed by approaches that fine-tune the same models on WSC-like data, e.g. DPR. To increase the impact of fine-tuning, language models are often fine-tuned on more than one dataset. Earlier approaches usually use synthetic datasets (Kocijan et al., 2019b; Ye et al., 2019; Yang et al., 2020), while later approaches resort to multi-task learning on related tasks (Khashabi et al., 2020; Lourie et al., 2021).

- The introduction of the WinoGrande dataset made most other smaller datasets less interesting to researchers, even if these small datasets are higher in quality. The majority of papers published after the release of WinoGrande (Sakaguchi et al., 2020) only evaluated their work on that dataset.

Approaches employing large-scale pre-trained language models use various input and output formats, which can affect the performance of the model. For a detailed comparison of different training objectives and their impact on the

| | WSC273 | WNLI | PDP | WINOGRANDE |
|---|---|---|---|---|
| (Trinh and Le, 2018) | 63.7% | – | 70% | – |
| (Radford et al., 2019) | 70.7% | – | – | – |
| (Klein and Nabi, 2019) | 60.3% | – | – | – |
| Prakash et al. (2019) | 70.17% | – | – | – |
| (Kocijan et al., 2019b) | 72.5% | 74.7% | – | – |
| (Kocijan et al., 2019a) | 71.8% | 74.7% | 86.7% | – |
| (Ruan et al., 2019) | 71.1% | – | – | – |
| (He et al., 2019) | 75.1% | 89% | 90.0% | – |
| (Ye et al., 2019) | 75.5% | 83.6% | – | – |
| (Sakaguchi et al., 2020) | 90.1% | 85.6% | 87.5% | 79.1% |
| (Brown et al., 2020) | 88.3% | – | – | 77.7% |
| (Yang et al., 2020) | – | – | – | 80.0% |
| (Lin et al., 2020) | – | – | – | 84.6% |
| (Khashabi et al., 2020) | – | – | – | 89.4% |
| (Lourie et al., 2021) | – | – | – | 91.3% |

Table 2.3: Results of language-model based approaches on the most commonly used datasets. Less commonly used evaluation sets include DPR and WSC285. The only non-English result by Melo et al. (2020), who evaluate their model on a set of Portugese Winograd Schemas, is not included.

performance of the pre-trained language model, I refer the reader to the work by Yordanov et al. (2020) and Liu et al. (2020).

## 2.4 Knowledge Base Completion

A knowledge base (KB) is a collection of facts, stored and presented in a structured way that allows a simple use of the collected knowledge for applications. In this work, a *knowledge base* is a finite set of triplets $\langle h, r, t \rangle$, where $h$ and $t$ are *head* and *tail* entities, while $r$ is a binary relation between them. Manually constructing a knowledge base is tedious and requires a large amount of labour. To speed up the process of construction, facts can be extracted from unstructured text automatically using, for instance, open information extraction (OIE) tools, such as ReVerb (Fader et al., 2011) or more recent neural approaches (Stanovsky et al., 2018; Hohenecker et al., 2020). Alternatively, missing facts can be inferred from existing ones using *knowledge base completion (KBC)* algorithms, such as

ConvE (Dettmers et al., 2018), TuckER (Balažević et al., 2019), or 5*E (Nayyeri et al., 2021).

It is desirable to use both OIE and KBC approaches to automatically construct knowledge bases. However, automatic extractions from text yield uncanonicalized entities and relations. An entity such as "the United Kingdom" may also appear as "UK", and a relation such as "located at" may also appear as "can be found in". A knowledge base with possible repeated occurrences of entities and relations is called *Uncanonicalized* or *Open* Knowledge Base. The research area concerned with handling such knowledge bases is called *Open Knowledge Base Completion (OKBC)*. If we fail to connect these occurrences and treat them as distinct entities and relations, the performance of KBC algorithms drops significantly (Gupta et al., 2019). If our target data are canonicalized, collecting additional uncanonicalized data from unstructured text is therefore not guaranteed to improve the performance of said models. An illustration of a knowledge base can be found on Figure 2.2.

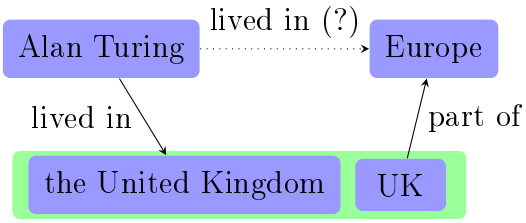



Figure 2.2: An example of a small knowledge base in which the fact whether Alan Turing lived in Europe is missing. If the knowledge base is canonicalized, "the United Kingdom" and "UK" are known to be the same entity. If the knowledge base is uncanonicalized or open, this information may not be given.

In this section, I first introduce the task of link prediction and its usual evaluation procedure and metrics. After this, I introduce the most commonly used datasets and models for knowledge base completion, the existing ways of dealing with open knowledge bases, and the existing approaches to using external data in knowledge base completion.

### 2.4.1 Evaluation of Knowledge Base Completion Models

The most common way to evaluate models for knowledge base completion is to evaluate them on the *link prediction* task. Let $\langle h, r, t \rangle$ be a test triplet, representing a true fact, unseen by the model. Given $\langle h, r, ? \rangle$, the model is used to rank all entities in the knowledge base from the most suitable to the least suitable tail, so that the entity no. 1 is the most likely tail according to the model. Ranks are usually reported in the *filtered setting*, which means that all other known correct answers, other than $t$, are removed from this list to reduce noise (Bordes et al., 2013). Let $r_t$ be the position or *rank* of the correct tail in this filtered list. The *reciprocal rank* of an example is then defined as $\frac{1}{r_t}$. The same process is repeated with the input $\langle ?, r, t \rangle$ and the head entity, treating $\langle ?, r, t \rangle$ and $\langle h, r, ? \rangle$ as two separate test instances.

The *mean rank (MR)* of the model is the average of all ranks across all test examples, both heads and tails. The *mean reciprocal rank (MRR)* is the average of all reciprocal ranks, both heads and tails. The *Hits@N* metric tells us how often the rank was smaller or equal to $N$. The related literature most commonly uses 1, 3, and 10 for values of $N$, however, 5, 30, and 50 are also occasionally reported.

In open knowledge bases, there may exist multiple equivalents of the missing entity in the knowledge base, i.e. synonyms. We say that such entities form a *cluster*. When evaluating a model, all entities in the cluster of the target entity are considered correct and $r_t$ is the best of their ranks. Not all Open Knowledge Base Completion datasets come with annotated clusters of entities, since annotations of large-scale datasets may be expensive. In such datasets, the possible existence of clusters is ignored and the model is evaluated as if the dataset was canonicalized.

### 2.4.2 Knowledge Base Completion Datasets

As described in this section, construction of good knowledge base completion datasets can be challenging. One of the assumptions in KBC is that all types of entities and relations are known in advance, also known as the *transductive setting*. Considering that the goal is to complete a knowledge base that is given in advance rather than handle newly added entities and relation types, this assumption is sensible. However, splitting the data in a knowledge base at random

to obtain a train, validation, and test split can lead to two potential problems. Firstly, entities in the validation and test set may not exist in the train set, violating the transductive hypothesis. Secondly, a random split can result in triviality of predictions, because many test instances can be obtained with simple inference patterns such as inversion of train instances.

In this section, the most common datasets for comparison of KBC and OKBC models are introduced. Several of them have been filtered to address the challenges of dataset construction, highlighted in the previous paragraph. Their statistics are given in Table 2.4):

| | OlpBench | ReVerb45k | ReVerb20k | Fb15k237 | Wn18rr |
|---|---|---|---|---|---|
| #entities | $2.4M$ | $27K$ | $11.1K$ | $14.5K$ | $41.1K$ |
| #relations | $961K$ | $21.6K$ | $11.1K$ | 237 | 11 |
| #entity clusters | N/A | $18.6K$ | $10.8K$ | N/A | N/A |
| #train triples | $30M$ | $36K$ | $15.5K$ | $272K$ | $86.8K$ |
| #valid triples | $10K$ | $3.6K$ | $1.6K$ | $17.5K$ | $3K$ |
| #test triples | $10K$ | $5.4K$ | $2.4K$ | $20.5K$ | $3K$ |

Table 2.4: Statistics of the introduced datasets. Only ReVerb45k and ReVerb20k come with gold entity clusters. Fb15k237 and Wn18rr are canonicalized, and OlpBench is too large to allow for a manual annotation of gold clusters.

**ReVerb45k and ReVerb20k** (Vashishth et al., 2018; Gupta et al., 2019), are small-scale OKBC datasets, obtained via the ReVerb OIE tool (Fader et al., 2011). These two datasets were adapted from knowledge base canonicalization datasets, coming with labelled gold clusters of equivalent entities (Galárraga et al., 2014). As described in the previous section, information on clusters of entities that refer to the same real-world entity can be used to improve the accuracy of the evaluation.

**OlpBench** (Broscheit et al., 2020) is a large-scale OKBC dataset automatically collected from the English Wikipedia. To avoid the problem of test leakage, the authors take multiple measures and filtering steps, providing training and validation sets with multiple levels of leakage removal and data quality. Throughout

all the experiments carried out in this work, only the highest-quality validation and train set are used and discussed. The train set with THOROUGH leakage removal ensures the removal of train triplets that are lexically too similar to a test triplet. More specifically, train triplets that can be obtained by either replacing entities with their synonyms or by changing the word order of a test triplet, are removed. Similarly, only the VALID-LINKED validation set is used in this thesis, since it contains only triplets with annotated entities.

**FB15K237 and WN18RR** (Toutanova et al., 2015; Dettmers et al., 2018) are subsets of the larger datasets FB15K and WN18, respectively. Both FB15K and WN18 were filtered to remove the test leakage through inverse relations and test triplets with entities that did not exist in the train set to obtain more difficult, but higher-quality datasets FB15K237 and WN18RR. FB15K237 was constructed from the most densely connected subset of the Freebase knowledge base (**?**), while WN18RR was obtained from the WordNet knowledge base (**?**). FB15K237, like many other datasets in this section, contains general knowledge about geography, sports, and celebrities. On the other hand, WN18RR contains linguistic information about words and relations between them. In the case of transfer learning, this creates a domain shift between pre-training and fine-tuning, potentially resulting in a worse overall performance.

### 2.4.3 Models for Knowledge Base Completion

In this section, I introduce all KBC models used in my experiments, chosen for their good performance across several KBC benchmarks. The high-level approach of the three models is similar. First, each entity and relation is assigned a vector in a low-dimensional vector space. These vectors are initialized randomly and trained jointly with the rest of the model. Head, relation, and tail vector embeddings are then used as the input to the model, which assigns the triplet a score, which should be high if the triplet is correct or low otherwise. Let us denote these vectors with $\mathbf{v}_h$, $\mathbf{v}_r$, and $\mathbf{v}_t$, respectively. The model can be either a neural network or a mathematical formula with desirable properties, as described in the following paragraphs.

I specifically do not include information on the performance of these models on benchmarks as their accuracy strongly depends on the training regime, choice of loss function, and other details, which are not consistent across their respective papers. Surveys of existing KBC methods have repeatedly shown that older models can outperform more recent approaches when trained in the same environment (Ruffinelli et al., 2020; Ali et al., 2020). Instead, I conduct an evaluation of these models in a controlled environment. It can be found in Chapter 5.

**TuckER** (Balažević et al., 2019) assigns a score to each triplet by multiplying the vectors with a core tensor $\mathcal{W} \in \mathbb{R}^{d_e \times d_e \times d_r}$, where $d_e$ is the dimension of entities, and $d_r$ is the dimension of relations. The parameters of the core tensor $\mathcal{W}$ are obtained through training. The model is prone to overfitting and a strong dropout rate of input features is crucial to train it well.

**ConvE** (Dettmers et al., 2018) assigns a score to each triplet by concatenating $\mathbf{v}_h$ and $\mathbf{v}_r$, reshaping them into a 2D matrix, and passing them through a convolutional neural network (CNN). The output of this CNN is a $d_e$-dimensional vector, which is multiplied with $\mathbf{v}_t$ and summed with a tail-specific bias term $b_t$ to obtain the score of the triplet. The CNN has a single layer and is preceded and followed by dropout and batch norm layers.

**5*E** (Nayyeri et al., 2021) model consider $\mathbf{v}_h$ and $\mathbf{v}_t$ to be complex projective lines and $\mathbf{v}_r$ a vector of $2 \times 2$ complex matrices. These correspond to a relation-specific Möbius transformation of projective lines. I refer the reader to the work of Nayyeri et al. (2021) for the details, which are here omitted for brevity. Unlike in ConvE and TuckER, there are no shared parameters between different relations and entities.

### 2.4.4 Existing Approaches to Open Knowledge Bases and Transfer of Knowledge

While large-scale transfer learning in KBC is not yet a widely-used method, multiple attempts to incorporate higher-level structures or external knowledge exist. The most common approach is to impose a hierarchy between entities or relations.

This information is already available inside the KB and many approaches just adapt the scoring function to raise its importance. *CTransR* (Huang et al., 2016) collects the input data into clusters and learns a separate relation embedding for each cluster. *Semantically Smooth Embedding (SSE)* (Guo et al., 2015) and *Type-embodied Knowledge Representation (TKRL)* (Xie et al., 2016b) assign types to entities and require the entities of the same type (determined through the `IsA` relation) to stay close together.

A different approach to incorporate additional information is to analyse paths. Unlike regular approaches which only analyse single relations (paths of length 1), models like *PTransE* (Lin et al., 2015) additionally incorporate information on the paths of longer lengths. The number of paths increases exponentially with the length, hence all such approaches resort to sampling and approximate methods. Incorporating paths improved the quality of embeddings at the expense of computational complexity. Thus, they do not appear in combination with more complex models.

The models described so far rely on the information provided by the knowledge base. However, a lot of knowledge is not incorporated in the knowledge bases directly. Various models try to increase the amount of available information by using human-written text. The simplest approach of incorporating natural text is initialization of parameters from precomputed word embeddings based on the textual description of the entities. This approach is used by the *NTN* model (Socher et al., 2013).

More advanced models, such as *DKRL* (Xie et al., 2016a) and *TEKE* (Wang and Li, 2016), train the entity embeddings jointly with the text model and try to align them. Wang and Li (2016) also define textual context embedding and try to learn the mapping between the textual and knowledge base embedding. Xie et al. (2016a), on the other hand, make use of an encoder that generates an embedding given a description of the entity. They show that their approach generalizes even to previously unseen entities when the description of these entities is available. Models that additionally incorporate natural text perform better when faced with previously unseen entities. Yao et al. (2019) make use of a large-scale pre-trained transformer model BERT to classify whether a fact is true. The main drawback

of such an approach is its speed – BERT is large and it can take weeks of GPU runtime to evaluate such a model on a benchmark test set.

Many of the ideas described in this section thus far have been later used to tackle open knowledge bases. Earlier attempts at open knowledge base completion are tied to existing work on the canonicalization of knowledge bases. To canonicalize open knowledge bases, automatic canonicalization tools cluster entities using manually defined features (Galárraga et al., 2014) or by finding additional information from external knowledge sources (Vashishth et al., 2018). Gupta et al. (2019) use clusters obtained with these tools to augment entity embeddings for KBC. Note that Gupta et al. (2019) use RNN-based encoders to encode relations, but not to encode entities. Broscheit et al. (2020), on the other hand, introduce a model with RNN-based encoders for both entities and relations, however, they compare their work on different datasets than Gupta et al. (2019). Finally, Chandrahas and Talukdar (2021) use both KB canonicalization tools and large-scale pre-trained model BERT, combining their predictions to make a more informed decision. The comparison of these approaches can be found in Chapter 5.

## 2.5 Bias in Natural Language Processing

Systems developed and trained on real-world data can contain pervasive biases and historical stereotypes that exist in the real world. The deployment of an unfair system can lead to harmful perpetuation of societal imbalances, resulting in systemic discrimination against certain groups of users. There exist numerous real-world examples where algorithmic bias led to discrimination with real-world consequences, such as unfairly predicting that black prisoners are more likely to re-offend, affecting the courtroom procedure,[10] or decreasing the accessibility of child welfare to communities of particular ethnic backgrounds (Chouldechova et al., 2018). This section provides the definitions of fairness and bias in natural language processing and machine learning and provides a fine-grained classification of relevant types of biases and their origins. This is followed by a discussion of the relevant datasets and approaches for bias detection.

[10] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

An argument can be made that unfair models are "bad for business" as they might drive away customers that are being discriminated against. However, such an argument ignores the situations described in the previous paragraph, where the creators and users of biased systems were not the ones being affected by the biases. Moreover, creating a system that treats a small ethnic group fairly might not return on the investment that is required to de-bias the existing system. We can thus safely say that in many of the situations, including the ones discussed later, the motivation for bias removal is largely political. The focus of this thesis is largely technical and I will hence avoid a detailed discussion on why digital products that reinforce systemic discrimination are problematic. A reader that is unconvinced about the importance of the problem can find a more detailed argumentation and further examples in the works of Baer (2019) and Altman (2020).

### 2.5.1 Defining Bias and Fairness

Unambiguously defining bias and fairness is non-trivial due to its tendency to manifest in diverse ways. As a result, numerous definitions of *fair* systems exist, many of them often contradicting each other (Kleinberg, 2018). Based on the task and situation, different definitions can apply depending on the situation.

The datasets and metrics employed in this work are covered by two definitions, which will be given and discussed below. A reader interested in alternative definitions can find them in the work of Mehrabi et al. (2019).

**Definition 1.** *Given a binary predictor $Y$ and a protected class $A$, $Y$ satisfies Statistical Parity if* $\mathbb{P}(Y|A = 0) = \mathbb{P}(Y|A = 1)$.

This definition is also known as *Demographical Parity* and is satisfied when the likelihood of the positive or correct outcome is the same regardless of whether the input example is an instance of a protected class. A more rigorous but harder metric to measure is *Conditional Statistical Parity*.

**Definition 2.** *Given a binary predictor $Y$, a set of legitimate factors $L$, and a protected class $A$, $Y$ satisfies Conditional Statistical Parity if* $\mathbb{P}(Y|L = 1, A = 0) = \mathbb{P}(Y|L = 1, A = 1)$.

A model is fair according to this definition if, whenever the *legitimate factors* are satisfied, the likelihood of the positive or correct outcome is the same regardless of whether the input example is an instance of a protected class or not.

These two definitions are conflicting when $\mathbb{1}_L$ and $\mathbb{1}_A$ are not independent random variables — that is when instances from the protected class are more or less likely to satisfy the legitimate factors. In this case, the Conditional Statistical Parity might be a more sensible definition, however, $L$ is often unknown and cannot always be controlled. As a consequence, most described metrics and benchmarks for bias detection aim to measure Statistical Parity, while taking different measures to decrease the correlation between $\mathbb{1}_L$ and $\mathbb{1}_A$.

#### 2.5.1.1 Types of Bias

Not all biased behaviour is caused by the same factors, nor does it have the same consequences. Different types of bias thus have to be measured and countered with different methods. The term *type of bias* has a potentially confusing double meaning, as it is also used to denote different types of protected groups, e.g. racial bias, gender bias, etc. To avoid confusion, I reserve the term *type of bias* exclusively to describe its source and not the protected group it affects.

Similarly as with the definition of fairness, numerous types of bias exist and not all can be tackled through technical means (one such instance being *funding bias* – when the research and development outcome is affected by the source of project funding). The relevant definitions of different types of bias are introduced below, and twenty more can be found in the work of Mehrabi et al. (2019).

**Definition 3.** *Historical Bias is bias that already exists in socio-technical issues in the world and is thus reflected in the sampled data.*

For example, a language model may be more likely to associate a nurse with women and a surgeon with men since historically, and therefore in most text samples, nurses are female and surgeons are male. Unless we explicitly look for counterexamples, any random sampling procedure is likely to contain this imbalance as this is reflective of the current or past state of the world.

**Definition 4.** *Representation Bias is a consequence of uneven data sampling from the population.*

A dataset of text snippets collected from English Wikipedia is likely to underrepresent women as 8 out of 9 biographies are about men. Any statistical analysis of such a dataset is unlikely to accurately reflect the real-world $\sim 50\%$ gender split, since masculine examples carry much more overall weight. A likely consequence of training on such a dataset is a model that performs better on examples of one gender than the other.

**Definition 5.** *Evaluation Bias is any type of bias that affects the evaluation of a model.*

Any type of bias that affects the evaluation of the model, usually imbalanced data in the test set, is an instance of evaluation bias. Suppose we are comparing two models for coreference resolution, one that is 70% accurate for both genders, and one that is 60% accurate for feminine pronouns and 75% accurate for masculine pronouns. If masculine and feminine pronouns are equally represented in the test data, the first model will be selected as the better of the two. If 80% of instances only contain masculine pronouns, the second model will seem better, despite its bias and worse overall performance on the actual population.

#### 2.5.1.2 Linguistic View of Gender

While the introduced methods in this work can be applied to any protected group, most experiments are focused on gender. Due to its strong presence in language, the impact of gender can be easy to observe and control. As a consequence, many resources are available for observation of the impact of gender on the performance of NLP systems, making it a good testbed for the development of bias-removal techniques. This section aims to introduce the linguistic background of gender in natural language and follows Ackerman (2019).

Gender in language exists in multiple different forms, which are not always consistent one with another. Moreover, detecting and controlling for some forms of gender might be easier than for others as will be visible from examples.

**Definition 6.** *Grammatical Gender consists of morphosyntactic features that affect the grammar of a language.*

Morphemes in languages can usually be placed in a category, such as *masculine, feminine, neuter,* etc. While the grammatical gender often implies the biosocial gender of an individual, this is not always the case. This phenomenon will be further discussed after the definition of biosocial gender.

In various indo-european languages, nouns are assigned a grammatical gender, often on a seemingly arbitrary basis. This can be easily seen from the fact that the same noun can be assigned a different gender in different languages, e.g. German for *table* is *der Tisch* (masc), while in Spanish it is *la mesa* (fem). Due to its strong presence in syntax, grammatical gender can usually be detected with simple rules and can serve as a useful proxy for measuring the impact of other types of gender on model behaviour. However, due to its inconsistent behaviour across languages, such approaches are not guaranteed to work on a multi-lingual level.

**Definition 7.** *Conceptual Gender is a broad term that covers any type of informal association between a gender and an entity, usually when a morpheme is strongly associated with one gender.*

In English, but also in many other languages, this can often be seen in the case of gender-imbalanced occupations and can be described as a consequence of historical bias. However, its very existence in the real world can affect the grammar and meaning of a sentence and it should not merely be dismissed as an undesired effect of biases. Asarina (2009), e.g. give an example from Russian, *Zubnoj (masc) vrach prishla (fem)*, meaning *(female) dentist came.* Since the word *vrach* (doctor) is so strongly associated with masculine gender, the adjective *zubnoj* (dental) assumes a masculine form, and the femininity of the dentist is only given away by the feminine form of the verb *prishla* (came). In English, the importance and existence of conceptual gender is more difficult to demonstrate due to the smaller presence of grammatical gender. Nevertheless, it does affect the meaning of a sentence, but the lines between reasonable assumptions and unjustified impact of bias can be blurred.

**Definition 8.** *Biosocial Gender is the internal gender of an individual.*

Note that this definition does not address the difference between sex and gender, nor does it address situations like *gender roles* and *gender identities*, which

all impact the biosocial gender of an individual. However, further exploration of these cases is the focus of psychology and biology, which is beyond the scope of this work. Instead, let us assume that a ground-truth biosocial gender of an individual always exists and is clearly determined at the time of writing of the text.

Biosocial gender is usually the type of gender that we have in mind when we say that a system is *gender-biased*. It is thus important to note that biosocial gender does not always match grammatical gender, e.g. the German word *das Mädchen* and Slovene word *deklič* both mean *girl*, but have *neuter* and *masculine* grammatical gender, respectively. Examples of such mismatches in English do exist, but are less common and natural, e.g. in the case of misgendering.

### 2.5.2 Bias-detection Datasets

It is easier to demonstrate that a system is biased in some way than to define a universal metric for comparison of different models. As introduced in previous sections, bias can exist in different forms and different systems may display it in different ways. This motivates the development of diverse datasets and metrics for bias detection, which are presented herein. All datasets in this chapter were designed to measure different types of gender bias that can affect the performance of a model for coreference resolution.

In all three of the described datasets, it is always assumed that a correct answer exists and can be unambiguously determined. Going back to the two introduced definitions of fairness, one can say that the legitimate factors $L$ are always satisfied and any difference in performance between the two compared subsets can be described as a consequence of bias. Of course, this is an oversimplification as we will see later in Chapter 4. Just because each example has an unambiguous answer, this does not imply that the results on subsets of the data are always directly comparable. Datasets for bias detection in other tasks exist as well, however, they are not relevant to this work.

**WinoGender.** WinoGender (Rudinger et al., 2018) is a coreference resolution dataset that follows the Wsc format and is aimed to measure historical gender bias. One of the two candidates is always an occupation, while the other

*The nurse notified the patient that **his** shift would be ending in an hour.*
*The patient: **False**, The nurse: **True***

Figure 2.3: An example from the WINOGENDER dataset (Rudinger et al., 2018). It is obvious from the context that the pronoun **his** refers to the nurse, despite nurses not being traditionally male. The gender of the pronoun does not affect its referent in any way and can be changed to the feminine or neutral form.

is a participant, and both are selected to be gender neutral. Examples intentionally contain occupations with strong imbalance in the gender ratio, as can be seen on Figure 2.3. The participant can be replaced with the neutral "someone" and three different pronouns (he/she/they) can be used. The aim of this dataset is to measure how the change of the pronoun gender affects the accuracy of the model.

From the point of view of bias, this dataset is designed to measure the impact of historical bias. From the point of view of gender in linguistics, the dataset measures the impact of (mis-)alignment of conceptual and grammatical gender. A model that is unable to overcome historical stereotypes is expected to perform much worse on anti-stereotypical examples than on stereotypical ones.

**WINOBIAS.** Similarly to the WINOGENDER dataset, WINOBIAS (Zhao et al., 2018) is a WSC-inspired dataset that measures gender bias in coreference resolution algorithms. Similarly to WINOGENDER, it contains instances of occupations with a high gender imbalance. It contains 3,160 examples of Winograd Schemas, equally split into validation and test sets. The test set examples are split into 2 types, where examples of type 1 are "harder" and should not be solvable using the analysis of co-occurrence, while type 2 examples are easier. Additionally, each of these subsets is split into anti-stereotypical and pro-stereotypical subsets, depending on whether the gender of the pronoun matches the most common gender in the occupation. The difference in performance between pro- and anti-stereotypical examples shows how biased a model is. Just like WINOGENDER, it measures the impact of historical bias and conceptual gender.

**GAP** Having introduced GAP as a dataset for coreference resolution in Section 2.2.1, this section only focuses on its use as a gender-bias-measuring tool. Exactly half of the examples in GAP are feminine and exactly half are masculine.

The naturally more prevalent masculine instances were downsampled on purpose to prevent the possible representation evaluation bias. Webster et al. (2018) define a bias measure as the ratio between the $F_1$-scores on the feminine and masculine subsets, $F_1^F \,/\, F_1^M$. An unbiased system is therefore expected to achieve a bias score around 1.

Unlike WinoGender and WinoBias, Gap consists of real-world examples and it is harder to pinpoint exactly what type of bias it detects. Manual inspection of the dataset does not reveal any adversarial examples where conceptual gender plays a significant role or goes against stereotypes. Any difference in model performance can thus likely be attributed to representation bias in the training data, which matches the motivation and justifications from the authors of the dataset (Webster et al., 2018). Since the examples are not created from templates, feminine and masculine examples are not guaranteed to come from exactly the same distribution. This is an important observation, which I further address in Chapter 4.

The motivation by Webster et al. (2018) is focused on biosocial gender, i.e. comparing the performance of models when the candidates are male and when they are female. However, in practice, the dataset measures the impact of grammatical gender, as the authors define the gender of an example to match the gender of the pronoun in question. While these two types of gender largely overlap in English, mismatches can happen, e.g., in the case of *personalization* or *misgendering*. I found examples containing such a mismatch not to have a strong presence in the Gap test set, however, the exact number is hard to estimate due to the lack of context in many of the examples. I highlight that addressing this is necessary before the dataset-creating approach by Webster et al. (2018) can be used on languages with a stronger presence of grammatical gender, e.g., Russian and German.

# Chapter 3

# Pre-training for Coreference Resolution

In this chapter, I introduce an unsupervised pre-training method specific to coreference resolution. In the first part of the chapter, I introduce the pre-training method and how it can be used together with the BERT model to tackle Winograd Schema Challenge. In the second part of the chapter, I show that the method generalizes beyond the Winograd Schema Challenge and can help with snippet-level coreference resolution in general. This chapter is based on papers (Kocijan et al., 2019b) and (Kocijan et al., 2019a).

## 3.1 Using Pre-trained Language Models for the Winograd Schema Challenge

As introduced in Section 2.2.2, WSC273 consists of 273 instances of the pronoun disambiguation problem. Each is a sentence (or two) with a pronoun referring to one of the two or more nouns; the goal is to predict the correct one. The task is challenging, since WSC examples are constructed to require human-like commonsense knowledge and reasoning. The best known solutions at the time of this research used deep learning, achieving an accuracy of 63.7% (Opitz and Frank, 2018; Trinh and Le, 2018). The problem is difficult to solve not only because of the commonsense reasoning challenge, but also due to the small existing datasets making it difficult to train neural networks directly on the task.

A common approach to transfer learning in NLP is to train a language model (LM) on large amounts of unsupervised text (Howard and Ruder, 2018) and use it, with or without further fine-tuning, to solve other downstream tasks. Building on top of a LM has proven to be very successful, producing state-of-the-art results (Devlin et al., 2019) on benchmark datasets like GLUE (Wang et al., 2018).

In this chapter, I first show that fine-tuning existing LMs on DPR is a robust method for improving the capabilities of the LM to tackle WSC273 and WNLI. This is surprising, because previous attempts to generalize only from the DPR dataset to WSC273 did not achieve a major improvement (Opitz and Frank, 2018). Secondly, I introduce a method for generating large-scale WSC-like examples. I use this method to create a 2.4M dataset from English Wikipedia,[1] which I further use together with DPR for fine-tuning the pre-trained BERT LM (Devlin et al., 2019). The dataset and the code of this have been made publicly available.[2] The best model achieves an accuracy of 72.5% and 74.7% on WSC273 and WNLI, improving the previous best solutions by 8.8% and 9.6%, respectively.

### 3.1.1 Winograd Schema Challenge Approach

I approach WSC by fine-tuning the pre-trained BERT LM (Devlin et al., 2019) on the DPR training set and further on a very large Winograd-like dataset introduced in this work. The DPR dataset, as introduced in Section 2.2.2, is split into a training set with 1322 examples and a test set with 564 examples. Six examples in the DPR training set reappear in WSC273 in a similar form. To avoid test set contamination, these examples were removed from DPR. The DPR training and test sets are used for fine-tuning the LMs and for validation, respectively.

Given a training sentence $\mathbf{s}$, the pronoun to be resolved is masked out from the sentence and the LM is used to predict the correct candidate in the place of the masked pronoun. Let $c_1$ and $c_2$ be the two candidates. BERT for Masked Token Prediction is used to find $\mathbb{P}(c_1|\mathbf{s})$ and $\mathbb{P}(c_2|\mathbf{s})$. If a candidate consists of several tokens, the corresponding number of `[MASK]` tokens is used in the masked

[1] `https://dumps.wikimedia.org/enwiki/` dump id: enwiki-20181201

[2] The code can be found at `https://github.com/vid-koci/bert-commonsense`.
The dataset and the models can be obtained from `https://ora.ox.ac.uk/objects/uuid:9b34602b-c982-4b49-b4f4-6555b5a82c3d`

sentence. Then, $\log \mathbb{P}(c|\mathbf{s})$ is computed as the average of log-probabilities of each composing token. If $c_1$ is correct and $c_2$ is not, the loss is:

$$L = -\log \mathbb{P}(c_1|\mathbf{s}) + \alpha \cdot \max(0, \log \mathbb{P}(c_2|\mathbf{s}) - \log \mathbb{P}(c_1|\mathbf{s}) + \beta), \tag{3.1}$$

where $\alpha$ and $\beta$ are hyperparameters. The loss term is a combination of negative log likelihood loss and max-margin loss. While using the max margin loss alone already gives good results, adding a negative log likelihood was empirically found to additionally improve results. The maximisation of $\mathbb{P}(c_1|\mathbf{s})$ continues even after $\log \mathbb{P}(c_1|\mathbf{s}) - \log \mathbb{P}(c_2|\mathbf{s}) > \beta$, but at a significantly smaller rate, as $\alpha$ was always larger than 1 for the best hyperparameter setups.

**WNLI Approach.** Models are additionally tested on the test set of the WNLI dataset. To use the same evaluation approach as for the WSC273 dataset, we can transform the examples in WNLI from the premise–hypothesis format into the masked words format. Since each hypothesis is just a substring of the premise with the pronoun replaced for the candidate, finding the replaced pronoun and one candidate can be done by finding the hypothesis as a substring of the premise. All other nouns in the sentence are treated as alternative candidates. The Stanford POS-tagger (Manning et al., 2014) is used to find the nouns in the sentence. The probability for each candidate is computed to determine whether the candidate in the hypothesis is the best match. Only the test set of the WNLI dataset is used, because it does not overlap with WSC273. I do not train or validate on the WNLI training and validation sets, because some of the examples share the premise. Indeed, when the above-mentioned rephrasing of the examples is used, the training, validation, and test sets start to overlap.

### 3.1.2 MaskedWiki Dataset

To get more data for fine-tuning, I automatically generate a large-scale collection of sentences similar to WSC. More specifically, the procedure searches a large text corpus for sentences that contain at least two occurrences of the same noun. The second occurrence of this noun is masked with the `[MASK]` token. Several possible replacements for the masked token are given, one for each non-masked noun in

the sentence. The obtained examples that are thus structurally similar to those in WSC, although it is not ensured that they fulfill all the requirements (see Section 2.2.2).

To generate such sentences, I choose English Wikipedia as the source text corpus, since it is a large-scale and grammatically correct collection of text with diverse information. I use the Stanford POS tagger (Manning et al., 2014) to find nouns. In this way, one can obtain a dataset with approximately 130M examples. I chose to downsample the dataset to $2.4M$ examples uniformly at random to obtain a dataset of manageable size. All experiments are conducted with this downsampled dataset only to which we refer as MaskedWiki throughout this work.

To determine the quality of the dataset, 200 random examples are manually categorized into 4 categories:

- Unsolvable: the masked word cannot be unambiguously selected with the given context. Example: *Palmer and Crenshaw both used Wilson 8802 putters , with [MASK] 's receiving the moniker " Little Ben " due to his proficiency with it . [Palmer/Crenshaw]*

- Hard: the answer is not trivial to figure out. Example: *At the time of Plath 's suicide , Assia was pregnant with Hughes 's child , but she had an abortion soon after [MASK] 's death . [Plath/Assia]*

- Easy: The alternative sentence is grammatically incorrect or is very visibly an inferior choice. Example: *The syllables are pronounced strongly by Gaga in syncopation while her vibrato complemented Bennett's characteristic jazz vocals and swing . Olivier added , " [MASK] 's voice , when stripped of its bells and whistles, showcases a timelessness that lends itself well to the genre . " [Gaga/syncopation]*

- Noise: The example is a result of a parsing error.

In the analyzed subset, 8.5% of examples were unsolvable, 45% were hard, 45.5% were easy, and 1% fell into the noise category.

| | WSC273 | WNLI |
|---|---|---|
| BERT_WIKI | 0.619 | 0.712 |
| BERT_WIKI_DPR | **0.725** | **0.747** |
| BERT | 0.619 | 0.658 |
| BERT_DPR | **0.714** | **0.719** |
| BERT-base | 0.564 | 0.630 |
| BERT-base_DPR | **0.623** | **0.705** |
| GPT | 0.553 | – |
| GPT_DPR | **0.674** | – |
| BERT_WIKI_$\text{DPR}_{\text{no_pairs}}$ | 0.663 | – |
| BERT_WIKI_$\text{DPR}_{\text{pairs}}$ | **0.703** | – |
| LM ensemble | 0.637 | – |
| Knowledge Hunter | 0.571 | – |

Table 3.1: Results on WSC273 and WNLI. The comparison between each language model and its DPR-tuned model is given. For each column, the better result of the two is in bold. The best result in the column overall is underlined. Results for the LM ensemble and Knowledge Hunter are taken from Trichelair et al. (2018). All models consistently improve their accuracy when fine-tuned on the DPR dataset.

### 3.1.3 Evaluation

In this work, the PyTorch implementation[3] of Devlin et al.'s (2019) pre-trained model, BERT-large, is employed. To obtain BERT_WIKI, the model is trained on MASKEDWIKI starting from the pre-trained BERT. The training procedure differs from the training of BERT (Devlin et al., 2019) in a few points. The model is trained with a single epoch of the MASKEDWIKI dataset, using batches of size 64 (distributed on 8 GPUs), Adam optimizer, a learning rate of $5.0 \cdot 10^{-6}$, and hyperparameter values $\alpha = 20$ and $\beta = 0.2$ in the loss function (Eq. (3.1)) are used. The values were selected from $\alpha \in \{5, 10, 20\}$ and $\beta \in \{0.1, 0.2, 0.4\}$ and learning rate from $\{3 \cdot 10^{-5}, 1 \cdot 10^{-5}, 5 \cdot 10^{-6}, 3 \cdot 10^{-6}\}$ using grid search. To speed up the hyperparameter search, the training (for hyperparameter search only) is done on a randomly selected subset of size $100,000$. The performance is then compared on the DPR test set.

Both BERT and BERT_WIKI are fine-tuned on the DPR training dataset to create BERT_DPR and BERT_WIKI_DPR.

[3] https://github.com/huggingface/pytorch-pretrained-BERT

| | non-assoc. | assoc. | unswitched | switched | consist. |
|---|---|---|---|---|---|
| BERT_WIKI | 0.597 | 0.757 | 0.573 | 0.603 | 0.389 |
| BERT_WIKI_DPR | **0.720** | **0.757** | **0.732** | **0.710** | **0.550** |
| BERT | 0.602 | 0.730 | 0.595 | 0.573 | 0.458 |
| BERT_DPR | **0.699** | **0.811** | **0.695** | **0.702** | **0.550** |
| BERT-base | 0.551 | 0.649 | 0.527 | 0.565 | 0.443 |
| BERT-base_DPR | **0.606** | **0.730** | **0.611** | **0.634** | **0.443** |
| GPT | 0.525 | 0.730 | 0.595 | 0.519 | 0.466 |
| GPT_DPR | **0.653** | **0.811** | **0.664** | **0.580** | **0.641** |
| BERT_WIKI_DPR$_{\text{no_pairs}}$ | 0.669 | 0.622 | 0.672 | 0.641 | 0.511 |
| BERT_WIKI_DPR$_{\text{pairs}}$ | **0.695** | **0.757** | **0.718** | **0.710** | **0.565** |
| LM ensemble | 0.606 | **0.838** | 0.634 | 0.534 | 0.443 |
| Knowledge Hunter | 0.583 | 0.5 | 0.588 | 0.588 | **0.901** |

Table 3.2: Results on WSC273 subsets. The comparison between each language model and its DPR-tuned model is given. For each column, the better result of the two is in bold. The best result in the column overall is underlined. Results for the LM ensemble and Knowledge Hunter are taken from Trichelair et al. (2018). Consistency is defined as the fraction of predictions that remain the same even when the two switchable candidates are swapped. All models consistently improve their accuracy when fine-tuned on the DPR dataset but LM-based approaches still achieve low overall consistency.

The DPR test set was used as the validation set. The fine-tuning procedure was the same as the training procedure on MASKEDWIKI, except that 30 epochs were used. The model was validated after every epoch, and the model with highest performance on the validation set was retained. The hyperparameters $\alpha$ and $\beta$ and the learning rate were selected with grid search from the same sets as for MASKEDWIKI training.

For comparison, experiments were also conducted on two other LMs, BERT-base (BERT with fewer parameters) and General Pre-trained Transformer (GPT) by Radford et al. (2018). The training on the BERT-base was conducted in the same way as for the other models. When using GPT, the probability of a word belonging to the sentence $\mathbb{P}(c|\mathbf{s})$ is computed by predicting the probability of the sequence following $\mathbf{s}$, conditioned on $\mathbf{s}$ and the words before it. This was called *partial loss* by Trinh and Le (2018).

Due to WSC's "special word" property, examples come in pairs. A pair of

examples only differs in a single word (but the correct answers are different). The model BERT_WIKI_DPR$_{\text{no_pairs}}$ is the BERT_WIKI model, fine-tuned on DPR, where only a single example from each pair is retained. The size of DPR is thus halved. The model BERT_WIKI_DPR$_{\text{pairs}}$ is obtained by fine-tuning BERT_WIKI on half of the DPR dataset. This time, all examples in the subset come in pairs, just like in the unreduced DPR dataset.

All models are evaluated on WSC273 and the WNLI test dataset, as well as the various subsets of WSC273. The results are reported in Tables 3.1 and 3.2.

**Discussion** Firstly, models fine-tuned on the DPR dataset consistently outperform their non-fine-tuned counterparts. The BERT_WIKI_DPR model outperforms other language models on 5 out of 6 sets that they are compared on. In comparison to the LM ensemble by Trinh and Le (2018), the accuracy is more consistent between associative and non-associative subsets and less affected by the switched parties. However, it remains fairly inconsistent, which is a general property of LMs.

Secondly, the results of BERT_WIKI seem to indicate that this dataset alone does not help BERT. However, when additionally fine-tuned to DPR, the accuracy consistently improves.

Finally, the results of BERT_WIKI$_{\text{no_pairs}}$ and BERT_WIKI$_{\text{pairs}}$ show that the existence of WSC-like pairs in the training data affects the performance of the trained model. MASKEDWIKI does not contain such pairs.

### 3.1.4 Summary and Impact

The introduced models achieve new state-of-the-art results on the WSC273 and WNLI datasets by fine-tuning the BERT language model on the DPR dataset and a newly introduced MASKEDWIKI dataset. The previous state-of-the-art results on WSC273 and WNLI are improved by 8.8% and 9.6%, respectively. Moreover, these were the first models to beat the majority baseline on WNLI.

Evidently, fine-tuning on WSC-like data consistently improves the fine-tuned language model's performance on WSC. The consistent improvement of several

language models indicates the robustness of this method. This is particularly surprising, because previous work (Opitz and Frank, 2018) implies that generalizing to WSC273 is difficult.

This work had substantial impact on the development of WSC solutions. He et al. (2019) and Ye et al. (2019) directly incorporated the introduced BERT model into their own approaches to WSC, augmenting it with an additional objective or with additional commonsense-reasoning pre-training, respectively. Sakaguchi et al. (2020) used a similar approach with the RoBERTa LM and WinoGrande dataset to achieve 90% accuracy on WSC273, showing that the introduced method can be used to solve the WSC – it simply required a stronger LM and more training data.

The introduced re-phrasing approach to WNLI became the standard way of testing LMs on it as part of the GLUE benchmark (Wang et al., 2018). Before that, models often achieved worse performance than the majority class baseline (Radford et al., 2018) or simply chose to skip the task (Devlin et al., 2019). By using the introduced reformulation, Liu et al. (2019b) achieved the first better-than-human performance on GLUE. It became one of *standard tricks* and was used by all best-scoring models on the GLUE benchmark (Liu et al., 2019c,b; Raffel et al., 2020; Yang et al., 2019). By scaling up the models and pre-training, using ensembles and additional training data, many of these models were able to obtain scores over 90% on WNLI.

Within months of this research work, the previously unsolvable Winograd Schema Challenge was solved using large pre-trained language models in the way described above. To some, this was a display of the power of pre-trained transformers, to others, a disappointing realization that the challenge was not *as difficult* as it had seemed. New tasks, not yet solvable with pre-trained language models, are being proposed, however, none of them have yet captured the public attention with their elegance and simplicity to the same extent.

## 3.2 Unsupervised Pre-training for Snippet-Level Co-reference Resolution

In this section, I revisit the method used to construct MaskedWiki in the previous section with the aim to improve the quality of the generated samples. Instead of masking noun phrases, I find passages of text where a personal name appears at least twice and mask one of its non-first occurrences. To make the disambiguation task more challenging, I also ensure that at least one other distinct personal name is present in the text in a position before the masked occurrence. I instantiate the method on English Wikipedia and generate the Wikipedia Co-REferences Masked (WikiCREM) dataset with 2.4M examples, which I make publicly available for further usage.[4]

Masking personal names instead of noun phrases results in higher sample quality. This will later be demonstrated both through qualitative and quantitative observations. Manual annotation shows that masking noun phrases is more likely to result in trivial or unnatural examples. Moreover, training on MaskedWiki without additional fine-tuning does not improve performance on the Wsc273 dataset, while training on WikiCREM does.

I show the value of WikiCREM by using it to fine-tune the Bert language model (Devlin et al., 2019), as introduced in the previous section. The models trained on this dataset are then tested on a wider collection of datasets to show the usefulness of the method beyond the Winograd Schema Challenge.

### 3.2.1 The WikiCREM Dataset

In this section, I describe how I obtained WikiCREM, highlighting differences with MaskedWiki. Starting from English Wikipedia,[5] I search for sentences and pairs of consecutive sentences with the following properties: at least two distinct personal names appear in the text, and one of them is repeated. This is a stricter condition than in the case of MaskedWiki, where any noun phrase is accepted.

[4] The code can be found at `https://github.com/vid-koci/bert-commonsense`.
The dataset and the models can be obtained from `https://ora.ox.ac.uk/objects/uuid:c83e94bb-7584-41a1-aef9-85b0e764d9e3`

[5] `https://dumps.wikimedia.org/enwiki/` dump id: enwiki-20181201

I do not use pieces of text with more than two sentences in order to collect concise examples only. Personal names in the text are called "candidates". One non-first occurrence of the repeated candidate is masked, and the goal is to predict the masked name given the correct and one incorrect candidate. In case of more than one incorrect candidate in the sentence, several datapoints are constructed, one for each incorrect candidate.

I ensure that the alternative candidate appears before the masked-out name in the text to avoid trivial examples. Thus, the example is retained in the dataset if:

1. the repeated name appears after both candidates, all in a single sentence; or
2. both candidates appear in a single sentence, and the repeated name appears in a sentence directly following.

Examples where one of the candidates appears in the same sentence as the repeated name, while the other candidate does not, are discarded as they are often too trivial.

I illustrate the procedure with the following example:

*In the reception room, a boy named **Billy** won't stop staring at **Don**. **Don** is drawing a picture and then rips it out of his book and hands it to **Billy**, getting up and leaving.*

Either second occurrence of "Billy" or "Don" can be masked. Let us choose "Don", to obtain the following example:

*In the reception room, a boy named **Billy** won't stop staring at **Don**. `[MASK]` is drawing a picture and then rips it out of his book and hands it to **Billy**, getting up and leaving.*

**Candidates:** Billy, **Don**

The goal is to determine which of the two candidates ("Billy","Don") has been masked out. The masking process resembles replacing a name with a pronoun, but the pronoun is not inserted to keep the process fully unsupervised and error-free.

I used the Spacy Named Entity Recognition (NER) library[6] to find the occurrences of names in the text. The resulting dataset consists of $2,438,897$ samples. $10,000$ examples are held out to serve as the validation set. Two more examples from the dataset can be found on Figure 3.1.

[6] `https://spacy.io/usage/linguistic-features#named-entities`

*Gina arrives and she is furious with Denise for not protecting Jody from Kingsley, as `[MASK]` was meant to be the parent.*
**Candidates:** Gina, **Denise**
*When Ashley falls pregnant with Victor's child, Nikki is diagnosed with cancer, causing Victor to leave `[MASK]`, who secretly has an abortion.*
**Candidates: Ashley**, Nikki

Figure 3.1: WikiCREM examples. Correct answers are given in bold.

Note that WikiCREM contains several hard examples. To resolve the first example, one needs to understand that Denise was assigned a task and the phrase "meant to be the parent" thus refers to her. To resolve the second example, one needs to understand that having an abortion can only happen if one falls pregnant first. Since both candidates have feminine names, the answer cannot be deduced just on the common co-occurrence of female names and the word "abortion". The example generation method, while having the advantage of being unsupervised, also does not give incorrect signals, since we know the ground-truth reference.

**WikiCREM statistics.** I analyze the dataset for gender balance by using the Gender guesser library[7] to determine the gender of the candidates. To mimic the analysis of pronoun genders performed in related works (Webster et al., 2018; Rudinger et al., 2018; Zhao et al., 2018), the gender of the correct candidates only is observed. There were 0.8M "male" or "mostly_male" names and 0.42M "female" or "mostly_female" names, the rest were classified as "unknown". The ratio between female and male candidates is thus estimated around 0.53 in favour of male candidates. This gender imbalance does not seem to have any negative impact on bias, as shown in Chapter 4.

However, this unsupervised generating procedure sometimes yields examples where the correct answer cannot be deduced given the available information; let us refer to these as *unsolvable examples*. To estimate the percentage of unsolvable examples, myself and other collaborators on this project manually annotated 100 randomly selected examples from the WikiCREM dataset. In order to prevent guessing, the candidates were not visible to the annotator. For each example, the

[7] https://pypi.org/project/gender-guesser/

annotator stated whether it was solvable or not, and attempted to answer the solvable examples. In 100 examples, 18 unsolvable examples were found, while achieving 95.1% accuracy on the rest, showing that the annotation error rate is tolerable. These annotations can be found in Appendix A.

However, as shown in Section 3.2.2.3, training on WikiCREM alone does not match the performance of using the training on the data from the target distribution. The data distribution of WikiCREM differs from the data distribution of the evaluation datasets. If we replace the `[MASK]` token with a pronoun instead of the correct candidate, the resulting sentence sometimes sounds unnatural and would not occur in human-written text. On the annotated 100 examples, the percentage of natural-sounding sentences is estimated to be 63%. While other 37% sentences are not incorrect, their distribution differs from the target data.

**Comparison of MaskedWiki and WikiCREM** As seen in Section 3.1.3, training on MaskedWiki on its own is not always enough and sometimes makes a difference only in combination with additional training on the Dpr dataset. In contrast, WikiCREM brings a much more consistent improvement over a wider range of datasets, strongly improving models' performance even when they are not fine-tuned on additional data. Before the experimental analysis, I conduct a manual investigation into the quality of WikiCREM and MaskedWiki to show a significant difference in the quality of the examples.

I and other collaborators on this project annotated 100 random examples from MaskedWiki in the same way as we did for WikiCREM. In MaskedWiki, we looked for examples where masked nouns can be replaced with a pronoun. Only in 7 examples did we obtain a natural-sounding and grammatically correct sentence. This is in stark contrast with WikiCREM, where we estimated that 63% of the annotated examples form a natural-sounding sentence when the appropriate pronoun is inserted, showing that WikiCREM consists of examples that are much closer to the target data. I again highlight that pronouns are not actually inserted into the sentences when the dataset is used in practice. The trained model is thus never fed any unnaturally-sounding examples. This analysis was performed to show that WikiCREM consists of examples with a data distribution closer to the target tasks than MaskedWiki.

### 3.2.2 Evaluation

To quantify the impact of WIKICREM, I introduce multiple different models and evaluate them on several datasets. The training setup and hyperparameter search exactly matches the one described in Sections 3.1.1 and 3.1.3, and will not be repeated for brevity.

#### 3.2.2.1 Evaluation Datasets

Seven different datasets are used to evaluate the performance of different models: GAP, WSC273, PDP, DPR, WNLI, WINOGENDER, and WINOBIAS. Their detailed description can be found in Sections 2.2.2.1 and 2.4.2. In this section, I will only describe the model-specific pre-processing steps that were taken with each of them.

**GAP.** This dataset additionally comes with a training and validation set. In addition to the overall performance on the test set, each model is also evaluated on its performance on the masculine subset ($F_1^M$) and feminine subset ($F_1^F$), as encouraged by Webster et al. (2018). The bias score, however, is not reported here. As later discussed in Chapter 4, the score is flawed and a better alternative has to be proposed first. The best performance at the time of this project was exhibited by the Referential Reader (Liu et al., 2019a), a GRU-based model with additional external memory cells.[8]

For each example, two candidates are given with the goal of determining whether they are the referent. In approximately 10% of the training examples, none of the candidates are correct. When training on the GAP dataset, I discard such examples from the training set. No examples were discarded from the validation or test sets.

When testing the model, the Spacy NER library is used to find all candidates in the sentence. Since the GAP dataset mainly contains examples with human names, I only retain named entities with the tag PERSON. I observe that in 18.5% of the test samples, the Spacy NER library fails to extract the candidate in question, making the answer for that candidate "FALSE" by default. This puts

[8]The better performing BERT-based version of the Referential Reader was added to the cited paper only later on.

the evaluated models at disadvantage when compared to external work. For this reason 7.25% of answers are always false negatives and 11.25% are always true negatives, regardless of the model. Taking this into account, the maximal $F_1$-score achievable by the evaluated models is capped at 91.1%.

I highlight that, when evaluating our models, this approach is stricter than previous ones (Liu et al., 2019a; Webster et al., 2018). While they count the answer as "correct" if the model returns a substring or overlap with the correct answer, here only the full answer is accepted. The aforementioned models return the exact location of the correct candidate in the input sentence, while my approach does not. This strictness is necessary because a substring of a correct answer could be a substring of several answers at once, making it ambiguous.

Even though WikiCREM and Gap both use text from English Wikipedia, they produce differing examples because their gathering processes differ. In Gap, passages with pronouns are collected and the pronouns are manually annotated, while WikiCREM is generated by masking names that appear in the text. Even if the same text is used, different masking process will result in different inputs and outputs, making the examples different. Moreover, the WikiCREM collection method does explicitly or implicitly resolve any of the pronouns in the text, making any similarity between the samples acceptable under the transductive hypothesis.

**Dpr.** This dataset comes with a training and a test set. I remove 6 examples in the Dpr training set that overlap with the Wsc dataset. The dataset consists of 1316 training and 564 test samples after the removal of overlapping examples. Unlike in Section 3.1.1, where the test set was used for validation of the model, I choose to use it as a test set on its own. To this end, I hold out 10% of the Dpr training set (131 examples) to use them as the validation set instead.

The best result on this dataset at the time was reported by Peng et al. (2015) using external knowledge sources and integer linear programming.

**WinoGender and WinoBias.** Despite being bias-measuring datasets, these two datasets are employed only to measure performance of the models. Both datasets detect bias by observing the change in prediction after the grammatical

gender of the pronoun is changed. However, in this work, the pronoun is masked-out during the pre-processing steps, which makes the model agnostic to this change. The bias-measuring component of these datasets thus cannot be utilized faithfully.

According to Rudinger et al. (2018), the best performance on WinoGender is exhibited by Durrett and Klein (2013) when used on the male subset of the dataset. This result is thus used as the baseline.

Results on WinoBias are reported for each of its subsets, split into *Type-1* (harder) and *Type-2* (easier) subsets, as introduced in Section 2.4.2, and further split depending on whether the examples are pro- or anti-stereotypical. The best performance on WinoBias is exhibited by Lee et al. (2017) and Durrett and Klein (2013), as reported by Zhao et al. (2018).

**Wnli, Wsc273, and Pdp.** These three datasets are used as test sets only and are approached in the same way as in Section 3.1.1. Pdp is treated the same way as Wsc273. Results from Section 3.1.3 are used for comparison on Wsc273 and Wnli, as they were the best reported results on those datasets at the time. The best result on Pdp was reported by Fähndrich et al. (2018).

#### 3.2.2.2 Experiments

Several different models are trained to evaluate the contribution of the Wiki-CREM dataset in different real-world scenarios. In **Scenario A**, no information of the target distribution is available. In **Scenario B**, the distribution of the target data is known and a sample of training data from the target distribution is available. Finally, **Scenario C** is the transductive scenario, where the unlabeled test samples are known in advance. All evaluations on the Gap test-set are considered to be Scenario C, because Bert has been pre-trained on the English Wikipedia and has thus seen the text in the Gap dataset at the pre-training time.

The evaluated models are described below.

**Bert.** This model, pretrained by Devlin et al. (2019), is the starting point for all models and serves as the soft baseline for Scenario A.

**BERT_WIKIRAND.** This model serves as an additional baseline for Scenario A and aims to eliminate external factors that might have worked against the performance of BERT. To eliminate the effect of sentence lengths, loss function, and the percentage of masked tokens during the training time, the RANDOMWIKI dataset has been generated. It consists of random passages from Wikipedia and has the same sentence-length distribution and number of datapoints as WIKICREM. However, the masked-out word from the sentence is selected randomly, while the alternative candidate is selected randomly from the vocabulary. BERT is then trained on this dataset in the same way as BERT_WIKICREM.

**BERT_WIKICREM.** BERT, additionally trained on WIKICREM. Its evaluation on non-GAP datasets serves as the evaluation of WIKICREM under Scenario A.

**BERT_DPR.** BERT, fine-tuned on 90% of the DPR training dataset and validated on the remaining 10% of samples. All datasets, other than GAP, were inspired by the Winograd Schema Challenge and come from a similar distribution. This model is used as the baseline for Scenario B. Note that this is **not** equivalent to BERT_DPR from Section 3.1.3 as that model was trained on the entire DPR training set and validated on the DPR test set.

**BERT_WIKICREM_DPR.** This model is obtained by fine-tuning BERT_WIKICREM on DPR using the same split as for BERT_DPR. It serves for the evaluation of WIKICREM under Scenario B.

**BERT_GAP_DPR.** This model serves for an additional comparison to the BERT_WIKICREM_DPR model. It is obtained by fine-tuning BERT_GAP on the DPR dataset.

**BERT_GAP.** This model is obtained by fine-tuning BERT on the GAP dataset. It serves as the baseline for Scenario C.

| | GAP $F_1$ | GAP $F_1^F$ | GAP $F_1^M$ |
|---|---|---|---|
| SOTA | 72.1% | 71.4% | 72.8% |
| BERT | 50.0% | 47.2% | 52.7% |
| BERT_WIKIRAND | 55.1% | 51.8% | 58.2% |
| BERT_WIKICREM | **59.0**% | **57.5**% | **60.5**% |
| BERT_GAP | 75.2% | 75.1% | 75.3% |
| BERT_WIKICREM_GAP | **77.4**% | **78.4**% | **76.4**% |
| BERT_DPR | 60.9% | 61.3% | 60.6% |
| BERT_GAP_DPR | **70.0**% | **70.4**% | **69.5**% |
| BERT_WIKICREM_DPR | 64.2% | 64.2% | 64.1% |
| BERT_ALL | 76.0% | 77.4% | 74.7% |
| BERT_WIKICREM_ALL | **78.0**% | **79.4**% | **76.7**% |

Table 3.3: Evaluation of trained models on the GAP test set and its subsets. All results in this table fall under Scenario C, since BERT was trained on English Wikipedia, and then used to construct GAP. The table is further split into sections separated with horizontal lines. Each section contains a model that has been trained on WIKICREM and models that have not been. The best result in each section is highlighted in bold. The best overall result is underlined. Scores on GAP are measured as $F_1$-score. The source of each state-of-the-art result (SOTA) is listed in Section 3.2.2.1.

**BERT_WIKICREM_GAP.** This model serves for the evaluation of WIKICREM for Scenario C and is obtained by fine-tuning BERT_WIKICREM on GAP.

**BERT_ALL.** This model is obtained by fine-tuning BERT on all available data from all the target datasets at once. Combined GAP-train and DPR-train data are used for training. The model is validated on the GAP-validation set and the WINOBIAS-validation set separately. Scores on both sets are then averaged to obtain the validation performance. Since both training sets and both validation sets have roughly the same size, both tasks are represented equally.

**BERT_WIKICREM_ALL.** This model is obtained in the same way as the BERT_ALL model, but starting from BERT_WIKICREM instead.

| | Type-1-anti | Type-1-pro | Type-2-anti | Type-2-pro |
|---|---|---|---|---|
| SOTA | 60.6% | 74.9% | 78.0% | 88.6% |
| Bert | 61.3% | 60.3% | 76.2% | 75.8% |
| Bert_WikiRand | 53.5% | 52.5% | 64.6% | 65.2% |
| Bert_WikiCREM | **65.2%** | **64.9%** | **95.7%** | **94.9%** |
| Bert_Gap | 64.6% | 63.8% | 88.1% | 87.9% |
| Bert_WikiCREM_Gap | **71.2%** | **70.5%** | **97.2%** | **98.2%** |
| Bert_Dpr | **78.0%** | **78.2%** | 85.6% | 86.4% |
| Bert_Gap_Dpr | 77.8% | 76.5% | **89.6%** | **89.1%** |
| Bert_WikiCREM_Dpr | 76.0% | 76.3% | 81.3% | 80.3% |
| Bert_all | **77.8%** | **77.2%** | 94.7% | 94.9% |
| Bert_WikiCREM_all | 76.8% | 75.8% | **98.7%** | **99.0%** |

Table 3.4: Evaluation of trained models on WinoBias subsets. The table is split into sections separated with horizontal lines. The first three entries correspond to Scenario A, and others fall under Scenario B. Each section contains a model that has been trained on WikiCREM and models that have not been. The best result in each section is highlighted in bold. The best overall result is underlined. Scores are given as the accuracy. The source of each SOTA result is listed in Section 3.2.2.1.

#### 3.2.2.3 Results

The results of the model evaluation on the test sets are shown in Tables 3.3 (Gap results), 3.4 (WinoBias results), and 3.5 (Wsc273, Wnli, WinoGender, and Pdp results). Notably, additional training on WikiCREM consistently improves the performance of the models in all scenarios and on most tests. Due to the small size of some test sets, some of the results are subject to deviation. This especially applies to Pdp (60 test samples) and Wnli (145 test samples).

We can see that Bert_WikiRand generally performs worse than Bert, with Gap and Pdp being notable exceptions. This shows that Bert is a strong baseline and that the improved performance of Bert_WikiCREM is not a consequence of training on shorter sentences or with a different loss function. Bert_WikiCREM consistently outperforms both baselines on all tests, showing that WikiCREM can be used as a standalone dataset.

Training on data from the target distribution (unsurprisingly) improves the performance the most. Models trained on Gap-train usually show more than

| | Dpr | Wsc273 | Wnli | WinoGender | Pdp |
|---|---|---|---|---|---|
| SOTA | 76.4% | **72.5**% | **74.7**% | 50.9% | 74.0% |
| Bert | 59.8% | 61.9% | 65.8% | 59.2% | 71.7% |
| Bert_WikiRand | 59.2% | 59.3% | 65.8% | 57.9% | 73.3% |
| Bert_WikiCREM | **67.4**% | **63.4**% | **67.1**% | **66.7**% | **76.7**% |
| Bert_Gap | 66.8% | 63.0% | 68.5% | 67.5% | **85.0**% |
| Bert_WikiCREM_Gap | **71.1**% | **64.1**% | **70.5**% | **75.4**% | 83.3% |
| Bert_Dpr | **83.3**% | 67.0% | 71.9% | 79.2% | 81.7% |
| Bert_Gap_Dpr | 79.4% | 65.6% | 72.6% | 75.8% | **86.7**% |
| Bert_WikiCREM_Dpr | 80.0% | **71.8**% | **74.7**% | **82.1**% | 76.7% |
| Bert_all | 80.1% | **70.0**% | 74.0% | **78.8**% | 81.7% |
| Bert_WikiCREM_all | **84.8**% | **70.0**% | **74.7**% | 76.7% | **86.7**% |

Table 3.5: Evaluation of trained models on the test sets of Wsc273, Wnli, WinoGender, and Pdp. The table is further split into sections separated with horizontal lines. The first three entries correspond to Scenario A, others fall under Scenario B. Each section contains a model that has been trained on WikiCREM and models that have not been. The best result in each section is in bold. The best overall result is underlined. Scores are given as the accuracy. The source of each SOTA is listed in Section 3.2.2.1.

a 20% increase in their $F_1$-score on Gap-test. Still, Bert_WikiCREM_Gap shows a consistent improvement over Bert_Gap on all subsets of the Gap test set. This confirms that WikiCREM works not just as a standalone dataset, but also as an additional pre-training in the transductive scenario.

Similarly, Bert_WikiCREM_Dpr outperforms Bert_Dpr on the majority of tasks, showing the applicability of WikiCREM to the scenario where additional training data is available. However, good results of Bert_Gap_Dpr show that additional training on a manually constructed dataset, such as Gap, can yield similar results as additional training on WikiCREM. The reason behind this difference is the impact of the data distribution. Gap, Dpr, and WikiCREM contain data that follows different distributions, which strongly impact the trained models. This can be seen when Bert_Gap is fine-tuned on Dpr to obtain Bert_Gap_Dpr, as the model's performance on Gap-test drops by 5.2%. WikiCREM's data distribution strongly differs from those of the test sets, as described in Section 3.2.1.

However, the best results are achieved when all available data are combined, as

shown by the models BERT_ALL and BERT_WIKICREM_ALL. BERT_WIKICREM_ALL achieves the highest performance on GAP, DPR, WNLI, and WINOBIAS among the evaluated models, and sets the new state-of-the-art result on GAP, DPR, and WINOBIAS. The new state-of-the-art result on the WINOGENDER dataset is achieved by the BERT_WIKICREM_DPR model, while BERT_WIKICREM_ALL and BERT_GAP_DPR set the new state-of-the-art result on the PDP dataset.

## 3.3 Discussion and Impact

In this chapter, I introduced WIKICREM, a large automatically-constructed dataset of training instances for pronoun resolution. When used to train the BERT language model, this dataset helped outperforming state-of-the-art models on 6 out of the 7 evaluated datasets. The employed data-generation procedure can be further applied to other large sources of text to generate more training sets for pronoun resolution.

This approach to unsupervised pre-training for coreference resolution was later revisited by other researchers. Shen et al. (2021) show that the same kind of pre-training can achieve even stronger results with ROBERTA (Liu et al., 2019c), a much better trained language model than BERT. Varkel and Globerson (2020) extend the pre-training objective by also masking pronouns, while Ye et al. (2020) show that despite being designed for coreference resolution, the introduced unsupervised pre-training helps with other tasks as well.

# Chapter 4

# Gender Bias in Pre-trained Models for Coreference Resolution

Diagnostic datasets that can detect biased models are an important prerequisite for bias reduction within natural language processing. However, undesired patterns in the collected data can make such tests incorrect. For example, if the feminine subset of a gender-bias-measuring coreference resolution dataset contains sentences with a longer average distance between the pronoun and the correct candidate, an RNN-based model may perform worse on this subset due to long-term dependencies. In this chapter, I demonstrate the existence of such undesired patterns in the GAP dataset for bias detection in coreference resolution. More specifically, I annotate GAP with spans of all personal names and show that examples in the female subset contain more personal names and a longer distance between pronouns and their referents, potentially affecting the bias score in an undesired way. To remove the impact of these patterns, I introduce a theoretically grounded method for weighting test samples to cope with such patterns in the test data. Using the introduced weighting method, I find the set of weights on the test instances that should be used for coping with these correlations, and re-evaluate 16 recently released coreference models.[1]

This weighting is a necessary step towards the evaluation of the models introduced in Chapter 3. As already mentioned there, the introduced models are not affected by the grammatical gender of the pronoun, making WINOGENDER and

[1]The annotations, the weights, and the code can be found at `https://github.com/vid-koci/weightingGAP`.

WinoBias an inappropriate choice for bias detection. Any conclusions about bias therefore have to be made on results on the Gap dataset, which suffers from two imbalances in the test set: Firstly, the feminine examples contain, on average, 0.75 more names per sentence than the masculine ones. Secondly, the correct candidate usually stands 0.5 candidates further away from the pronoun in feminine examples compared to masculine ones. These are all imbalances that can affect the score of a model.

In this chapter, I first introduce a general-purpose weighting method for removal of such imbalances. Then, I analyse the Gap dataset, demonstrating how the method can be used to address undesired evaluation bias. Having introduced a more accurate bias score for Gap dataset, I evaluate several recently released models for coreference, including the ones introduced in Chapter 3. Finally, I discuss the impact of unsupervised learning based on the scores of the models. This chapter is based on the publication (Kocijan et al., 2021).

## 4.1 Weighting Method

In this section, the sample-weighting method is presented. While, for ease of presentation, I describe the method on datasets for gender-bias detection, the method can be applied to any type of protected group and it also generalizes to tasks where one needs to detect biases among $n > 2$ classes, e.g., racial bias, by observing every pair of classes separately. The current version of the method assumes that accuracy is used as a metric of performance. I leave the analysis of other potential metrics to future work.

### 4.1.1 Definitions and Objectives

Let $D$ be a bias-testing dataset with $n$ examples $D = \{x_1, \ldots, x_n\}$. Let $A$ and $B$ be non-overlapping subsets of $D$. Let us assume that $A \cup B = D$, i.e., we ignore examples outside of observed sets, if any. The aim is to compare the performance of a model on $A$ and $B$ in order to see if the model is biased.

Let $S_1, \ldots, S_m$ be subsets of $D$, such that $S_j$ consists of all examples with a property that is not specific to the sets $A$ and $B$ but that could have an impact on

the performance of an evaluated model. For example, in the context of coreference resolution, one of the observed properties can be the number of referents in an example. In this case one set $S_k$ would consist of all examples with exactly $k$ potential referents. Note that these sets may overlap, as properties do not have to be mutually exclusive. We assume that these properties and sets are explicitly identified beforehand, and we refer to them as *identified properties*.

Let $C \subseteq D$ be a set of examples that a model solves correctly. Generally, the accuracy of a model corresponds to $|C| \,/\, |D|$. However, the performance of a model and hence $C$ may have a significantly different overlap with $S_j$ than with $D \setminus S_j$. Less formally, a model may be more/less likely to solve examples in the set $S_j$. To obtain an accurate bias measure, properties that do not influence the bias should be evenly distributed across $A$ and $B$. If this is not the case for a bias-detection dataset, we adapt the bias metric so that $S_j \cap A$ carries equal weight as $S_j \cap B$ in the final score.

To achieve this, we can assign to each example $x_i$ its weight $w_i$ and replace the accuracy with the weighted accuracy. We aim to find a set of weights $W$, such that $\sum_{x_i \in A \cap S_j} w_i = \sum_{x_i \in B \cap S_j} w_i$ for any $S_j$. Additionally, let us impose the following restrictions on the weights:

- Balance between the observed sets:

$$\sum_{x_i \in A} w_i = \sum_{x_i \in B} w_i \,.$$

- Fixed sum:

$$\sum_{i=1}^{n} w_i = n \,.$$

- Non-negativity:

$$w_i \geq 0, \text{ for all } i \in \{1, \ldots, n\}.$$

The first two will simplify the future derivations, while the last one is put in place to avoid the situation where an incorrect answer is preferred over the correct one. A direct consequence of the first two is that the sum of all weights of one gender is fixed to $\sum_{x_i \in A} w_i = \sum_{x_i \in B} w_i = \frac{n}{2}$.

There could exist several sets of weights that meet the above criteria. Among them, we prefer the distribution that minimizes the potential exacerbation of other

patterns in the data, that is, any changes in the bias score of a model that are not directly related to the above-identified properties. Let $\text{Acc}^D(C)$ and $\text{Acc}^D_W(C)$ be the unweighted and weighted accuracy, respectively, obtained by a set of correct answers $C$ on a set $D$. Since bias scores compare the performance on both $A$ and $B$, we aim to minimize both

$$|\text{Acc}^A_W(C \cap A) - \text{Acc}^A(C \cap A)| \text{ and} \tag{4.1}$$

$$|\text{Acc}^B_W(C \cap B) - \text{Acc}^B(C \cap B)| \tag{4.2}$$

for any $C \subseteq D$. This objective covers two cases:

- When $C$ corresponds to the correct answers of a model, we aim minimize the difference in weighted and unweighted accuracy on the sets $A$ and $B$.
- When $C$ is a set of examples with some property other than the ones captured by $S_1, \ldots, S_m$, we aim to retain its original overlap with $A$ and $B$. The overlap between sets with unidentified properties and the sets $A$ and $B$ should not be removed, as they may be an important indicator of the underlying bias. An example of such a property in the context of gender bias in NLP is the amount of out-of-vocabulary words. However, this can be a consequence of poor selection of the data, used to construct the vocabulary.

The introduced method minimizes the upper bound on the differences (4.1) and (4.2). By considering the upper bound rather than the average case, we avoid making assumptions about the distribution of $C$.

**Theorem 1** (Upper Bound on Introduced Noise)**.** *To minimize the upper bounds of*

$$|Acc^A_W(C \cap A) - Acc^A(C \cap A)| \text{ and}$$

$$|Acc^B_W(C \cap B) - Acc^B(C \cap B)|$$

*for any unknown set $C \subseteq D$, it is sufficient to minimize*

$$\sum_{\substack{x_i, x_j \in A \\ i>j}} max(w_i, w_j) + \sum_{\substack{x_i, x_j \in B \\ i>j}} max(w_i, w_j).$$

**Proof of Theorem 1** Let us derive the criterion function for the set $A$, as the derivation for set $B$ is analogous. We can denote

$$T_C := |\mathrm{Acc}_W^A(C \cap A) - \mathrm{Acc}^A(C \cap A)|.$$

First, we simplify $T_C$, piece by piece:

$$\mathrm{Acc}^A(C \cap A) = \frac{|C \cap A|}{|A|}$$

$$\mathrm{Acc}_W^A(C \cap A) = \frac{2}{n} \sum_{x_i \in C \cap A} w_i \,.$$

Combining this with the formula for $T_C$, we get:

$$\begin{aligned} T_C &= \frac{2}{n} \left( \sum_{x_i \in C \cap A} w_i \right) - \frac{|C \cap A|}{|A|} \\ &= \frac{2}{n} \sum_{x_i \in C \cap A} (w_i - \frac{n}{2|A|}) \,. \end{aligned}$$

To minimize $|T_C|$, we thus have to minimize

$$|\sum_{x_i \in C \cap A} (w_i - \frac{n}{2|A|})|.$$

To minimize the upper bound, we take a look at the scenarios that give the largest value of $|T_C|$. Let $w_{A(i)}$ be the $i$-th smallest weight corresponding to an example in $A$. For brevity, we use the constant $\lambda := \frac{n}{2|A|}$. The following properties hold:

$$\begin{aligned} \sum_{i=1}^{|A|} w_{A(i)} &= \sum_{x_i \in A} w_i = \frac{n}{2} \\ \sum_{i=1}^{|A|} (w_{A(i)} - \lambda) &= \sum_{i=1}^{|A|} (w_{A(i)} - \frac{n}{2|A|}) \\ &= \sum_{x_i \in A} w_i - \frac{n}{2} = 0 \,. \end{aligned}$$

In the scenario where $T_C$ is maximal, $C \cap A$ will include $|C \cap A|$ examples with either the largest or smallest weights. Let $k := |C \cap A|$, and let us first take a look

at the case where examples with the largest $k$ weights are in $C \cap A$. To minimize all such cases, we aim to minimize the following term:

$$\sum_{k=1}^{|A|} \sum_{i=\frac{n}{2}-k+1}^{|A|} (w_{A(i)} - \lambda) = \sum_{i=1}^{|A|} (w_{A(i)} - \lambda) i \,. \tag{4.3}$$

On the other hand, we aim to maximize the opposite case, i.e., when examples with the smallest $k$ weights are in $C \cap A$. The objective that we aim to maximize can be written as follows:

$$\begin{aligned}
&\sum_{k=1}^{|A|} \sum_{i=1}^{k} (w_{A(i)} - \lambda) = \\
&= - \sum_{k=1}^{|A|} \sum_{i=k+1}^{|A|} (w_{A(i)} - \lambda) \\
&= - \sum_{i=1}^{|A|} (w_{A(i)} - \lambda) i + \sum_{i=1}^{|A|} (w_{A(i)} - \lambda) \\
&= - \sum_{i=1}^{|A|} (w_{A(i)} - \lambda) i + 0 \\
&= - \sum_{i=1}^{|A|} (w_{A(i)} - \lambda) i \,.
\end{aligned}$$

We can see that maximizing this term is equivalent to minimizing term (4.3). Minimizing term (4.3) is therefore sufficient. We further simplify it:

$$\begin{aligned}
&\sum_{i=1}^{|A|} (w_{A(i)} - \lambda) i = \\
&= \sum_{\substack{x_i, x_j \in A \\ i>j}} \max(w_i - \lambda, w_j - \lambda) + \sum_{x_i \in A} (w_i - \lambda) \\
&= \sum_{\substack{x_i, x_j \in A \\ i>j}} \max(w_i - \lambda, w_j - \lambda) + 0 \\
&= \sum_{\substack{x_i, x_j \in A \\ i>j}} \max(w_i, w_j) - \lambda \cdot \frac{|A|(|A|-1)}{2} \,.
\end{aligned}$$

Since the second part of the term is constant, it is sufficient to minimize the following objective:

$$\sum_{\substack{x_i,x_j \in A \\ i>j}} \max(w_i, w_j)\,.$$

In the same way, the objective for the examples in set $B$ can be computed. Summing them up, we obtain the following objective:

$$\min \sum_{\substack{x_i,x_j \in A \\ i>j}} \max(w_i, w_j) + \sum_{\substack{x_i,x_j \in B \\ i>j}} \max(w_i, w_j)\,.$$

### 4.1.2 Solving the Optimization Problem

All listed conditions and criteria can be phrased as a linear program. The balance between the subsets, fixed sum, non-negativity, and removing correlations are linear constraints. The optimization objective can be phrased as a linear function by introducing the auxiliary variables $m_{i,j}; 1 \leq i, j \leq n$ for $\max(w_i, w_j)$. The following constraints have to hold for each of them: $m_{i,j} \geq w_i$ and $m_{i,j} \geq w_j$.

To summarize, I collect all derived constraints for the linear program below:

- $\sum_{x_i \in A} w_i = \sum_{x_i \in B} w_i$.
- $\sum_{i=1}^{n} w_i = n$.
- $w_i \geq 0$ for all $i \in \{1, \ldots, n\}$.
- $\sum_{x_i \in A \cap S_j} w_i = \sum_{x_i \in B \cap S_j} w_i$ for all $S_j$.
- For all $i, j$, such that $i < j$ and either $w_i, w_j \in A$ or $w_i, w_j \in B$: $m_{i,j} \geq w_i$ and $m_{i,j} \geq w_j$.

The criterion function is equal to

$$\min \sum_{\substack{x_i,x_j \in A \\ i>j}} m_{i,j} + \sum_{\substack{x_i,x_j \in B \\ i>j}} m_{i,j}\,.$$

A linear-program solver can then be used to find the minimum to this function.

## 4.2 Experiments

In this section, I demonstrate the use of the introduced weighting method on the GAP dataset. First, I show that feminine examples contain more candidates than masculine examples, and that the correct candidate usually stands further away from the pronoun in feminine examples than in masculine ones. The introduced weighting solves these imbalances, as several unbiased baselines obtain scores closer to 1 after the weighting (1 is a balanced score). Finally, I re-evaluate 16 publicly released models for coreference resolution, observing that the majority of these models were only slightly affected by these properties.

It is important to note that there is no guarantee that the used re-weighting scheme is the single correct scheme as other unidentified imbalances may exist. Even if potentially not perfect, the weighted accuracy is the most balanced metric on GAP available, removing all known imbalances and making it the best existing choice for the evaluation of models for coreference resolution.

### 4.2.1 The GAP Dataset Analysis

Recall that GAP is a corpus of challenging examples of pronoun resolution from English Wikipedia. It was introduced as a gender-balanced dataset, so that exactly half of the pronouns are masculine and half are feminine (Webster et al., 2018). The test set, which this chapter focuses on, consists of 2000 text spans. For each text span one pronoun has to be resolved and the model is expected to find the candidates on its own. It is guaranteed that candidates are always personal names from the input text and that at most one of them is the correct referent. Webster et al. (2018) define the bias measure as the ratio between the $F_1$-scores on the feminine and masculine subsets, $F_1^F \,/\, F_1^M$. An unbiased system is therefore expected to achieve a bias score around 1. An example from GAP can be found below:

*Kathleen first appears when Theresa and Myra visit* ***her*** *in a prison.*

*Kathleen:* ***True****, Theresa:* ***False***

During the scoring, the output of any evaluated model is compared to two candidates specified by the example.

Note that any incorrect candidate adds noise to the bias score. If a model answers *Theresa*, it will be penalized with an additional false-positive outcome, unlike a model that answered *Myra*, despite both being equally wrong. Since there is never more than one correct candidate per sentence and the candidates are not known in advance, comparing the prediction only with the correct candidate is thus not just sufficient, but also a more accurate bias measure. So, for measuring bias, I replace the $F_1$ score with accuracy, which has already been used as a performance metric in coreference resolution before (Emami et al., 2019; Rahman and Ng, 2012; Sakaguchi et al., 2020).

To be able to observe the effect of the weighting method, I first introduce a *plain* accuracy-based bias metric *acc-Bias*. I measure the accuracy on positive candidates in the masculine subset $A_M$ and the accuracy on positive candidates in the feminine subset $A_F$ and define acc-Bias as $A_F \,/\, A_M$. Results of this metric will be compared to a later-introduced weighted accuracy. Text spans with no positive examples are dropped, reducing the size of the test set by approximately 10%.

### 4.2.2 Baselines

I re-implement the random and token distance baselines introduced by Webster et al. (2018). First, I find all personal names in the input text using an off-the-shelf named entity recognition (NER). Each baseline is implemented with two NER systems: Google Cloud NL API[2] and Spacy en_core_web_lg,[3] abbreviated Spacy-lg. Additionally, as I have manually labeled all spans in the GAP test set that correspond to a personal name, I use these annotations to implement *Ground-Truth* baselines, which are thus not affected by potential mistakes of the NER systems.

In the random baseline implementation, a random personal name is picked from the list. Note that my implementation of the random baselines exhibits a different performance than the one from Webster et al. (2018). They report adding heuristics to eliminate obviously incorrect candidates; I do not follow them to avoid adding any noise.

[2] `https://cloud.google.com/natural-language/`
[3] `https://spacy.io/`

In the token distance baseline implementation the personal name closest to the pronoun is selected. The distance is measured in the number of tokens using the Spacy tokenizer. I rename this baseline as *Dist-1 baseline* and introduce *Dist-2* and *Dist-3 baselines*, where I pick the second closest and third closest personal name, respectively. If there are fewer than 2 or 3 candidates in the sentence, then I consider all answers to be `False`, that is, no answer is given. No higher-order distance-based baselines are introduced because their accuracy drops and with it the denominator in the bias score. This amplifies the noise caused by mistakes of the NER system and makes the results inconclusive.

Assuming unbiased NER systems and balanced data, the baselines should achieve a bias score very close to 1. The results of all baselines on the GAP test set are reported in Table 4.1, where we can see that most of the bias scores strongly differ from 1. In the next section, I show that imbalanced data are the reason behind this. Notice that the acc-Bias score of a model usually deviates further from 1 than its $F_1$-Bias score. These results empirically support our intuition that $F_1$-Bias is less representative than acc-Bias, as noise from negative candidates makes $F_1$-Bias less sensitive. Thus, the accuracy-based bias metric is more appropriate than its $F_1$-score counterpart.

### 4.2.3 Analysis of GAP

This section contains information on the distribution of the GAP test set. Figure 4.1 shows the distribution of the number of names per sentence in the masculine and the feminine subset of the test set. The difference between the masculine and the feminine subset is evident, with masculine examples usually containing fewer names per sentence than feminine examples. This histogram clearly shows why the random baseline did not achieve a bias score around 1.

In Figure 4.2 we can see how often the correct entity is the closest one to the pronoun, how often it is the second closest, third closest, etc. We can observe a similar pattern, with the closest and second closest candidate being correct more often in the masculine subset. The distribution is the source of a seemingly biased performance of the Dist-$k$ baselines.

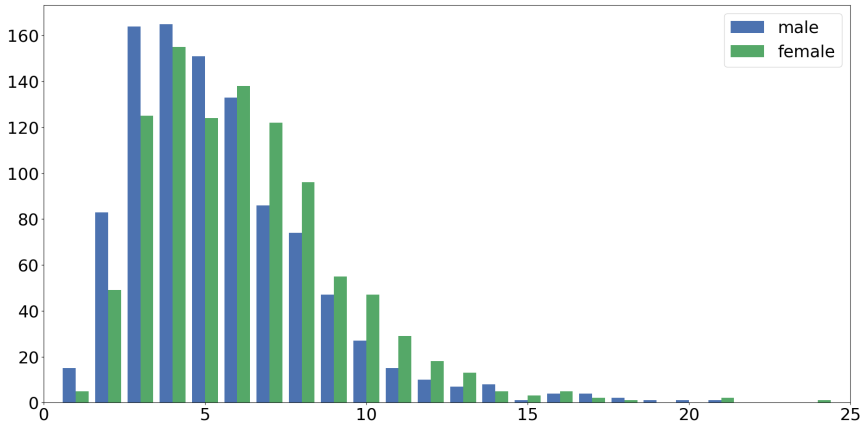



Figure 4.1: Histogram showing the number of personal names per sentence in the GAP dataset. The X-axis shows the number of names in the sentence, and the Y-axis the number of sentences with the corresponding number of personal names. The blue and green columns show the data for masculine and feminine examples, respectively.

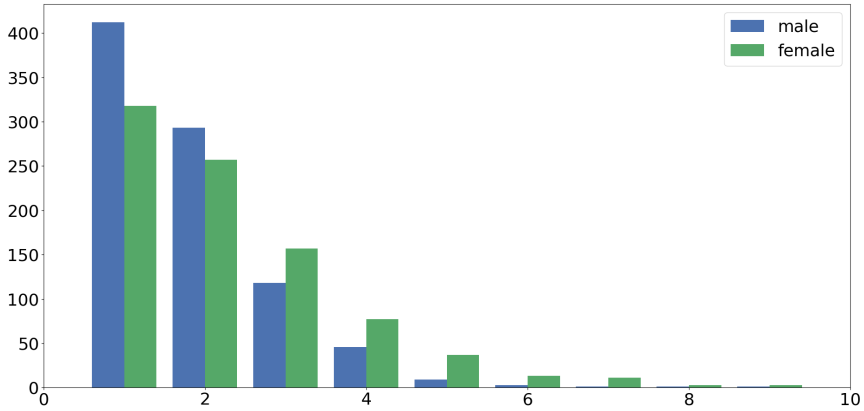



Figure 4.2: Histogram showing how often the correct entity is the closest, second closest, third closest, etc. entity to the pronoun in the GAP dataset, Y-axis showing the number of such sentences. The blue and green columns show the data for masculine and feminine examples, respectively.

| Baseline | $F_1$ | Accuracy | $F_1$-Bias | acc-Bias |
|---|---|---|---|---|
| Ground-Truth Random | 0.305 | 0.224 | 0.884 | 0.849 |
| Spacy-lg Random | 0.286 | 0.211 | 0.904 | 0.870 |
| Google-NER Random | 0.295 | 0.218 | 0.937 | 0.907 |
| Ground-Truth Dist-1 | 0.463 | 0.412 | 0.850 | 0.776 |
| Spacy-lg Dist-1 | 0.423 | 0.375 | 0.887 | 0.816 |
| Google-NER Dist-1 | 0.446 | 0.399 | 0.875 | 0.799 |
| Ground-Truth Dist-2 | 0.353 | 0.310 | 0.923 | 0.882 |
| Spacy-lg Dist-2 | 0.319 | 0.263 | 0.917 | 0.907 |
| Google-NER Dist-2 | 0.354 | 0.309 | 0.946 | 0.915 |
| Ground-Truth Dist-3 | 0.228 | 0.156 | 1.270 | 1.347 |
| Spacy-lg Dist-3 | 0.205 | 0.134 | 1.490 | 1.585 |
| Google-NER Dist-3 | 0.219 | 0.150 | 1.312 | 1.426 |

Table 4.1: Performance and bias metrics on baseline systems on GAP, implemented with two different NER systems as well as the ground-truth personal names. The reported performance of the random classifier is obtained by averaging the performance over $10,000$ repetitions. Note that deviations can happen when NER-system extractions are incorrect.

The analysis of the manually annotated spans of personal names shows that masculine examples on average contain 5.55 personal names (standard deviation 3.18), while feminine examples contain 6.30 names on average (standard deviation 3.44). This confirms the hypothesis about imbalances in the data and explains why the Ground-Truth random baseline achieved a bias score different from 1.

Similarly, all annotated personal names in each sentence can be sorted by distance to the pronoun in the same manner as done by the Dist-$k$ baselines. Let us find the position of the correct candidate on this ordered list. The average position of the correct candidate in the masculine subset is 1.86 (standard deviation 1.19) candidates away from the pronoun, while the average position in the feminine subset is 2.32 (standard deviation 1.54) candidates away from the pronoun, explaining the bias scores of the Dist-k baselines. Examples with no correct candidate were not considered in this statistic.

### 4.2.4 Weighting GAP

Using manual annotations of personal names, let $N_k$ be the set of all examples with exactly $k$ personal names, and let $D_k$ be the set of all examples where the correct candidate is the $k$-th closest candidate to the pronoun. The $N_k$ and $D_k$ sets form the sets that were generically denoted as $S_1, \ldots, S_m$ in Section 4.1. Thus, the sets $N_k$ and $D_k$ are used as input to the balancing method to obtain a linear program, which I solve with the LINPROG optimization tool from Matlab, version R2019b.

Note that balancing of GAP with downsampling does not scale. To obtain a dataset that is balanced only w.r.t. the number of candidates in a sentence or only w.r.t. distance, the dataset has to be downsampled to 75% of its size. To obtain a dataset that is balanced w.r.t. both, the dataset has to be downsampled below 70% of its size, which is a significant drop in the number of examples. Should one want to remove an additional undesired pattern, this number would likely drop even further, making the pruning method unscalable.

The obtained weighted bias metric is termed *W-Bias*. I highlight that the introduced constraints are not a guarantee that W-Bias is completely balanced, as other imbalances in the data may exist. However, given Theorem 1, the known imbalances have been balanced out, while having introduced the least noise possible. This makes the introduced metric preferable over the existing one, i.e., no weighting. A visualization of the weights is given in Section 4.2.5.

To assess the introduced weights, the unbiased baselines are evaluated on the newly introduced W-Bias metric. To confirm that the weighting method does not introduce noise relative to unidentified properties, two ablation experiments are performed. In the first one the distance is ignored, while in the second experiment the number of candidates is ignored. To this end, I introduce two more bias metrics: $\text{W}_{\text{num}}$-Bias and $\text{W}_{\text{dist}}$-Bias. In $\text{W}_{\text{num}}$-Bias, the sets $D_k$, $k \in \mathbb{N}$ were not included as the input to the balancing procedure. $\text{W}_{\text{num}}$-Bias is only balanced with respect to the number of names per sentence. On the other hand, $\text{W}_{\text{dist}}$-Bias does not include the sets $N_k$, $k \in \mathbb{N}$, meaning that it is only balanced with respect to the distance between the pronoun and the correct answer. I show that, for random baselines, the following holds: $|1 - \text{W-Bias}| \leq |1 - \text{W}_{\text{dist}}\text{-Bias}| \leq |1 - \text{acc-Bias}|$, that is, balancing relative to distance does not exacerbate bias

scores of random baselines, and additional balancing relative to the number of names further decreases deviations from the unbiased score (of 1). Similarly, it is shown that for Dist-$k$ baselines, $|1 - \text{W-Bias}| \leq |1 - \text{W}_{\text{num}}\text{-Bias}| \leq |1 - \text{acc-Bias}|$.

The results are reported in Table 4.2. In the columns that correspond to $\text{W}_{\text{num}}$-Bias and $\text{W}_{\text{dist}}$-Bias numbers in italics are expected to be similar to the numbers predicted by W-Bias. Notably, the inequations in the previous paragraph hold for all baselines, showing that the weights indeed do not exacerbate the bias of unidentified properties. Moreover, it is evident that the W-Bias scores achieved by the baselines are consistently closer to 1 than their acc-Bias scores, confirming that the introduced weights balance the bias metric. In particular, the W-Bias score of Ground-Truth baselines is equal to 1, i.e., unbiased. The minimal deviation from 1 of the Ground-Truth W-bias score for the *Dist-3* baseline is a consequence of a disagreement between our span annotations with the spans of gold labels. Bias scores of the *Dist-2* and *Dist-3* baselines implemented with NER systems are subject to larger deviations because these baselines are more sensitive to disagreement between the NER system and my annotations of the name spans.

| Baseline | acc-Bias | W-Bias | $\text{W}_{\text{num}}$-Bias | $\text{W}_{\text{dist}}$-Bias | $W_t$-Bias |
|---|---|---|---|---|---|
| Ground-Truth Random | 0.849 | *1.000* | *0.995* | 0.899 | *1.000* |
| Spacy-lg Random | 0.870 | *0.975* | *0.980* | 0.905 | *0.984* |
| Google-NER Random | 0.907 | *1.019* | *1.021* | 0.949 | *1.020* |
| Ground-Truth Dist-1 | 0.776 | *1.000* | 0.804 | *1.000* | *1.000* |
| Spacy-lg Dist-1 | 0.816 | *1.015* | 0.824 | *1.029* | *1.018* |
| Google-NER Dist-1 | 0.799 | *0.986* | 0.793 | *1.016* | *0.994* |
| Ground-Truth Dist-2 | 0.882 | *1.000* | 0.920 | *1.000* | *1.000* |
| Spacy-lg Dist-2 | 0.907 | *0.962* | 0.932 | *0.977* | *0.968* |
| Google-NER Dist-2 | 0.915 | *1.000* | 0.983 | *1.001* | *1.026* |
| Ground-Truth Dist-3 | 1.347 | *1.006* | 1.266 | *1.010* | *1.007* |
| Spacy-lg Dist-3 | 1.585 | *1.118* | 1.494 | *1.152* | *1.200* |
| Google-NER Dist-3 | 1.426 | *1.154* | 1.368 | *1.111* | *1.116* |

Table 4.2: Performance and bias metrics on baseline systems on GAP, implemented with two different NER systems as well as the ground-truth personal names. The reported performance of the random classifier is obtained by averaging the performance over $10,000$ repetitions. If the evaluated baseline is expected to achieve a score of 1 on some metric due to balancing, the score is written in *italics*. Note that deviations can happen when NER-system extractions are incorrect.

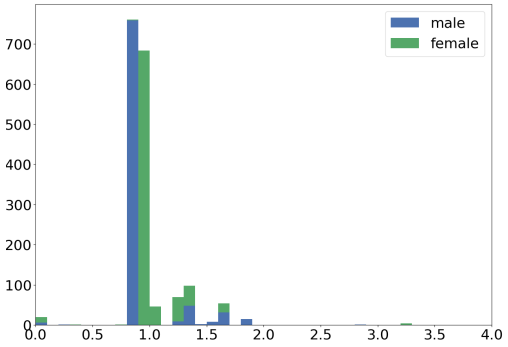



Figure 4.3: Histogram showing the distribution of W-Bias weights, split into intervals of size 0.1. Blue shading corresponds to masculine examples, while the green shading corresponds to feminine examples. The weights are centered around 1. The nine largest weights are not included in the histogram, as they have values over 4.

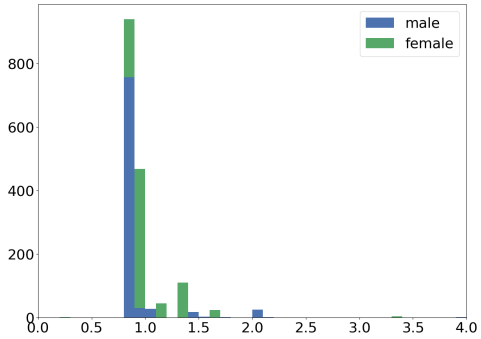



Figure 4.4: Histogram showing the distribution of $W_t$-Bias weights, split into intervals of size 0.1. Blue shading corresponds to masculine examples, while the green shading corresponds to feminine examples. The weights are centered around 1. The largest weight (7.68) is not included.

Weighting with respect to one of the imbalances sometimes helped balancing the baseline that was affected by the other imbalance. For example, balancing the number of names per sentence ($W_{num}$-Bias) resulted in improved bias scores of all Ground-Truth Distance baselines. This implies that there exists a correlation between the number of personal names in the sentence, the distance between the pronoun, and the correct candidate in the GAP test set.

### 4.2.5 Analysis of Weights

A visualization of W-Bias weights of examples is shown in Figure 4.3. It confirms that weights gravitate around 1 despite the constraints. Moreover, fewer than 1% of the weights are set to 0. A manual investigation into these examples shows that they are often very long text spans with long lists of names, such as family trees or cast lists. Several of them are not grammatically correct sentences, but rather lists from Wikipedia that were not removed during the annotation.

However, the distribution of W-Bias weights contains some outliers, that is,

examples with unusually large weights. There are 9 weights larger than 4.0 that are not pictured: 4.84, 4.97, 4.97, 5.43, 6.41, 7.05, 7.05, 9.20, and 9.72, all of them corresponding to male examples. Ten examples with the largest weights have a weight average of 6.29 (the average weight overall is 1.0). These large weights may result in the instability of the results.

Sets at the tails of graphs on Figures 4.2 and 4.1 can be highly gender-imbalanced. The weighting method counteracts these imbalances by assigning large weights to the weights in the under-represented class. While this is theoretically correct, it may be undesirable as it means that few out-of-distribution examples carry a lot of weight in the final score, which could introduce noise.

Such large weights can be avoided by removing highly-imbalanced subsets of the data. I introduce a trimmed $W$-score, called $W_t$-score. Examples with more than 15 personal names and examples where the correct candidate is the $k$-th closest for $k \geq 5$ are removed from this score, reducing the size of the dataset to 1670 examples (83.5% of the original size). Numbers $k \geq 5$ and 15 personal names were selected manually by consulting Figures 4.2 and 4.1 . The rest of the examples are assigned new weights with the introduced method. The ten largest weights of $W_t$-score have a weight average of 3.3, strongly reducing the problem of outliers. Comparing $W_t$-bias with $W$-bias in Table 4.2 shows that such outliers mainly affected Dist-2 and Dist-3 baselines.

A visualization of $\mathrm{W_t}$-Bias weights in Figure 4.4 shows that trimming largely solves the problem of outliers, since the largest few weights now carry much less weight than before. Moreover, there are no examples with weight 0. Female weights are slightly larger on average, because there are more male (865) than female (805) examples in the trimmed dataset. I did not conduct any additional trimming to avoid further decreasing the size of the dataset.

The dataset after trimming that was done as part of the $\mathrm{W_t}$-Bias score is visualized below. The distribution of the number of names and how close the correct candidate is after *trimming* are reported in Figures 4.5 and 4.6. The trimmed dataset contains fewer highly-imbalanced subsets.

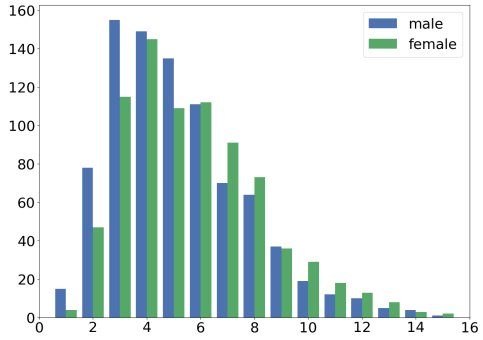



Figure 4.5: Histogram showing the number of personal names per sentence in the GAP dataset after the trimming conducted for the $W_t$-Bias score. The blue and green columns show the data for masculine and feminine examples, respectively.

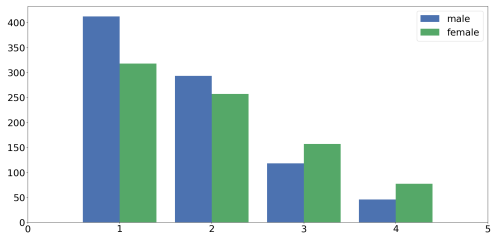



Figure 4.6: Histogram showing how often the correct entity is the closest, second closest, third closest, etc. entity to the pronoun in the GAP dataset after the trimming conducted for the $W_t$-Bias score. The blue and green columns show the data for masculine and feminine examples, respectively.

## 4.3 Evaluation of Bias in Coreference Models

Having shown that the introduced measure strongly reduces the impact of observed imbalances in the data, I re-evaluate recent models for coreference resolution. Following Webster et al. (2018), I consider systems that detect name spans for inference automatically and access labelled spans only to output predictions. I thus do not consider models that were submitted at the Kaggle competition on the GAP dataset, because they do not conform to this norm (Webster et al., 2019). The results are reported in Table 4.3.

Comparing acc-Bias and W-Bias, only a few models change their bias score visibly, indicating that not all models were equally affected by the observed imbalances. While these bias scores are not directly comparable with the original $F_1$-based bias metric, I hypothesize that the imbalances in the data also affected that score. Comparing W-Bias and $W_t$-Bias shows that most of the models were minimally affected by the outliers in the weights.

The better performing models tend not to change their bias scores significantly. I hypothesize that they are less affected by the observed imbalances in the data distribution. At the same time, a larger denominator (female score) in the bias

| | $F_1$ | $F_1$-Bias | acc-Bias | W-Bias | $W_t$-Bias |
|---|---|---|---|---|---|
| [1]BERT | 0.500 | 0.88 | 0.86 | 0.85 | 0.87 |
| [1]BERT_WIKICREM | 0.590 | 0.95 | 0.93 | 0.90 | 0.92 |
| [1]BERT_GAP | 0.752 | 0.99 | 0.97 | 0.96 | 0.97 |
| [1]BERT_DPR | 0.612 | 1.00 | 0.96 | 0.94 | 0.96 |
| [1]BERT_ALL | 0.760 | 1.03 | 1.03 | 1.03 | 1.04 |
| [1]BERT_GAP_DPR | 0.704 | 1.01 | 1.00 | 1.00 | 1.00 |
| [1]BERT_WIKICREM_GAP | 0.778 | 1.01 | 1.00 | 1.00 | 1.01 |
| [1]BERT_WIKICREM_DPR | 0.646 | 0.99 | 0.98 | 0.97 | 0.96 |
| [1]BERT_WIKICREM_ALL | 0.783 | 1.02 | 1.01 | 1.00 | 1.01 |
| [2]BERT_BASE | 0.824 | 0.97 | 0.97 | 0.96 | 0.96 |
| [2]BERT_LARGE | 0.856 | 0.97 | 0.96 | 0.96 | 0.97 |
| [3]SPANBERT_BASE | 0.855 | 0.96 | 0.95 | 0.95 | 0.95 |
| [3]SPANBERT_LARGE | 0.877 | 0.95 | 0.94 | 0.93 | 0.93 |
| [4]E2E | 0.733 | 0.93 | 0.92 | 0.92 | 0.91 |
| [5]E2E_ADV | 0.747 | 0.93 | 0.91 | 0.93 | 0.90 |
| [6]REFREADER | 0.794 | 0.96 | 0.95 | 0.97 | 0.97 |

Table 4.3: Evaluation of several state-of-the-art models for coreference resolution on GAP, with several bias scores reported. [1](Kocijan et al., 2019a); [2](Joshi et al., 2019); [3](Joshi et al., 2020); [4](Lee et al., 2018); [5](Subramanian and Roth, 2019); [6](Liu et al., 2019a). I used publicly shared code and models in all cases, except for Referential Reader[6], where code was not publicly available at the time. Instead, the evaluation was performed on the results provided by the authors. The numbers differ from the paper, as the authors averaged results over several seeds, but only shared one version. The results from Joshi et al. (2019) differ from their paper, as the author shared a different checkpoint.

formula results in a smaller absolute difference. Similarly, RNN-based models (models [4,5,6]) change their scores more than transformer-based models (models [1,2,3]), implying that RNN-based models were more affected by the number of candidates and the distance between the correct candidate and the pronoun than transformer-based models.

**Statistical significance of bias metrics.** The randomization test (Yeh, 2000) was employed to compare the $W_t$-Bias scores of a few models, listed in Table 4.3. E.g., the difference between BERT and BERT_GAP is significant ($p = 0.024$), as is the difference between BERT_WIKICREM and BERT_WIKICREM_GAP

($p = 0.017$). Fine-tuning BERT on GAP thus seems to significantly increase its bias score, implying that its predictions are not as biased. On the other hand, the difference of the E2E and E2E_ADV models is not significant ($p = 0.364$), implying that the seemingly negative impact of adversarial sampling could be a coincidence.

## 4.4 Summary and Discussion

In this chapter, I have introduced a test-set weighting method to remove undesired imbalances in bias-measuring datasets, without exacerbating other potentially undesired patterns.

The method was demonstrated on the GAP test set, which contained such undesired irregularities. I annotated the dataset with spans of all personal names and introduced the bias metrics W-Bias and $W_t$-Bias that balance out the observed irregularities. While there is no guarantee that these scores balance out all data irregularities in GAP, it was shown that they balance out the ones that we are aware of.

A potential improvement of the method constitutes removing the need to identify the biases in the data, however, this step is common to existing methods that deal with bias. It is not unreasonable to expect that the existence of bias has to be noticed and demonstrated before one can start planning the de-biasing. This already satisfies the prerequisites to use the introduced method. Manual annotation of examples like the one in this work is not always necessary, as automatic tools (e.g., NER systems) can be used. However, manual annotation likely ensures the high quality of the test data.

With the GAP bias metric corrected, the impact of pre-training on biased behaviour of models can be discussed. As can be seen from Table 4.3, models generally improve their bias scores with additional pre-training. One could claim that WIKICREM, DPR, and GAP fine-tuning helps with model de-biasing, and that WIKICREM somehow alleviates the negative impact of BERT's pre-training. While this may hold to some degree, the main reason behind bias score improvement is likely simpler.

With more pre-training, models may pick up undesired societal biases, but primarily, they get more accurate. As models slowly converge towards the per-

fect score on the test set, they simultaneously converge towards the perfect bias score as well. In the observed setup, the correct answer always exists and can be unambiguously determined. Therefore, further improving the performance of an already well-performing model often results in an improved bias score. As counter-intuitive as this may seem, more pre-training can help against biased behaviour on tasks where models already achieve high performance and the answer can be unambiguously determined. More pre-training of course cannot be expected to decrease biased behaviour on truly ambiguous examples and open-ended tasks such as language generation. In coreference resolution, one can safely assume that truly ambiguous examples are rare in practice and this approach might be viable.

Implications of these results for bias-measuring benchmark design will be further discussed in Chapter 7.

# Chapter 5

# Transfer Learning for Knowledge Base Completion

In this chapter, I introduce the approach to knowledge transfer from one collection of facts to another without the need for entity or relation matching. Before the introduction of this method, there were no other widely adopted methods for transfer of knowledge in knowledge base completion. The method works for both *canonicalized* knowledge bases and *uncanonicalized* or *open knowledge bases*, i.e., knowledge bases where more than one copy of a real-world entity or relation may exist. Such knowledge bases are a natural output of automated information extraction tools that extract structured data from unstructured text. The main contribution is a method that can make use of large-scale pre-training on facts, which were collected from unstructured text, to improve predictions on structured data from a specific domain. The introduced method is most impactful on small datasets such as ReVerb20k, where a 6% absolute increase of mean reciprocal rank and 65% relative decrease of mean rank over the previously best method was achieved, despite not relying on large pre-trained models like Bert.

More specifically, I replace embeddings of entities and relations with RNN-based encoders, which encode entities and relations from textual representations to embeddings. They are pre-trained jointly with a KBC model on a large OKBC benchmark. This pre-trained KBC model and encoders are then used to initialize the final model that is later fine-tuned on a smaller dataset. More specifically, KBC parameters that are shared among all inputs are used as an initialization

of the same parameters of the fine-tuned model. When initializing the input-specific embeddings, I introduce and compare two approaches: Either the pre-trained entity and relation encoders are also used and trained during the fine-tuning, or they are used in the beginning to compute the initial values of all entity and relation embeddings, and then dropped.

I evaluate this approach with three different KBC models and on five datasets, showing consistent improvements on most of them. I show that pre-training turns out to be particularly helpful on small datasets with scarce data by achieving SOTA performance on the ReVerb20K and ReVerb45K OKBC datasets and consistent results on the larger KBC datasets Fb15k237 and Wn18rr. The obtained results imply that even larger improvements can be obtained by pre-training on a larger corpus.

This chapter is based on the publication **?**. First section introduces the modifications to KBC models, necessary for transfer learning, together with all KBC models used in the experiments. Second section describes all performed experiments, while third section describes and discusses the results. The fourth section discusses the significance of the outcomes and future developments. The code used for the experiments is available at `https://github.com/vid-koci/KBCtransferlearning`.

## 5.1 Model for Transfer Learning

In this section, I introduce the model architecture and how a pre-trained model is used to initialize the model for fine-tuning. The setup consists of two encoders, one for entities and one for relations, and a KBC model. Given a triplet $\langle h, r, t \rangle$, the entity encoder is used to map the head $h$ and the tail $t$ into their vector embeddings $\mathbf{v}_h$ and $\mathbf{v}_t$, while the relation encoder is used to map the relation $r$ into its vector embedding $\mathbf{v}_r$. These are then used as the input to the KBC algorithm of choice to predict their score (correctness) using the loss function, as defined by the KBC model. The two parts of the model are architecturally independent of each other and will be described in the following paragraphs. An illustration of the approach is given in Figure 5.1.

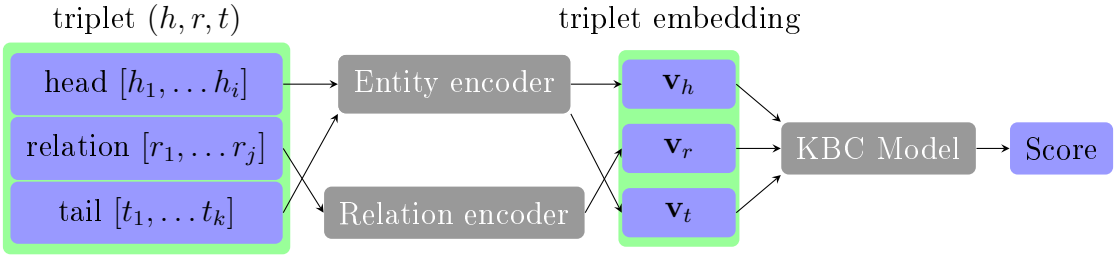



Figure 5.1: Diagram of the introduced approach. Green and blue blocks represent data, grey blocks represent models, and arrows represent data flow. Entity and relation encoders are used to map a triplet $(h, r, t)$ from their names (textual representations) to their vector embeddings $(\mathbf{v}_h, \mathbf{v}_r, \mathbf{v}_t)$. These vectors are then used as the input to a KBC algorithm of choice to compute the score of the triplet.

### 5.1.1 Encoders

I compare two types of mappings from an entity to a low-dimensional vector space. The first approach is to assign each entity and relation its own embedding, initialized randomly and trained jointly with the model. This is the default approach used by most KBC models, however, to distinguish it from the RNN-based approach, I denote it NoEncoder.

The second approach that I test is to use an RNN-based mapping from the textual representation of an entity or relation (name) to its embedding. I use the GloVe word embeddings (Pennington et al., 2014) to map each word into a vector, and then use them as the input to the entity encoder, implemented as a Gru (Cho et al., 2014). To separate it from the NoEncoder, I call this the Gru encoder. Broscheit et al. (2020) test alternative encoders as well, but find that RNN-based approaches (Lstms, in their case) perform the most consistently across the experiments.

The number of the NoEncoder parameters grows linearly with the number of entities, while the number of the parameters of the Gru encoder grows linearly with the vocabulary. For large knowledge bases, the latter can significantly decrease the memory usage, as the size of the vocabulary is often smaller than the number of entities and relations.

**Transfer between datasets.** In this work, pre-training is always done using GRU encoders, as the transfer from the NOENCODER to any other encoder requires entity matching, which we avoid.

When fine-tuning is done with a GRU encoder, its parameters are initialized from the pre-trained GRU parameters. The same applies for the vocabulary, however, if the target vocabulary includes any unknown words, their word embeddings are initialized randomly.

For the initialization of the NoEncoder setup, the pre-trained GRU is used to generate initial values of all vector embeddings by encoding their textual representations. Any unknown words are omitted, and entities with no known words are initialized randomly.

An equivalent process is used for relations. During preliminary experiments, I have also tried pre-training the encoders on the next-word-prediction task on English Wikipedia, however, that turned out to have a detrimental effect on the overall performance compared to randomly-initialized GRUs ($2-3\%$ MRR drop and slower convergence). That line of experiments was not continued.

### 5.1.2 Knowledge Base Completion Models

I use three models for knowledge base completion, CONVE (Dettmers et al., 2018), TUCKER (Balažević et al., 2019), and 5*E (Nayyeri et al., 2021), chosen for their strong performance on various KBC benchmarks. The models have already been introduced in Section 2.3.3, so this section only covers specific implementation choices and how transfer of knowledge is carried out.

Recall that TUCKER assigns a score to each triplet by multiplying the vectors with a core tensor $\mathcal{W} \in \mathbb{R}^{d_e \times d_e \times d_r}$, where $d_e$ is the dimension of entities, and $d_r$ is the dimension of relations. Throughout this work, I make the simplifying assumption that $d_e = d_r$ and use the same dropout rates for all inputs to reduce the number of hyperparameters. During the transfer, $\mathcal{W}$ from the pre-trained model is used to initialize $\mathcal{W}$ in the fine-tuned model.

CONVE assigns a score to each triplet by passing $\mathbf{v}_h$ and $\mathbf{v}_r$ through a CNN and aims to map it into $\mathbf{v}_t$. The output, a $d_e$-dimensional vector, is multiplied with $\mathbf{v}_t$ and summed with a tail-specific bias term $b_t$ to obtain the score of the triplet.

Just like with TuckER, it is assumed that all dropout rates in the model are the same to reduce the number of hyperparameters. During transfer, the parameters of the pre-trained CNN are used as the initialization of the CNN in the fine-tuned model. Bias terms of the fine-tuned model are initialized at random because they are entity-specific.

5*E model has no shared parameters between different relations and entities. Pre-training thus only serves to initialize the embeddings.

At the time of the evaluation, the model is given a triplet with a missing head or tail and is used to rank all the possible entities based on how likely they are to appear in place of the missing entity. Following Dettmers et al. (2018), the head-prediction samples are transformed into tail-prediction samples by introducing reciprocal relations $r^{-1}$ for each relation and transforming $\langle ?, r, t \rangle$ into $\langle t, r^{-1}, ? \rangle$. Following Gupta et al. (2019), the name of the reciprocal relation is created by adding the prefix "inverse of".

During preliminary experiments, I have also experimented with BoxE (Abboud et al., 2020), however, I have decided not to use it for further experiments, since it was much slower to train and evaluate in comparison to other models. A single round of training of BoxE with Gru encoders on OlpBench takes over 24 days, making it an impractical choice.

## 5.2 Experiments

This section describes the experimental setup and the baselines. All experiments were conducted on the five datasets previously introduced in Section 2.3.2: OlpBench, ReVerb20k, ReVerb45k, Fb15k237, and Wn18rr.

### 5.2.1 Experimental Setup

The performance of the randomly-initialized and pre-trained Gru and NoEncoder variants of each of the three KBC models are observed and compared. Pre-training was done with Gru encoders on OlpBench. In this section, I cover all the details and implementation choices in the experimental setup. Not all choices in experiment design are optimal w.r.t. performance; instead, they were

chosen to obtain a fair comparison between different models. For example, I make several simplifying assumptions to reduce the hyperparameter space, as already noted in the model description. While these simplifications can result in a performance drop, they allow us to run exactly the same grid search of hyperparameters for all models, excluding the human factor or randomness from the search.

Following Ruffinelli et al. (2020), I use 1-N scoring for negative sampling and cross-entropy loss for all models. To put it simply, I compute the loss on a test sample $\langle h, r, ? \rangle$ by computing cross-entropy over all possible tails, maximizing the likelihood of predicting the correct one. The Adam optimizer (Kingma and Ba, 2015) was used to train the network. I follow Dettmers et al. (2018) and Balažević et al. (2019) with the placement of the batch norm and dropout in ConvE and TuckER, respectively. To find the best hyperparameters, grid search is used for all experiments to exhaustively compare all options. Despite numerous simplifications the re-implementations of the baselines (non-pre-trained NoEncoder models) perform comparable to the original reported values. The experiments were performed on a DGX-1 cluster, using one Nvidia V100 GPU per experiment.

**Pre-training setup.** Pre-training was only done with GRU encoders, as discussed in Section 5.1.1. Due to the large number of entities in the pre-training set, 1-N sampling is performed only with negative examples from the same batch, and batches of size 4096 were used, following Broscheit et al. (2020). The learning rate was selected from $\{1 \cdot 10^{-4}, 3 \cdot 10^{-4}\}$, while the dropout rate was selected from $\{0.2, 0.3\}$ for ConvE and $\{0.3, 0.4\}$ for TuckER. For 5*E, the dropout rate was not used, but N3 regularization was (Lacroix et al., 2018), with its weight selected from {0.1,0.03}.

For TuckER, models with embedding dimensions 100, 200, and 300 were trained. The best model of each dimension was saved for fine-tuning. For ConvE, models with embedding dimensions 300 and 500 were trained, and the best model for each dimension was saved for fine-tuning. Following Gupta et al. (2019), I used a single 2D convolution layer with 32 channels and $3 \times 3$ kernel size. When the dimension of entities and relations is 300, they are reshaped into $15 \times 20$ inputs, while the $20 \times 25$ input shapes are used for 500-dimensional embeddings. For 5*E,

models with embedding dimensions 200 and 500 were trained, and the best model for each dimension was saved for fine-tuning.

Following Broscheit et al. (2020), each model was trained for 100 epochs. Testing on the validation set is performed each 20 epochs, and the model with the best overall mean reciprocal rank (MRR) is selected.

**Fine-tuning setup.** Fine-tuning is performed in the same way as the pre-training, however, the models were trained for 500 epochs, and a larger hyperparameter space was considered. More specifically, the learning rate was selected from $\{3 \cdot 10^{-5}, 1 \cdot 10^{-4}, 3 \cdot 10^{-4}\}$. The dropout rate was selected from $\{0.2, 0.3\}$ for ConvE and $\{0.3, 0.4\}$ for TuckER. The weight of N3 regularization for the 5*E models was selected from $\{0.3, 0.1, 0.03\}$. The batch size was selected from $\{512, 1024, 2048, 4096\}$. The same embedding dimensions as for pre-training were considered.

### 5.2.2 Baselines

I compare this work to a collection of baselines, re-implementing and re-training them where appropriate.

**Models for Knowledge Base Completion.** The main three KBC models, ConvE, TuckER, and 5*E are evaluated with and without encoders. The results of these models obtained by related work are included and compared to other KBC models from literature, such as BoxE (Abboud et al., 2020), ComplEx (Trouillon et al., 2017), TransH (Wang et al., 2014), and TransE (Bordes et al., 2013). I highlight that results from external work were usually obtained with more experiments and a broader hyperparameter search compared to experiments from my work, where I tried to ensure exactly the same environment for a large number of models, for fair comparisons.

**Models for Knowledge Base Canonicalization.** Gupta et al. (2019) use Cesi (Vashishth et al., 2018) for knowledge base canonicalization to improve the predictions of KBC models, testing multiple methods to incorporate such data into the model. The tested methods include graph convolution neural networks (Bruna

et al., 2014), graph attention neural networks (Veličković et al., 2017), and newly introduced local averaging networks (LANs). Since LANs consistently outperform the alternative approaches in all of their experiments, they are included here as the only baseline of this type.

**Transfer Learning from Larger Knowledge Bases.** Lerer et al. (2019) release pre-trained embeddings for the entire WikiData knowledge graph, computed with their BigGraph mehod. I use them to initialize the TuckER model. I initialize the entity and relation embeddings with their respective embeddings from WikiData. Since pre-trained WikiData embeddings are only available for dimension $d = 200$, I compare them to the pre-trained and randomly initialized NoEncoder_TuckER of dimension $d = 200$ for a fair comparison. I do not re-train the WikiData embeddings with other dimensions due to the computational resources required. Moreover, I do not use this baseline on KBC datasets, since a potential training and test set cross-contamination could not be avoided. WikiData was constructed from Freebase and is linked to WordNet, the knowledge bases used to construct the Fb15k237 and Wn18rr datasets.

**Transfer Learning from Language Models.** Pre-trained language models can be used to answer KB queries (Petroni et al., 2019; Yao et al., 2019). I compare the results to KG-Bert on the Fb15k237 and Wn18rr datasets (Yao et al., 2019), and to Okgit on ReVerb45K and ReVerb20K datasets (Chandrahas and Talukdar, 2021). Using large transformer-based language models for KBC can be slow. I estimate that a single round of evaluation of KG-Bert on the ReVerb45K test set takes over 14 days on a Tesla V100 GPU, not even accounting for training or validation. Yao et al. (2019) report in their code repository that the evaluation on Fb15k237 takes over a month.[1] For comparison, the evaluation of any other model in this work takes up to a maximum of a couple of minutes. Not only is it beyond my resources to perform equivalent experiments for KG-Bert as for other models, but I also consider this approach to be impractical for link prediction.

[1] https://github.com/yao8839836/kg-bert/issues/8

| | MR | MRR | H@10 |
|---|---|---|---|
| GRU_TUCKER | **57.2K** | .053 | .097 |
| GRU_CONVE | **57.2K** | .045 | .086 |
| GRU_5*E | 60.1K | **.055** | **.101** |
| (Broscheit et al., 2020) | – | .039 | .070 |

Table 5.1: Comparison of pre-trained models on OlpBench with the previous best result. The best value in each column is written in **bold**.

Chandrahas and Talukdar (2021) combine BERT with the approach introduced by Gupta et al. (2019), taking advantage of both knowledge base canonicalization tools and large pre-trained transformers at the same time. Their approach is more computationally efficient since only the pair $\langle h, r \rangle$ is encoded with BERT instead of the entire triplet. This reduces the number of required passes through the transformer model by an order of magnitude. At the time of this work, this was the best-performing published approach to OKBC.

## 5.3 Experimental Results

This section contains the outcome of pre-training and fine-tuning experiments. In addition, zero-shot transfer is investigated in the later part.

For each model, its mean rank (MR), mean reciprocal rank (MRR), and Hits at 10 (H@10) metrics on the test set are reported. I selected $N = 10$ for comparison, since it was the most consistently Hits@N metric reported in related work. I do report the Hits@N performance for other values of N, the validation set performance, the running time, and the best hyperparameters in Appendix B.

**Pre-training results.** The performance of the pre-trained models on OlpBench is given in Table 5.1. The introduced models obtain better scores than the previous best approach based on COMPLEX, however, I mainly attribute the improvement to the use of better KBC models.

**OKBC results.** The results of experiments on REVERB20K and REVERB45K are given in Table 5.2. The models strictly improve their performance when pre-trained on OLPBENCH. This improvement is particularly noticeable for NOEN-

| Model | Pre-trained? | ReVerb20K MR | ReVerb20K MRR | ReVerb20K H@10 | ReVerb45K MR | ReVerb45K MRR | ReVerb45K H@10 |
|---|---|---|---|---|---|---|---|
| NoEncoder_TuckER | no | 2611 | .196 | .267 | 5692 | .109 | .138 |
| | yes | *303* | *.379* | *.540* | *780* | *.299* | *.453* |
| NoEncoder_ConvE | no | 1419 | .282 | .380 | 2690 | .232 | .333 |
| | yes | *227* | *.400* | *.568* | *666* | *.345* | *.500* |
| NoEncoder_5⋆E | no | 2301 | .228 | .334 | 3460 | .152 | .212 |
| | yes | *780* | *.249* | *.363* | *3279* | *.189* | *.261* |
| GRU_TuckER | no | 581 | .364 | .505 | 1398 | .302 | .420 |
| | yes | *245* | *.397* | *.558* | *706* | *.331* | *.477* |
| GRU_ConvE | no | 334 | .387 | .540 | 824 | .343 | .488 |
| | yes | ***184*** | *.409* | *.573* | *600* | *.357* | *.509* |
| GRU_5⋆E | no | 395 | .390 | .546 | 836 | .357 | .508 |
| | yes | *202* | ***.417*** | ***.586*** | ***596*** | ***.382*** | ***.537*** |
| ♦ Okgit(ConvE) | yes▲ | 527 | .359 | .499 | 773.9 | .332 | .464 |
| † CaRe(ConvE, LAN) | no | 973 | .318 | .439 | 1308 | .324 | .456 |
| † TransE | no | 1426 | .126 | .299 | 2956 | .193 | .361 |
| † TransH | no | 1464 | .129 | .303 | 2998 | .194 | .362 |
| NoEncoder_TuckER$_{d=200}$ | no | 2855 | .184 | .248 | 5681 | .109 | .138 |
| | yes | 329 | .369 | .528 | 805 | .275 | .426 |
| BigGraph_TuckER$_{d=200}$ | yes▲ | 1907 | .215 | .291 | 2285 | .234 | .337 |

Table 5.2: Comparison of different models with and without pre-training on the OKBC benchmarks ReVerb20K and ReVerb45K. The scores of each model are reported with and without pre-training, with the better of the two written in *italics*. Separated from the rest with two lines are the previous best results on the datasets, and TuckER results with $d = 200$ for a fair BigGraph comparison. The best overall value in each column is written in **bold**. Results denoted with † and ♦ were taken from (Gupta et al., 2019) and (Chandrahas and Talukdar, 2021), respectively.

▲ Unlike all other models with a *yes* entry, BigGraph_TuckER and Okgit were not pre-trained on OlpBench, but on pre-trained WikiData embeddings and with the masked language modelling objective, respectively.

CODER models, which tend to overfit and achieve poor results without pre-training. However, when initialized with a pre-trained model, their ability to generalize improves. 5*E seems to be an exception to this, likely because there are no shared parameters between relations and entities, resulting in a weaker regularization. GRU-based models do not seem to suffer as severely from overfitting, but their performance still visibly improves when they are pre-trained on OLPBENCH.

Finally, the best introduced model outperforms the state-of-the-art approach by Chandrahas and Talukdar (2021) on REVERB20K and REVERB45K. Even when compared to pre-trained GRU_CONVE, which is based on the same KBC model, OKGIT(CONVE) and CARE(CONVE, LAN) lag behind. This is particularly surprising because the BERT and ROBERTA language models, used by OKGIT(CONVE), had received several orders of magnitude more pre-training on unstructured text, making the results of the introduced approach even more significant.

Similarly, the initialization of models with BIGGRAPH seems to help the performance, however, such an approach is in turn outperformed by a NOENCODER model, initialized with pre-trained encoders instead. This indicates that the suggested pre-training is much more efficient, despite the smaller computational cost.

**KBC results.** To evaluate the impact of pre-training on larger canonicalized knowledge bases, I compare the performance of models on FB15K237 and WN18RR. For brevity, I treat the choice of an encoder as a hyperparameter and report the better of the two models in Table 5.3. Detailed results are given in Appendix B.

Pre-trained models outperform their randomly initialized counterparts as well, however, the differences are usually smaller. There are several reasons that can explain the small difference in performance, primarily the difference in dataset size. The best-performing models on FB15K237 and WN18RR only made between 3 and 12 times more steps during pre-training than during fine-tuning. For comparison, this ratio was between 250 to 1000 for REVERB20K. The smaller improvements on FB15K237 and WN18RR can also be explained by the domain shift, as already described in Section 2.3.2.

Table 5.3 additionally includes multiple recently published implementations of CONVE, 5*E, and TUCKER, as well as other well-performing models in KBC.

| Model | pre-trained? | FB15K237 | | | WN18RR | | |
|---|---|---|---|---|---|---|---|
| | | MR | MRR | H@10 | MR | MRR | H@10 |
| TuckER | no | 166 | .358 | .545 | 4097 | *.468* | .528 |
| | yes | *151* | *.363* | *.550* | *3456* | .467 | *.529* |
| ConvE | no | 212 | .320 | .504 | 6455 | .429 | .479 |
| | yes | *200* | *.325* | *.510* | *5792* | *.435* | *.486* |
| 5⋆E | no | 152 | .353 | .539 | *2450* | *.492* | *.583* |
| | yes | ***143*** | *.357* | *.544* | 2636 | *.492* | .582 |
| [1] ConvE | no | 244 | .325 | .501 | 4187 | .43 | .52 |
| [2] ConvE | no | – | .339 | .536 | – | .442 | .504 |
| [3] TuckER | no | – | .358 | .544 | – | .470 | .526 |
| [4] 5⋆E | no | – | **.37** | **.56** | – | **.50** | **.59** |
| [5] KG-Bert | yes | 153 | – | .420 | **97** | – | .524 |
| [6] BoxE | no | 163 | .337 | .538 | 3207 | .451 | .541 |
| [7] ComplEx | no | – | **.37** | **.56** | – | .48 | .57 |
| [8] ComplEx-Dura | no | – | **.371** | **.560** | – | .491 | .571 |
| [8] Rescal-Dura | no | – | .368 | .550 | – | **.498** | .577 |

Table 5.3: Comparison of different models with and without pre-training on the KBC benchmarks FB15K237 and WN18RR. The scores of each model are reported with and without pre-training, with the better of the two written in *italics*. Separated from the rest with two lines, previous scores obtained with the ConvE, 5⋆E, and TuckER models are listed, followed by other well-performing models in the literature. The best overall result in each column is highlighted in **bold**.
[1](Dettmers et al., 2018); [2](Ruffinelli et al., 2020);[3](Balažević et al., 2019);[4](Nayyeri et al., 2021);[5](Yao et al., 2019); [6](Abboud et al., 2020);[7](Lacroix et al., 2018);[8](Zhang et al., 2020).

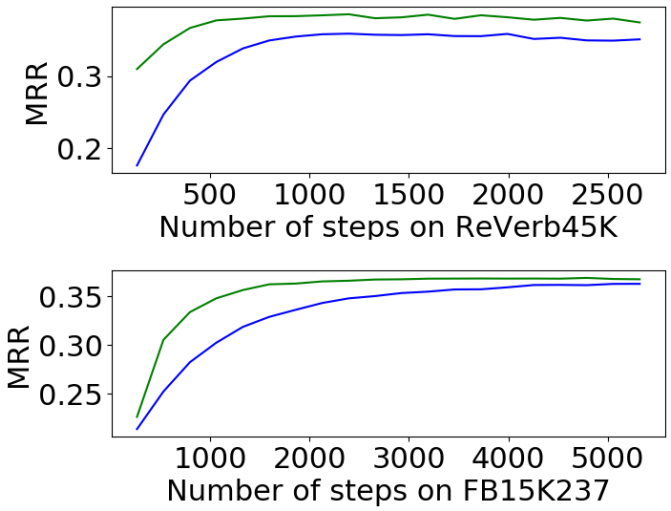



Figure 5.2: Comparing convergence of the best randomly initialized (blue) and pre-trained (green) models on ReVerb45k and Fb15k237. Pre-trained models converge in fewer training steps despite a smaller learning rate.

The comparison with all these models should be taken with reservation, as other reported models were often trained with a much larger hyperparameter space, as well as additional techniques for regularization (e.g., Dura (Zhang et al., 2020) or label smoothing (Balažević et al., 2019)) and sampling (e.g., self-adversarial sampling (Abboud et al., 2020)). Due to the large number of models and baselines, I could not expand the hyperparameter search without compromising the fair evaluation of all compared models.

In Appendix B, the best hyperparameters for each model are given. The pre-trained models usually obtain their best result with a larger dimension compared to their randomly-initialized counterparts. Pre-training thus serves as a type of regularization, allowing us to fine-tune larger models. For this reason, pre-training on even larger datasets and increasing the number of parameters is likely to result in even more significant improvements across many KBC datasets in future work.

Not only does pre-training allow training of larger models, the pre-trained models also require fewer training steps to obtain a similar performance, as seen in Figure 5.2. Even though the pre-trained models were fine-tuned with a smaller learning rate, they converge in fewer steps, which can be attributed to the pre-training. The experiments summarized in the figure were performed with the best

hyperparameter setup for both pre-trained and randomly initialized models.

Finally, I highlight that language modelling as pre-training hardly justifies the enormous computational cost. On the majority of metrics, KG-Bert performs worse than models with fewer parameters and less pre-training, with a notable exception of remarkable MR on the Wn18rr dataset.

### 5.3.1 Zero-Shot Experiments

| | | ReVerb20k | | | ReVerb45k | | |
|---|---|---|---|---|---|---|---|
| Model | Pre-trained? | MR | MRR | Hits@10 | MR | MRR | Hits@10 |
| NoEncoder_TuckER | no | 4862 | .001 | .002 | 8409 | .001 | .001 |
| | yes | *1486* | *.012* | *.026* | *1737* | *.019* | *.041* |
| NoEncoder_ConvE | no | 4600 | .002 | *.004* | 8464 | .001 | *.001* |
| | yes | *1480* | *.003* | .000 | *1962* | *.003* | *.001* |
| NoEncoder_5⋆E | no | 4795 | .001 | .000 | 8445 | .001 | .002 |
| | yes | *1422* | *.005* | .000 | *1701* | *.014* | *.011* |

Table 5.4: Comparison of the zero-shot performance of different models on the OKBC benchmarks ReVerb20k and ReVerb45k. The scores of each model are reported with and without pre-training on OlpBench, with the better of the two written in *italics*. Note that the model that was not pre-trained is equivalent to a random baseline.

To gain a better understanding of what kind of knowledge is transferred between the datasets, I investigate the zero-shot performance of pre-trained models on ReVerb20k and ReVerb45k, where the impact of pre-training was the strongest. If pre-training improvement was mainly due to the direct memorization of facts, the zero-shot performance should already be high without fine-tuning.

The results of the zero-shot evaluation are given in Table 5.4. Given is the performance of all models with and without pre-training, the latter being equivalent to a random baseline. Note that since no fine-tuning takes place, the choice of the encoder for evaluation does not matter. I thus chose to only evaluate one of them.

Observing the results, we see that pre-training the models results in a lower MR, but not necessarily in a much higher MRR. Even when MRR does improve, the difference is much smaller than when comparing fine-tuned models in Table 5.2. This implies that the improvement induced by pre-training likely does not happen

only due to the memorization of facts from OlpBench. On the other hand, the MR of pre-trained models is comparable or even better than the MR of randomly initialized NoEncoder models, fine-tuned on the ReVerb datasets, as reported in Table 5.2. Hence, pre-trained models carry a lot of "approximate knowledge", which is consistent with earlier remarks on pre-training serving as a type of regularization.

Knowing that the ReVerb20k and ReVerb45k test sets consist of facts which contain at least one previously unseen entity or relation, this experiment can be viewed as a test of out-of-distribution generalization. Comparing the zero-shot MRR results with the OlpBench results suggests that while OKBC models are capable of out-of-domain generalization to unseen entities and relations, there is still room for improvement.

## 5.4 Summary and Outlook

In this chapter, I have introduced a novel approach to transfer learning between various knowledge base completion datasets. The main strength of the introduced method is the ability to benefit from pre-training on uncanonicalized knowledge bases, constructed from facts that were collected from unstructured text. Scaling the introduced method up would let us train large-scale pre-trained models, which have already shown to be incredibly successful in NLP. The proposed method was tested on 5 different datasets, demonstrating that pre-training improves the performance of models. Pre-training turned out to be particularly beneficial on small-scale datasets, where the most significant gains were obtained, i.e. a 6% absolute increase of MRR and a 65% decrease of MR in comparison to the previously best method on ReVerb20k, despite not relying on large pre-trained models like Bert.

There are several directions of future work, including but not limited to scaling of pre-training to larger models and datasets, or investigating the impact of the encoder architecture. However, rather than focusing just on achieving better performance, it is important to understand what patterns such models are picking up. This question will, to some degree, be investigated in the next chapter, where multiple diagnostics of pre-trained KBC models are performed.

# Chapter 6

# Diagnostic Analysis of Pre-trained Models for Knowledge Base Completion

In this chapter, I introduce a novel dataset for the analysis of pre-trained models for Knowledge Base Completion (KBC), called DOGE (Diagnostics of Open knowledge Graph Embeddings). It consists of 6 subsets and is designed to measure multiple properties of a pre-trained model for KBC: robustness against synonyms, ability to perform deductive reasoning, presence of gender stereotypes, consistency with reverse relations, and coverage of different areas of general knowledge. This dataset was designed to help future researchers and engineers to forecast how using a pre-trained model could impact their own target knowledge base. Following the work done so far, special focus is put on the detection of gender bias.

Existing approaches to the analysis of KBC models often look at what structures these models can capture in theory and what mathematical properties they have (Abboud et al., 2020; Nayyeri et al., 2021). However, when dealing with pre-trained models, we are not only interested in what these models are theoretically capable of, but also what inference patterns they have actually picked up during pre-training and what knowledge they may introduce into our target knowledge base. As we will see in this chapter, there is a mismatch between the theoretical properties and practical behaviour.

The DOGE dataset is constructed manually by myself instead of crowdsourcing, to ensure a high degree of quality and correctness. After collecting the dataset, it is

used to evaluate the models obtained in Chapter 5. Somewhat surprisingly, I find that the behaviour of models on diagnostic datasets is not always consistent with the behaviour on other benchmarks, indicating that the existing benchmarks may not be rigorously testing the above-mentioned patterns. The DOGE dataset can thus serve as an additional way of testing OKBC models, providing more insight than just measuring overall performance.

The difference in the results compared to standard KBC datasets can be explained as a consequence of a different approach to dataset construction. As already discussed in Section 2.3.2, standard KBC datasets are usually obtained by randomly splitting an existing knowledge base. This tend to result in a train/test split where test facts can be very similar to train set, encouraging some level of overfitting. The latter behaviour can be observed in the process of training, where the MR on the validation set often starts decreasing by the time the best MRR is obtained, implying that the model is losing some of its ability to generalize. Since DOGE was constructed in a different way, it requires generalization to facts that may differ significantly from the train facts.

This chapter contains four sections. First one introduces the dataset and the metrics. Second section describes the experimental setup, with the results and their discussion reported in the third section. Finally, the fourth section summarizes the results and discusses their importance.

## 6.1 Dataset

The DOGE dataset, more precisely, a collection of diagnostic datasets, is aimed for the evaluation of pre-trained models for open knowledge base completion and measures the following properties:

- Coverage of different areas of general knowledge,
- Consistency when dealing with synonyms,
- Consistency when dealing with inverse relations,
- Deductive reasoning,

- Gender stereotypes,
- Impact of gender stereotypes on deductive reasoning.

There is a lot of diversity in the individual subsets of DOGE, some overlap while others are constructed in completely distinct ways. However, several common properties are shared by all of them. First, the evaluated model is always used to rank only a small set of possible answers rather than all entities in the knowledge base. This is to guarantee that only one answer is correct and to avoid noisy results. Second, all grammatical clues such as articles are either removed, or used in a way that does not automatically exclude any of the possible answers. Finally, English Wikipedia is strictly avoided as a resource because of its common presence in pre-training data or as a resource in dataset construction. More specific design choices and data properties will be presented in their respective sections.

### 6.1.1 Coverage of General Knowledge

When an individual is constructing a novel dataset for pre-training, they may want to know what type of knowledge the dataset contains and where collecting more data could be beneficial. To test this, I construct a collection of facts from the online version of Encyclopedia Britannica, 50 per each of its 12 categories:

- Entertainment & Pop Culture
- Geography & Travel
- Health & Medicine
- Lifestyles & Social Issues
- Literature
- Philosophy & Religion
- Politics, Law & Government
- Science
- Sports & Recreation

- Technology
- Visual Arts
- World History

Overall, this gives a dataset of 600 examples, designed to be used as a test set. Each example is a triplet $\langle h, r, t \rangle$ with a head or tail missing, e.g. $\langle$Nelson Mandela, first black president of, ?$\rangle$. In exactly half of the examples the missing entity is head, and half of the time it is the tail. 10 possible answers are provided, exactly one of them being correct. For the above example, the possible answers are *South Africa, United Nations, United Kingdom, United States of America, Japan, Nazi Germany, Vietnam, Cuba, Soviet Union, China*, with *South Africa* being the correct answer. The model should compute the score of each answer independently from alternative answers.

On Encyclopedia Britannica, each of the 12 above-mentioned categories is further split into more fine-grained subcategories. The *Science* category, for example, is further split into physics, mathematics, biology, chemistry, etc. When sampling data, all subcategories are equally represented in a category with a difference of up to 1 example due to rounding. To ensure that the collected data are representative of general knowledge and not obscure facts, source articles are always taken from the *Featured Articles* section of each subcategory. At most one fact is collected from each article to get a greater variety of samples. While the manually selected fact is usually extracted directly from the text as if obtained with an open information extraction system, I occasionally had to make minor changes and rewriting to ensure clarity and to remove ambiguity. Moreover, all negative answers are taken strictly from the same article or articles from the same subcategory to ensure their relevancy.

Results and scores are reported only per-category and *not* for the entire dataset. This is because the representation of different areas of knowledge can be imbalanced. Due to the above described choices in sampling, the subcategory of *Baseball* has more representatives than *The Middle Ages*, for example. Even though I provide the subcategory of each example, I only report results over full categories as subcategories may have too few examples to make the results conclusive. Note

that the dataset has some Western, particularly American bias, as Encyclopedia Britannica more often featured articles about USA and Western Europe. Since pre-training and all experiments are conducted in English and collected from English sources, this does not present a problem. However, it is important to be aware of this if the dataset is used in a different cultural context.

### 6.1.2 Robustness to Synonyms

Robustness to synonyms is an important feature of any OKBC system because, in principle, handling uncanonicalized data requires a model to recognize which entities in the knowledge base correspond to the same real-world entity. To measure this, I manually find all examples from the *general knowledge* part of DOGE where one of the entities or a relation can be replaced with a synonym. In this way, I obtain 78 instances where one of the entities can be swapped with a synonym, and 361 instances where the relation can be replaced with a synonym. Personal names are never replaced (e.g. *Nelson Mandela* into *Mandela*) because the transformations are usually trivial. For each of these examples, a *twin* example is provided where said relation or entity is swapped for its synonym. The Oxford English Dictionary was used as the source of synonyms.

Let $r_a$ be the rank of the correct answer on some original instance, and $r_b$ be the rank on its *twin* instance. We are interested in the change between the two, $r_a - r_b$, which should be 0 if the answer is unchanged. I report the overall mean and variance of random variable $r_a - r_b$. The smaller the variance, the more robust against synonyms the model is. The mean is reported only to control for possible noise introduced by the annotation. If the value of the mean strongly deviates from 0, one may conclude that *twin* instances are on average more difficult or easier than the originals, making the results inconclusive. As will be seen during the evaluation, the absolute value of the mean was always below 1 and usually close to 0, indicating a high quality of data samples.

### 6.1.3 Robustness to Inverse Relations

Consistency in the case of inverse relations is another desirable feature of an OKBC model. More specifically, given two test instances $\langle h, r, ? \rangle$, $\langle ?, r^{-1}, h \rangle$ and the same

set of possible answers, it is desirable that the model gives the same answer in both cases. To test this, 520 examples from the *general knowledge* part of DOGE were identified where an inverse relation can be easily found. For each of them, a *twin* instance with an inverse relation was created, as described above. The example ⟨Nelson Mandela, first black president of, ?⟩ has a twin instance ⟨ ?, has the first black president, Nelson Mandela⟩. Note that both the correct answer and the set of alternative answers remains the same.

Similarly as for the synonym test, I report the mean and standard deviation of $r_a - r_b$, where $r_a$ is the rank of the correct answer on the original example and $r_b$ is the rank of the same answer in the *twin* example. The interpretation of the results follows the same logic: The smaller the deviation, the better the model. The mean is reported only for quality control. As demonstrated later at the time of evaluation, the absolute value of the mean is always below 0.5 and often close to 0, indicating that a possible deviation is not caused by poor annotation.

### 6.1.4 Deductive Reasoning

The ability to apply collected *general knowledge* to specific instances during fine-tuning is one of the most desirable properties for a system to have. If our target knowledge base contains a fact that ⟨Mary, visited, Broadway⟩ and our pre-trained model can predict that ⟨Broadway, located in, Manhattan⟩, then it should also include ⟨Mary, visited, Manhattan⟩ into the target knowledge base. This is an instance of deductive reasoning, that is, applying general knowledge to specific instances.

The opposite scenario, inductive reasoning, cannot be easily tested for pre-trained KBC models. It is hard to find out whether the captured *general knowledge* has been learned explicitly because it appeared in the training data, or implicitly by generalizing from numerous specific instances.

For 70 facts $\langle h, r, t\rangle$ from the *general knowledge* set of the DOGE dataset, I create a *deductive variant*. Suppose that tail $t$ in the *general knowledge* triplet $\langle h, r, t\rangle$ has to be predicted. I additionally create facts $\langle X, p, t\rangle$ and $\langle X, q, h\rangle$, where $X$ denotes a personal name and $\langle X, q, h\rangle \wedge \langle h, r, t\rangle \implies \langle X, p, t\rangle$. In the example above, the model is tested whether it can predict the tail of the triplet ⟨Mary,

visited, Manhattan⟩ after it has been additionally provided with the fact ⟨Mary, visited, Broadway⟩. This prediction can only be done successfully if the model combines the newly given fact with its background knowledge that ⟨Broadway, part of, Manhattan⟩. Personal names were randomly selected from the 100 most common male and female names in the USA in the last 100 years.[1]

More specifically, a model is evaluated on three different setups. First, it is evaluated on the relevant subset of the *general knowledge* dataset (examples $\langle h, r, t \rangle$), testing whether it contains the relevant background knowledge. The score obtained on this set serves as a soft upper bound on the score the model can obtain on the deductive reasoning test; a model in principle should not be able to do deductive reasoning without the relevant background knowledge. Second, the model is evaluated on the test $\langle X, p, t \rangle$ without seeing $\langle X, q, h \rangle$, an instance that it cannot answer due to the lack of information. Scores obtained here serve as a soft lower bound on the final score, and aim to detect potential noise such as annotation artefacts. Finally, in the third experiment, the model is given facts $\langle X, q, h \rangle$ and evaluated on $\langle X, p, t \rangle$. In my experimental setup, the addition of facts to the model was done through fine-tuning; a more detailed description is given in the experimental section.

For triplets $\langle h, r, t \rangle$ where the head was missing, the data construction procedure is analogous. The presented setup is not the only way in which deductive reasoning could be tested and more rigorous probings could be made. I leave such analyses to future work.

### 6.1.5 Gender Stereotypes

It may be undesirable if harmful societal stereotypes are added to the target knowledge base. To detect the presence of potential historical gender bias and stereotypes, I test to what degree the model associates gender-imbalanced occupations with typically feminine and masculine names. Following Rudinger et al. (2018) and Zhao et al. (2018), I collected 50 highly gender-imbalanced jobs based on US

[1] Taken from `https://www.ssa.gov/OACT/babynames/decades/century.html` on 2 March 2021.

occupation statistics,[2] half stereotypically female and half stereotypically male. By randomly sampling from the list of names, already introduced in the previous section, I generated 200 stereotypical and 200 anti-stereotypical examples of type $\langle X, \text{is}, Y \rangle$, where $X$ is a name and $Y$ is an occupation. Unlike in datasets from previous sections where 10 possible answers were given, the model has to select between all 50 occupations.

It is important to note that while a gold answer does exist, it is impossible to deduce it due to the lack of context. The models are therefore expected to obtain scores similar to ones of a random baseline. Rather than to evaluate correctness, this dataset was created to detect possible associations between gendered names and gender-imbalanced occupations, i.e. detect the presence of conceptual gender. Whether such association is desirable or not wholly depends on the application. One could claim that the detected stereotypes are a reflection of the real world and therefore can improve the accuracy of predictions. In the absence of counter-evidence, connecting a name with an occupation that is often associated with that name's gender is statistically indeed more likely to be correct. On the other hand, the mere presence of such stereotypes in applications that should be stereotype-agnostic (e.g. law) could present a severe problem.

### 6.1.6 Impact of Stereotypes on Deductive Reasoning

It can be statistically acceptable that $\langle \text{Mary}, \text{is}, ? \rangle$ is more likely answered as *childcare worker* than *car mechanic* when no additional information is given. However, if the target knowledge base contains a triplet $\langle \text{Mary}, \text{repairs}, \text{cars} \rangle$, labelling *Mary* to be a *childcare worker* is not just biased, but also incorrect. To detect how predictions of pre-trained OKBC models on such examples change in the presence of additional background information, I provide 200 facts about occupations, 4 per occupation.[3] These are then used as additional inputs to the model.

For each test triplet $\langle X, \text{is}, Y \rangle$, where $X$ is a personal name and $Y$ is an occupation, two additional examples are constructed. Two true facts about occupation

[2] Taken from the bureau of labor statistics, 2 March 2021 `https://www.bls.gov/cps/cpsaat11.htm`

[3] Oxford English Dictionary and Career Explorer were used as the source of these facts. `https://www.careerexplorer.com/`

$Y$ are provided, e.g. ⟨car mechanic, repairs, cars⟩ and ⟨car mechanic, diagnoses, malfunctioning cars⟩, with the occupation swapped out for the name (⟨Mary, repairs, cars⟩). One of them is added to the train set, and one to the validation set. Moreover, all 200 general facts about occupations are also stored as a test set for a separate experiment.

This provides us with a similar setup as in the case of general deductive reasoning. Firstly, we can investigate how a model behaves when it has to guess the profession of an individual, as described in the previous section. Secondly, we can test its general background knowledge about occupations. Finally, we can find out how well the model determines the individuals' occupations when additionally provided with knowledge about what these individuals do. Again, we expect this score to be bound by the first two results. I additionally split this score into the *stereotypically feminine*, *stereotypically masculine*, *anti-stereotypically feminine*, and *anti-stereotypically masculine* sets to observe the possible difference between genders.

Testing a model on this dataset allows us to investigate to what degree the gender stereotypes impact the ability of the model to perform deductive reasoning. In other words, one can investigate to what degree the model overcomes stereotypes in the presence of counter-evidence.

## 6.2 Experiments

In this section, I compare various models from Chapter 5 on the DOGE dataset. Since pre-trained models with a larger dimension of embeddings have generally proved to be more successful, I only perform experiments with the largest variant of each model. For the majority of the datasets, the experiments should be conducted in a zero-shot setting. There, I follow the same setup as in Section 5.3.1 with all models. All KBC models were tested only with GRU encoders, since the choice of an encoder makes no difference in a zero-shot setting.

The datasets that test a models' ability to perform deductive reasoning require additional training. On these datasets, each KBC model was evaluated both with a GRU and NOENCODER setup in the same way as in Chapter 5. When evaluating the model on the DOGE gender stereotype dataset, the best hyperparameter setup

is found on the training and validation set. The learning rate is taken from $\{3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}\}$. The dropout rate for ConvE and TuckER is taken from $\{0.2, 0.3\}$ and $\{0.3, 0.4\}$, respectively. The weight of N3 regularization for 5*E is taken from $\{1.0, 0.3\}$. The same setup was then used for fine-tuning on the train set of the deductive reasoning dataset, where a validation set was not constructed.

**Impact of GloVe Embeddings.** When pre-training on the OlpBench dataset, word embeddings were initialized from GloVe vectors (Pennington et al., 2014). Unfortunately, such word embeddings are known to contain undesired gender stereotypes (Bolukbasi et al., 2016). To understand their impact on potentially biased behaviour of pre-trained KBC models, I additionally pre-train Gru_ConvE and Gru_TuckER models with randomly initialized word embeddings. They were pre-trained on OlpBench with the same hyperparameter setup as their GloVe-initialized counterparts. I did not repeat this experiment with a 5*E model due to larger computational cost of such an experiment.

## 6.3 Experimental Results

This chapter contains the results of all evaluated models on all parts of the Doge dataset. Results are reported in MR, MRR, and Hits@1 metrics, except for parts of the dataset that measure consistency, for which the mean and standard deviation are reported instead. Hits@N metrics for other values of N were not reported due to the small number of possible answers (10 or 50).

### 6.3.1 General Knowledge

All three pre-trained KBC architectures are compared on the general knowledge part of the Doge dataset, with results reported in Table 6.1. It is evident that the models are fairly consistent with each other w.r.t. how well they do in different categories. This is unsurprising since they were all trained on OlpBench. Nevertheless, it also means that it can be difficult to conclude which of the models handles what type of data better. ConvE seems to often outperform other models in most categories, despite often ranking the lowest of the three models on commonly used KBC benchmarks.

Inconsistent results between DOGE and most other datasets can be explained by the difference in their construction. Unlike other KBC datasets, DOGE was not obtained by randomly splitting an existing knowledge base. Models trained on OLPBENCH and evaluated on DOGE face a shift of data distribution, requiring better generalization capabilities. This is the first result of this kind, indicating that existing KBC benchmarks only cover a specific scenario which may incorrectly reward models with lower generalization capabilities.

| | TUCKER | | | CONVE | | | 5*E | | |
|---|---|---|---|---|---|---|---|---|---|
| category | MR | MRR | H@1 | MR | MRR | H@1 | MR | MRR | H@1 |
| Entertainment & Pop Culture | 3.74 | .515 | .36 | 3.06 | .593 | .44 | 3.42 | .552 | .40 |
| Geography & Travel | 2.26 | .657 | .46 | 2.26 | .714 | .58 | 2.36 | .692 | .54 |
| Health & Medicine | 3.36 | .541 | .36 | 2.24 | .669 | .48 | 3.14 | .556 | .38 |
| Lifestyle & Social Issues | 2.76 | .556 | .32 | 2.90 | .584 | .40 | 3.34 | .538 | .34 |
| Literature | 3.70 | .518 | .36 | 2.96 | .576 | .38 | 3.94 | .458 | .28 |
| Philosophy & Religion | 2.92 | .579 | .40 | 2.72 | .634 | .46 | 2.92 | .611 | .44 |
| Politics, Law & Government | 2.70 | .634 | .46 | 2.54 | .667 | .52 | 3.00 | .604 | .44 |
| Science | 3.82 | .467 | .26 | 3.84 | .525 | .38 | 3.66 | .553 | .42 |
| Sports & Recreation | 3.52 | .521 | .34 | 3.14 | .538 | .36 | 3.44 | .494 | .28 |
| Technology | 2.96 | .636 | .50 | 3.00 | .633 | .50 | 2.94 | .672 | .56 |
| Visual Arts | 4.30 | .483 | .30 | 4.28 | .440 | .26 | 3.68 | .538 | .36 |
| World History | 3.08 | .565 | .36 | 3.14 | .608 | .46 | 3.04 | .615 | .46 |

Table 6.1: Comparison of the three pre-trained architectures on the general knowledge subset of the DOGE dataset. Each column contains scores of a model on different categories, as defined by Encyclopedia Britannica.

### 6.3.2 Model Consistency

All three KBC pre-trained architectures are compared on their ability to make consistent prediction in the presence of synonyms and inverse relations, with results reported in Table 6.2. None of the models are particularly consistent on either of the tests, with CONVE being slightly better than 5*E and TUCKER. This improvement in consistency might be due to the generally better performance of CONVE on DOGE rather than due to its design. Overall, it is safe to conclude that none of the tested models can be called *consistent*, which is unfortunately common

for neural network models and has already been observed when testing neural approaches on other tasks, e.g. the Winograd Schema Challenge in Chapter 3.

| | TuckER | | ConvE | | 5*E | |
|---|---|---|---|---|---|---|
| | mean | stdev | mean | stdev | mean | stdev |
| Entity Synonyms | 0.628 | 2.712 | 0.731 | 2.552 | 0.628 | 2.820 |
| Relation Synonyms | 0.235 | 2.402 | −0.033 | 1.492 | 0.499 | 2.100 |
| Inverse Relations | 0.196 | 2.292 | 0.117 | 1.729 | −0.031 | 2.141 |

Table 6.2: A comparison of three pre-trained architectures on subsets of the Doge dataset that probe the consistency of predictions. Mean values are usually close to zero, indicating that the deviation was not caused by poor annotation.

### 6.3.3 Deductive Reasoning

In this section, I report the results on the test of deductive reasoning. The scores obtained by all three evaluated models are given in table 6.3. All models achieve approximately chance performance on the *No Added Facts* experiments, indicating that the answers of this subset cannot be deduced through undesired clues. Moreover, all models achieve a better-than-random guessing score on the *Background Knowledge* experiments, indicating that they do contain the relevant background knowledge for many instances. If the results of the models on this set of instances were poor, the results from this whole section would be inconclusive. Finally, the results on *With Added Facts* lie somewhere in between the two, as expected. This indicates that the models can indeed combine newly obtained facts with their background knowledge, however, a big gap to *Background Knowledge* results indicates that there is a lot of room for improvement.

The impact of the encoder varies by KBC model. For ConvE and TuckER there is no conclusive answer to which encoder is the better choice as it depends on the choice of metric. NoEncoder versions obtain better MR, while Gru versions obtain better MRR and Hits@1 scores. For 5*E, NoEncoder is visibly the superior choice, making the NoEncoder_5*E the best model for deductive reasoning out of all the compared setups. The reason behind this likely lies in its shallow architecture. Since there are no shared parameters or stacked layers,

training primarily affects the embeddings of entities in the training data while preserving the structure of everything else.

| | TuckER | | | ConvE | | | 5*E | | |
|---|---|---|---|---|---|---|---|---|---|
| | MR | MRR | H@1 | MR | MRR | H@1 | MR | MRR | H@1 |
| Background Knowledge | 3.26 | .556 | .373 | 3.01 | .598 | .435 | 3.24 | .574 | .408 |
| No Added Facts | 5.13 | .299 | .087 | 4.86 | .343 | .130 | 4.91 | .339 | .145 |
| With Added Facts$_{\text{Gru}}$ | 4.35 | .383 | .159 | 4.16 | .427 | .217 | 4.19 | .426 | .217 |
| With Added Facts$_{\text{NoEnc.}}$ | 4.09 | .372 | .116 | 4.06 | .411 | .174 | 3.90 | .485 | .319 |

Table 6.3: Comparison of the three pre-trained architectures on the deductive reasoning subset of the Doge dataset. *Background Knowledge* and *No Added Facts* results serve as soft upper and lower bounds to the *With Added Facts* results, which can be correctly deduced only by combining newly added facts with the relevant background knowledge. Fine-tuning is done both with Gru and NoEncoder setup to compare the impact of the two.

### 6.3.4 Gender Stereotypes

Finally, all deductive reasoning tests are repeated on the subset of Doge that detects gender stereotypes. The results are given in Table 6.4 and are reported separately for each gender and separately for stereotypical and anti-stereotypical occupations. Finally, I test whether the model predictions are affected by the stereotypes indicating the presence of historical bias. To see whether the difference in performance is statistically significant, I use the Wilcoxon signed-rank test. More specifically, I test whether swapping the name of a gender that is not stereotypically associated with an occupation for a name of the opposite gender impacts the rank of the correct answer.

The gender stereotypes are strongly present, resulting in a model being correct more often when the gender of the name matches the conceptual gender of the occupation. By comparing the results across all examples, both stereotypical and anti-stereotypical, we can see that the masculine names are on average ranked higher, likely due to representation bias in the training data. Moreover, all models obtain decent scores on background knowledge about occupations, making deductive reasoning experiments conclusive.

| No Added Facts | TuckER MR | TuckER MRR | TuckER H@1 | ConvE MR | ConvE MRR | ConvE H@1 | 5*E MR | 5*E MRR | 5*E H@1 |
|---|---|---|---|---|---|---|---|---|---|
| St Masculine | 22.51 | .096 | .02 | 24.27 | .083 | .02 | 22.9 | .096 | .01 |
| St Feminine | 21.07 | .093 | .00 | 18.45 | .138 | .02 | 20.95 | .119 | .04 |
| Anti-St Masculine | 29.96 | .079 | .03 | 32.18 | .037 | .00 | 30.32 | .066 | .01 |
| Anti-St Feminine | 29.21 | .062 | .00 | 28.41 | .0642 | .00 | 28.11 | .072 | .02 |
| p-value | $6.04 \cdot 10^{-15}$ | | | $2.19 \cdot 10^{-15}$ | | | $1.48 \cdot 10^{-11}$ | | |
| Background Knowledge | 9.575 | .417 | .275 | 7.910 | .4307 | .265 | 7.155 | .503 | .345 |

| With Added Facts$_{\text{Gru}}$ | TuckER MR | TuckER MRR | TuckER H@1 | ConvE MR | ConvE MRR | ConvE H@1 | 5*E MR | 5*E MRR | 5*E H@1 |
|---|---|---|---|---|---|---|---|---|---|
| St Masculine | 18.09 | .154 | .08 | 18.69 | .176 | .09 | 14.58 | .184 | .04 |
| St Feminine | 20.61 | .138 | .03 | 18.22 | .224 | .14 | 15.33 | .175 | .04 |
| Anti-St Masculine | 24.37 | .084 | .01 | 25.22 | .079 | .00 | 19.16 | .198 | .10 |
| Anti-St Feminine | 26.83 | .109 | .03 | 25.63 | .098 | .03 | 21.36 | .119 | .03 |
| p-value | $7.20 \cdot 10^{-9}$ | | | $2.62 \cdot 10^{-12}$ | | | $8.87 \cdot 10^{-6}$ | | |

| With Added Facts$_{\text{NoEnc.}}$ | TuckER MR | TuckER MRR | TuckER H@1 | ConvE MR | ConvE MRR | ConvE H@1 | 5*E MR | 5*E MRR | 5*E H@1 |
|---|---|---|---|---|---|---|---|---|---|
| St Masculine | 16.68 | .168 | .07 | 15.68 | .210 | .07 | 21.72 | .114 | .02 |
| St Feminine | 20.23 | .125 | .00 | 16.20 | .210 | .08 | 22.05 | .111 | .03 |
| Anti-St Masculine | 23.10 | .098 | .03 | 21.36 | .099 | .01 | 28.47 | .073 | .01 |
| Anti-St Feminine | 27.57 | .092 | .03 | 22.58 | .1343 | .05 | 26.55 | .073 | .01 |
| p-value | $7.03 \cdot 10^{-13}$ | | | $3.81 \cdot 10^{-10}$ | | | $5.43 \cdot 10^{-9}$ | | |

Table 6.4: Comparison of three pre-trained architectures on the gender-bias detection subset of Doge. The dataset is split into Stereotypical (St) and Anti-Stereotypical (Anti-St) examples. The first of the three tables contains information on the experiments that do not require fine-tuning: stereotype detection and background knowledge. The second and the third table contain information on deductive reasoning about gender for Gru and NoEncoder encoders, respectively.

The observed patterns persist when additional relevant data is added to the models, regardless of the choice of the encoder. It is worth noting that the improvement from the additional knowledge about entities only moderately helps the models regardless of the gender or stereotypes. Connecting this with the results on deductive reasoning, we can conclude that models contain gender stereotypes and are unable to overcome them in the presence of additional data. The reason why they cannot overcome them lies in their general poor ability to perform deductive

reasoning rather than a strong presence of stereotypes. Unfortunately, this means that all analyzed pre-trained KBC models can introduce gender-stereotyped facts into the target knowledge base, even in presence of counter-evidence.

### 6.3.5 The Impact of Word Embeddings

Finally, I report the results of re-training ConvE and TuckER on OlpBench without GloVe word embeddings. The results on OlpBench are reported in Table 6.5. Evidently, the absence of GloVe vectors significantly decreases the performance of both models.

| | MR | MRR | H@10 |
|---|---|---|---|
| Gru_TuckER$_{\text{GloVe}}$ | *57.2K* | *.053* | *.097* |
| Gru_TuckER$_{\text{No_GloVe}}$ | 65.1$K$ | .042 | .077 |
| Gru_ConvE$_{\text{GloVe}}$ | *57.2K* | *.045* | *.086* |
| Gru_ConvE$_{\text{No_GloVe}}$ | 69.2$K$ | .030 | .057 |

Table 6.5: Comparison of GloVe impact on OlpBench. TuckER and ConvE are both trained with GloVe and with randomly-initialized word embeddings. The better of the two in each column is written in *italics*.

To see whether models without GloVe embeddings are less susceptible to gender stereotypes, I evaluate them on the gender stereotype subset of the Doge dataset. The results can be found in Table 6.6. Despite the slightly lower performance, the general trends in the results of these experiments are very similar to the results with the GloVe embeddings in Table 6.4. This indicates that GloVe embeddings definitely are not the only source of gender stereotypes and while their removal results in an undesired performance drop, it does not decrease the biased behaviour.

## 6.4 Summary and Discussion

In this chapter, I introduced Doge, a novel diagnostic dataset for the analysis of pre-trained models for knowledge base completion. I used it to evaluate pre-trained models introduced in Chapter 5. The results indicated that the existing pre-trained models lack robustness against synonyms and inverse relations and

| No Added Facts | TuckER | | | ConvE | | |
|---|---|---|---|---|---|---|
| | MR | MRR | H@1 | MR | MRR | H@1 |
| St Masculine | 23.34 | .071 | .00 | 23.32 | .109 | .03 |
| St Feminine | 20.56 | .1396 | .05 | 23.40 | .113 | .03 |
| Anti-St Masculine | 31.14 | .042 | .00 | 27.80 | .054 | .00 |
| Anti-St Feminine | 27.97 | .104 | .05 | 27.70 | .070 | .00 |
| p-value | $3.03 \cdot 10^{-16}$ | | | $1.67 \cdot 10^{-12}$ | | |
| Background Knowledge | 10.19 | .342 | .200 | 10.70 | .338 | .195 |

| With Added Facts$_{\text{Gru}}$ | TuckER | | | ConvE | | |
|---|---|---|---|---|---|---|
| | MR | MRR | H@1 | MR | MRR | H@1 |
| St Masculine | 19.54 | .113 | .01 | 19.12 | .170 | .07 |
| St Feminine | 21.21 | .132 | .04 | 21.94 | .149 | .05 |
| Anti-St Masculine | 25.47 | .078 | .02 | 23.39 | .099 | .02 |
| Anti-St Feminine | 26.80 | .113 | .04 | 26.23 | .088 | .02 |
| p-value | $9.59 \cdot 10^{-10}$ | | | $3.51 \cdot 10^{-07}$ | | |

Table 6.6: Comparison of pre-trained ConvE and TuckER architectures without GloVe embeddings on the gender-bias detection subset of Doge. The dataset is split into a Stereotypical (St) and Anti-Stereotypical (Anti-St) examples. The first of the two tables contains information on experiments that do not require fine-tuning: stereotype detection and background knowledge. The second of the two tables contains the information on deductive reasoning about gender for Gru encoders. Fine-tuning with the NoEncoder setup was omitted for brevity.

strongly rely on gender stereotypes. On a more positive note, they seem to cover all major areas of knowledge decently. The main highlighted problem with the existing pre-trained KBC models is the persistence of harmful stereotypes, which the model cannot remove even with the introduction of counter-evidence. It may thus happen that a pre-trained model includes an incorrect fact ⟨Mary, is, childcare worker⟩ into the target knowledge base even if the knowledge base contains the fact ⟨Mary, repairs, cars⟩.

The observation made in Chapter 4 about models getting better bias scores as their performance improves applies to this dataset as well. Better-trained models are likely to obtain better consistency scores, even if no other steps to improve their consistency were taken. They are simply more likely to get the answer right. This does not necessarily apply to other tests in Doge, though. A KBC model

can contain all the world knowledge and still not be able to deal with newly-added data, that is, perform deductive reasoning. Diagnostic subsets of DOGE based on truly ambiguous cases and newly introduced data samples may thus prove useful even when the pre-training is scaled up.

# Chapter 7

# Conclusion

In this thesis, I have discussed and worked on two important topics, both active areas of research with important prospects for the future. Firstly, I analysed the impact of pre-training, showing the importance of unsupervised pre-training tailored specifically to the target task. On two tasks, coreference resolution and knowledge base completion, my introduced approaches outperformed general-purpose masked language modelling alone. The difference was particularly stark in the case of open knowledge base completion, where the introduced models outperformed BERT with many fewer parameters and considerably less pre-training.

The second topic that I addressed are societal biases, which are picked up by these models during pre-training. Pre-trained natural language processing models appear in everyday applications, e.g., search engines. As such, the long-term negative impact of biases against some groups of people can be colossal. It is thus crucial to understand where these models fail and who is affected. In my work, I first introduced a method for dealing with imbalanced test sets, and designed a general-purpose dataset for diagnostics in open knowledge base completion, where there were no existing diagnostic datasets before.

For unsupervised pre-training, the future developments may look quite straightforward. With the introduction of larger and better pre-trained transformer models, the expressivity of a model is rarely the bottleneck. Significant improvements are no longer obtained through minor changes in the architecture of a model, thus putting more focus on the data and training regime. Task-specific pre-training focuses mostly on the former, and includes providing higher-quality data that does

not have to be manually labelled. It is limited by two things: structural properties of text that can be exploited to construct data similar to the target task, and the ability of researchers to find and take advantage of them. Since the amount of readily-available online text grows quickly while manual annotation remains expensive, one can only expect that the interest in unsupervised methods will continue to grow.

Two bigger challenges lie ahead of bias detection and mitigation – one political and one technical. The political challenge is to keep emphasizing the need for bias mitigation, particularly in the industry. Since putting effort into fairness often does not result in quick performance improvements or revenue, it can seem like a nuisance. However, this is not a challenge specific to natural language processing. Instead, it is part of a political movement, larger than just the computer science research community.

A more technical obstacle that the field will have to overcome are the limitations of diagnostic benchmarks that focus only on unambiguous examples. As noted in Chapters 4 and 6, more pre-trained and better-performing models tend to obtain better bias scores. If an amazingly-performing model is evaluated on data where the answer can always be determined unambiguously, its internal biases against the examples of a protected class may be cloaked by its ability to give the correct answer. This leaves the impression that the model is not biased. Hiding biases by improving the performance of a model can be acceptable on tasks where truly ambiguous or open-ended examples are indeed rare, e.g. coreference resolution. On such tasks, it can be possible to decrease the biased performance of a model by adding more pre-training data.

However, natural language processing tasks, particularly those that require language generation, do not always have an unambiguous answer, with open-ended and ambiguous examples appearing commonly. It is for those tasks that better diagnostic tests based on behavioural testing and open-ended examples have to be constructed. Designing such tests is difficult and was so far often avoided. Theoretical definitions on ambiguous examples are often conflicting and defining the *acceptable* behaviour can be tricky as a consequence. A diagnostic test without a firm theoretical background and a good motivation is unlikely to see widespread

use. As a result, the majority of existing datasets focus on the simple but insufficient scenario, where there is obviously only one correct answer. To make the bias-detecting tests stronger and long-lasting, the research community will have to move to tests with open-ended questions, even if that requires a better theoretical justification and motivation.

I expect a strong future development in both of these two research problems. With the introduction of two task-specific pre-training approaches, an algorithm for tackling imbalanced test sets, and a novel diagnostic dataset for open knowledge base completion, this thesis provides a strong foundation for the future development of both of them.

# Appendix A

# Annotated WikiCREM Examples

The appendix contains the 100 manually annotated examples of WikiCREM.

1. Throughout training camp , Jackson competed to be the Bengals ' third cornerback on the depth chart against Darqueze Dennard . On August 2 , 2016 , it was announced that `[MASK]` had suffered a torn pectoral muscle and would have to undergo surgery .
   **ambiguous**
   Pronoun in place of `[MASK]`?: **No**
   Annotator's answer: **N/A**
   Correct?: **N/A**

2. The Ark " consisted of a giant rowboat with a small engine which Beek used as his first ferry vessel . " The `[MASK]` " carried oars in the event of engine failure .
   **not ambiguous**
   Pronoun in place of `[MASK]`?: **no**
   Annotator's answer: **Ark**
   Correct?: **yes**

3. However , John was able to gain the lost estates by a marriage to Joanna of Urgell , granddaughter of Peter IV of Aragon . `[MASK]` fought with Aragon against Castile , but helped his brother Peter , Cardinal of Foix and Arles , to crush insurgents from Aragon .
   **not ambiguous**

Pronoun in place of [MASK]?: **yes**
Annotator's answer: **John**
Correct?: **yes**

4. Ultravox had gone on to greater success with Midge Ure fronting the band , but when Simon left the band in 1988 , Billy Currie formed a new band which later included [MASK] .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Midge Ure**
Correct?: **no**

5. The poem describes the poet 's idyllic family life with his own three daughters , Alice , Edith , and Anne Allegra : " grave [MASK] , and laughing Allegra , and Edith with golden hair . "
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Alice**
Correct?: **yes**

6. Koch and Eide searched Cho 's belongings and found a pocket knife , but they did not find any items that they deemed threatening . [MASK] also described a telephone call that he received from Cho during the Thanksgiving holiday break from school .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

7. As Rajveer was able to successfully lead the escape of them both , Harleen now entrusts Rajveer , subsequently falling in love with him . When Rajveer goes out of his hotel with [MASK] , he sees that they are wanted by Interpol .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**

Annotator's answer: **Harleen**
Correct?: **yes**

8. Elmas , Su Masu in Sardinian language , is a " comune " of the Metropolitan City of Cagliari in the Italian region of Sardinia , located about northwest of Cagliari . Until 1989 `[MASK]` was a district of Cagliari .
   **not ambiguous**
   Pronoun in place of `[MASK]`?: **yes**
   Annotator's answer: **Elmas**
   Correct?: **yes**

9. Later in the year , Li Keyong did send Li Sizhao and Zhou to capture Xi and Ci Prefectures , which had become under Zhu 's control when `[MASK]` conquered Huguo earlier in 901 .
   **not ambiguous**
   Pronoun in place of `[MASK]`?: **yes**
   Annotator's answer: **Zhu**
   Correct?: **yes**

10. Elisha told Hazael to tell Hadadezer that he would recover , and he revealed to Hazael that the king would recover but would die of other means . The day after he returned to Hadadezer in Damascus , `[MASK]` suffocated him and seized power himself .
    **not ambiguous**
    Pronoun in place of `[MASK]`?: **yes**
    Annotator's answer: **Hazael**
    Correct?: **yes**

11. In 1946 , Hill and Knowlton dissolved their partnership , and Knowlton took over the direction of Hill & Knowlton Cleveland , which closed shortly after Knowlton's retirement in 1962 . `[MASK]` also maintained an interest in music .
    **ambiguous**
    Pronoun in place of `[MASK]`?: **no**

Annotator's answer: **N/A**
Correct?: **N/A**

12. First , God is revealed with Law , and secondly , God is revealed as Person . [MASK] 's anger at Moses for not speaking to the rock on the second occasion , highlights that this is not the spiritual picture He wanted portrayed .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **God**
Correct?: **yes**

13. He now said he had seen Acreman follow Cheryl Fergeson up a staircase leading to the auditorium and then heard her scream , " No " and " Don 't . " Later that day , [MASK] warned Sessum not to tell anyone what he had seen .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Acreman**
Correct?: **yes**

14. Meidi finally figures it out , but does not reveal to Qi Yue and Ah Meng until [MASK] confesses .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

15. Brett went back to Leary , expecting to be turned down again , but this time , Leary gave Brett the aircraft he wanted . " Perhaps " , [MASK] speculated , " Leary had heard from Washington " .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Brett**
Correct?: **yes**

16. Maurice White spoke to Stepney on the morning of May 17 , 1976 , but later that day , Earth , Wind & Fire keyboardist Larry Dunn received a phone call , informing him that [MASK] had died of a heart attack .
    **not ambiguous**
    Pronoun in place of [MASK]?: **no**
    Annotator's answer: **Stepney**
    Correct?: **yes**

17. Li Yi sought aid from Gao , who personally led two thousand cavalry soldiers to aid Li Yi , causing Dou to withdraw . Gao thereafter sought to submit to Tang , through [MASK] .
    **not ambiguous**
    Pronoun in place of [MASK]?: **no**
    Annotator's answer: **Li Yi**
    Correct?: **yes**

18. At Cambridge , Rose studied under Hubert Middleton and Edward Joseph Dent from 1935 to 1939 . [MASK] started his academic career at The Queen 's College , Oxford .
    **not ambiguous**
    Pronoun in place of [MASK]?: **yes**
    Annotator's answer: **Rose**
    Correct?: **yes**

19. Brady and Bolger leave with Rothbaum , and Rothbaum demands the money Brady owes him . When Rothbaum threatens to kill them if they don 't pay up , Bolger shoots Rothbaum 's thugs , and Brady stabs [MASK] , killing him .
    **not ambiguous**
    Pronoun in place of [MASK]?: **no**
    Annotator's answer: **Rothbaum**
    Correct?: **yes**

20. In the reception room , a boy named Billy won 't stop staring at Don . [MASK] is drawing a picture and then rips it out of his book and hands it to

Billy , getting up and leaving .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Don**
Correct?: **yes**

21. Hasan and Stein agree that Harsa became king in 1089 . Utkarsa was disliked and soon deposed , with a half - brother called Vijayamalla supporting [MASK] and being at the forefront of the rebellion against the king .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

22. Yet this did not prevent Leisegang from reasserting that Aristotle 's own pattern of thinking was incompatible with a proper understanding of Plato . " Therein Cherniss believed Jaeger to be contrary to [MASK] , and Leisegang was unsympathetic to compatibility between Plato and Aristotle in both and above .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Leisegang**
Correct?: **yes**

23. Aska later decides not to rule Cephiro because Fuu told her that the Pillar can think only of Cephiro , but since Lady Aska loves the people of Fahren , she cannot complete the task of Pillar in [MASK] .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Cephiro**
Correct?: **yes**

24. As the weeks wore on , it became evident that Nick could be Lujack 's twin brother , hence , Alex 's son . [MASK] soon became obsessed with Nick and Mindy warned him that Alex wouldn 't give up until she got him .

**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Alex**
Correct?: **yes**

25. Due to Jijii 's psychological manipulation , Ichi believes that Kaneko is his brother and confronts him . Kaneko shoots the side of [MASK] 's leg , causing Ichi to slit Kaneko 's throat in front of Takeshi .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Ichi**
Correct?: **yes**

26. Frank , Jump and other members of the gang go to Clay 's social club , where Frank tells Clay that he wants a percentage of all Clay 's profits . When Clay insults him , [MASK] shoots the Mafioso .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

27. The spoils were to be divided between Shivaji , Kootab Shah and Bijapur . With the agreement concluded and with [MASK] giving him money , horses and artillery , Sivajee set out in March 1677 for his invasions via Kurnool , Cuddapah and Madras .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

28. Billingsley 's response was a gift—bow ties for Ace . [MASK] 's reply was to ask Billingsley for some matching socks so he would be well - dressed when he was refused admittance again .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**

Annotator's answer: **Ace**
Correct?: **yes**

29. In 890 , when Zhu asked Luo Hongxin the military governor of Weibo Circuit for permission to go through Luo 's territory to attack Hedong , Luo refused . `[MASK]` reacted by sending Ding , Ge , Pang Shigu , and Huo Cun to attack Weibo .
**not ambiguous**
Pronoun in place of `[MASK]`?: **yes**
Annotator's answer: **Zhu**
Correct?: **yes**

30. However , it is recorded that Lewis was born in 1381 and sent to the school at Oxford at age 10 ; it is also known that Chaucer 's " Treatise on the Astrolabe " was written for `[MASK]` .
**not ambiguous**
Pronoun in place of `[MASK]`?: **yes**
Annotator's answer: **Lewis**
Correct?: **yes**

31. Claiborne and the other survivors are rescued , thanks to quick action by Taylor and Harold . At the interment , `[MASK]` begs Claiborne to take him and Harold on his expedition to K2 , the second highest peak in the world .
**not ambiguous**
Pronoun in place of `[MASK]`?: **yes**
Annotator's answer: **Taylor**
Correct?: **yes**

32. That year , LONGi signed a contract with Yingli to cooperate on monocrystalline products . In early 2016 , `[MASK]` signed a $1 .
**ambiguous**
Pronoun in place of `[MASK]`?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

33. Katie is taken to Children 's Hospital , and Louise and Wes find themselves being arrested for " what authorities are calling the worst case of child abuse they 've ever seen " . Shortly before his trial begins , [MASK] kills himself .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Wes**
Correct?: **yes**

34. When it becomes clear that Warren has shifted his interest from Marjorie to Bernice , Marjorie sets about humiliating Bernice by tricking her into going through with bobbing her hair . When [MASK] comes out of the barbershop with the new hairdo , her hair is flat and strange .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Bernice**
Correct?: **yes**

35. Rebello defeated David Cho via Unanimous Decision at PXC 26 - Meanest Game Face on August 20 , 2011 . [MASK] made his WEC debut at WEC 39 , losing to Kenji Osawa via split decision .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Rebello**
Correct?: **yes**

36. Karen overhears Bernadette talking to Keanu , who [MASK] thinks may be the baby 's father .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Karen**
Correct?: **yes**

37. Murray Abraham , Daphne Rubin - Vega , Henry Winkler , Kathryn Boule and Judy Kuhn also got their start with Theatreworks . [MASK] has won many awards in its long history .

**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Theatreworks**
Correct?: **yes**

38. In Jacmel , three weeks prior to his reunion with Marie , Paul spends time with his lover and fiancée Natasha . Natasha harbors feelings of mistrust for [MASK] , who left for New York after the earthquake , and spent three years there without having ever contacted her .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Marie**
Correct?: **no**

39. After the match , Deuce 'n Domino attacked Snuka and Slaughter until Tony Garea and Rick Martel came into the ring to assist Snuka and [MASK] .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Slaughter**
Correct?: **yes**

40. Later , Jude and Noah realize that they will be working together , as [MASK] is a new sideline reporter assigned to the Devils .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

41. As such , Michael , Madeline , Sam , Fiona , and Jesse are all hell - bent on exacting revenge for Nate 's murder . Eventually , Michael , with his former mentor Tom Card helping him , tracks [MASK] 's killer , Tyler Grey , to Panama .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**

Annotator's answer: **Nate**
Correct?: **yes**

42. Eileen later follows Des to Erinsborough to check up on him and she takes an instant dislike to [MASK] ' housemate , Daphne Lawrence .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Des**
Correct?: **yes**

43. Although Armstrong was a third party not in privity with Leyland , and a stranger to the car purchase transaction , nonetheless Armstrong was permitted to rely on the non - derogation rights of the car owners relative to [MASK] .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Leyland**
Correct?: **yes**

44. Its theological center and the Fatima Masumeh Shrine are prominent features of Qom . Another very popular religious site of pilgrimage formerly outside the city of [MASK] but now more of a suburb is called Jamkaran .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Qom**
Correct?: **yes**

45. Schult was married to Elke Koska for 25 years , who Schult considers his muse - she was also his manager , now in cooperation with Anna Zlotovskaya , the Russian classical violinist , [MASK] married in 2010 .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Schult**
Correct?: **yes**

46. Gordon wakes up and successfully escapes from Nygma . Shaking off [MASK] 's pursuit , Gordon reaches Bruce and Selina 's hideout and collapses .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Nygma**
Correct?: **yes**

47. During World War II , Hill , as well as Lewis , filed for conscientious objector status . After the war , [MASK] , Hill and a small group of former conscientious objectors created the Pacifica Foundation in Pacifica , California .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Lewis**
Correct?: **yes**

48. Lana tries to intervene but is punched in the stomach by Dino . Luca lunges at Dino but [MASK] pushes him to the ground .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Dino**
Correct?: **yes**

49. On October 28 , 2010 , Facebook banned Rapleaf from scraping data on Facebook , and [MASK] said it would delete the Facebook IDs it had collected .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Rapleaf**
Correct?: **yes**

50. After arriving at the camp , Charlie apologizes to Claire , but Claire tells him to leave her and her son alone . [MASK] then goes into the jungle , and opens a hiding place where he is keeping Virgin Mary statues to put the one Eko gave him .
**not ambiguous**

Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Charlie**
Correct?: **yes**

51. When Sylvie belittles Babe , she leaves Sylvie by the canal in the rain , although Sylvie is found and Shirley realises that Babe left [MASK] to die and disowns her .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Sylvie**
Correct?: **yes**

52. Gregory , as an infant , drowned in a bathtub when Kay became distracted from a call from Sam . [MASK] and Kay ended up divorcing .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Sam**
Correct?: **yes**

53. Stefanie in Rio is a 1960 West German romantic comedy film directed by Curtis Bernhardt and starring Carlos Thompson , Sabine Sinjen and Andréa Parisy . It is a sequel to the 1958 film " [MASK] " .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

54. Kent Ling and his team of assassins are then forced to rescue Ling Hung , but it involved them and Ling Hung having to be in a very deadly gun battle against Kam Tin 's henchmen and unfortunately [MASK] 's team are all killed in the process .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Kent Ling**
Correct?: **yes**

55. On the weekend of January 14 , 2017 , Walker was planning to compete at the 2016 Montana ProRodeo Circuit Finals in Great Falls . [MASK] was in 2nd place in the circuit standings with $14 , 351 so far .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Walker**
Correct?: **yes**

56. Samantha , Jennifer , Billy , Taylor and Coop leave by the end of the season . In the season finale , [MASK] gives birth to Michael 's son and agrees to share motherhood with the returning Jane .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

57. Appears in " " Bucky was a worker who encountered Michael Myers as he wandered around an electrical power plant . [MASK] told Michael that he was not permitted on the grounds .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Bucky**
Correct?: **yes**

58. In August of the same year , Laura came to Coronation Street to tell Alan and Elsie Tanner that she was getting re - married and wanted to drop [MASK] 's loan - although Elsie refused .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Elsie**
Correct?: **no**

59. When Spike lands , Jerry sticks out his tongue and throws Spike 's lips over his own face , provoking [MASK] to chase him around the corner .
**not ambiguous**

Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Spike**
Correct?: **yes**

60. A year later she met Friedrich Schiller and played Luise Miller in his first performance of Kabale und Liebe . Sophie Albrecht and [MASK] had similar interests and became close friends .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Friedrich Schiller**
Correct?: **yes**

61. Walcott lost the count as Ali circled around a floored Liston and [MASK] tried to get him back to a neutral corner .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Walcott**
Correct?: **yes**

62. At the World Matchplay , Whitlock recorded wins over Kevin Painter , Raymond van Barneveld and James Wade to reach the semi - finals of the event for the second time , with [MASK] stating he was playing his best darts in five years .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

63. In the long period that Lars Semb was manager at Moss Jernverk he traveled almost yearly to the mining areas and he subsequently stayed with the local agents . [MASK] was totally dependent on charcoal that the surrounding farmers produced .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**

Annotator's answer: **Moss Jernverk**
Correct?: **yes**

64. Armstrong felt impressed with the style of Hansen 's work . In June 2002 , Armstrong and [MASK] signed a formal agreement .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Hansen**
Correct?: **yes**

65. In 1235 and 1239 the da Camino managed to obtain the rule in Treviso , but the second time they were betrayed by Alberico da Romano , who expelled the Guelphs from the city . However , with Gherardo III da Camino the [MASK] regained prominence .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Guelphs**
Correct?: **yes**

66. The small forward Shamell Stallworth made a three - pointer with the clock reset already and gave the victory to Pinheiros . This game was extremely important because [MASK] because Pinheiros finished the regular season in front of Flamengo precisely by direct confrontation .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Pinheiros**
Correct?: **yes**

67. In the next round , Williams faced Alona Bondarenko and once again won with only dropping 5 games . In the quarterfinals , for the third match in a row , [MASK] only dropped five games this time to Czech Lucie afářová .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Williams**
Correct?: **yes**

68. Most traffic is along the stretch between Falkner and Ripley . The Mississippi Department of Transportation calculated an average of 14 , 000 vehicles passing along the route near [MASK] .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

69. Asmodeus kills Vicki and then attacks Dave and Susan . Dave and [MASK] flee to a cemetery and destroy the demon with a cross .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Susan**
Correct?: **yes**

70. In France , Andrianarivo met with former President Albert Zafy on June 11 , 2007 ; Zafy had also met with Ratsiraka and former Deputy Prime Minister Pierrot Rajaonarivelo in the previous days . [MASK] and Ratsiraka met with Zafy again on June 25 .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Andrianarivo**
Correct?: **yes**

71. His testimony addressed the key " lie " : that Clinton was allegedly pressuring Betty Currie and Blumenthal himself to attest that it was Lewinsky who initially pursued [MASK] , not vice versa .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Clinton**
Correct?: **yes**

72. Leornardo sent Garrett , Vasili Dassiev , Shoji Soma , and Daniel Whitehall to Giza to acquire a power source from a Brood vessel after destroying the Brood inside . [MASK] approved the idea of using the power source to run

the rejuvenations chambers found by another team .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

73. Eventually , Saori and Yoshino rejoin Shuichi 's group of friends , though Saori says she still hates Yoshino and Momoko . Shuichi and Anna start dating , much to the surprise of their friends and [MASK] 's sister .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

74. Gmina Branice contains the villages and settlements of Bliszczyce , Boboluszki , Branice , Dzbańce , [MASK] - Osiedle , Dzierżkowice , Gródczany , Jabłonka , Jakubowice , Jędrychowice , Lewice , Michałkowice , Niekazanice , Posucice , Turków , Uciechowice , Włodzienin , Wódka and Wysoka .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Dzbańce**
Correct?: **yes**

75. When she sees Franky getting into Luke 's car , she gets into a van with Matty . Grace joins him , wanting to talk , but Liv rushes up and demands they follow Franky and [MASK] .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Luke**
Correct?: **yes**

76. She explained that Laura was frantic and told her that Luis jumped in the water channel and that she was unable to see him anymore . Supposedly , the group of friends met [MASK] at the park and started looking for Luis .
**not ambiguous**

Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Laura**
Correct?: **yes**

77. Whether it was Sidney who next challenged Vere to a duel or the other way around , Vere did not take it further , and the Queen personally took Sidney to task for not recognizing the difference between his status and [MASK] 's .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Vere**
Correct?: **yes**

78. Wei attempts to rescue Ku , only to find out that Po deduced Wei 's identity as a cop , since [MASK] was too skilled compared to the rest of his gang .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Po**
Correct?: **no**

79. After an unsuccessful evening on the town , Clark takes Sarah to the Indian side of Calcutta , where they attend a party at the home of a wealthy socialite . There , [MASK] seduces Sarah by challenging her to taste life .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Clark**
Correct?: **yes**

80. The incapacitated Mike is stabbed repeatedly by Erin , who ties a rope around his neck , attaches the other end to a tractor , and drives the vehicle until Mike 's neck snaps . [MASK] stumbles outside , and discovers Danny , who is barely alive .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Erin**
Correct?: **yes**

81. Slingsby was married to the sister of Lawford 's wife , hence why Lawford had to give Slingsby a chance to command the Light Company to prove himself which angered Sharpe . [MASK] was regarded as a poor officer who was often drunk .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

82. Hans and Gerda 's mutual attraction is a challenge , as Gerda is navigating her changing relationship to Lili ; but Hans ' long - time friendship with and affection for [MASK] cause him to be supportive of both Lili and Gerda .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

83. Sirius " sailed in ballast , having unloaded a cargo of hay at Røsneshavn after departing Tromsø . She had left [MASK] in the morning of 17 May 1940 . "
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Røsneshavn**
Correct?: **yes**

84. Harold does not take the news well , but Karl eventually convinces him to fight . [MASK] has an operation and begins chemotherapy after speaking to Stephanie Scully .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Harold**
Correct?: **yes**

85. Ron knocks Dale out and leaves her in a locked car filling with exhaust , sadistically goading Andrew into braving his agoraphobia in order to save

her . Andrew manages to save her and wound Ron ; reviving , Dale deals [MASK] a death blow .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Ron**
Correct?: **yes**

86. Evans introduced two romantic interests for Corrigan : Anina Kreemar , the wealthy niece of Corrigan 's bureau chief , and [MASK] 's friendly rival Jennever Brand , a spirited female agent of a rival clandestine spy agency .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Corrigan**
Correct?: **yes**

87. Her fifth victim was Pillama , aged 60 , and killed at Maddur Vyadyanathapura . [MASK] was a temple priest at Hebbal temple .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Pillama**
Correct?: **yes**

88. Sandy tells Rizzo she plans to watch the race and offers to help [MASK] despite the rumors about Rizzo 's character that have been spread around school .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Rizzo**
Correct?: **yes**

89. Qianru 's mother , Fengyi tries to talk her into accepting the fact that Huanhuan is autistic but Qianru is unwilling to face reality . After some time , under Wenxin 's patient persuasion , [MASK] finally agrees to send Huanhuan to a school for children with special needs .
**not ambiguous**

Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Qianru**
Correct?: **yes**

90. After Trey pushes away Guy , Trey finally realized that Alex was right all along and that Guy has been trying to break them up . Trey and Alex kick [MASK] out of their home and later apologizes to Alex .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Guy**
Correct?: **yes**

91. Burns 's clashes with Smith was perhaps most obvious at the notorious New York City concert in 1998 where Burns attacked [MASK] after the vocalist repeatedly and deliberately knocked one of Burns 's cymbal stands to the floor .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Smith**
Correct?: **yes**

92. When Viki asks if he loves Echo , Charlie hesitates , and Viki storms off . On April 12 , Viki asks [MASK] again whether or not he loves Echo ; he says he does .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Charlie**
Correct?: **yes**

93. In pre - sentence proceedings , Chen 's father , Edward Chen , was reported as saying : During his final plea on 2 February 2006 , Chen said : On 15 February 2006 [MASK] was sentenced to life imprisonment .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**

Annotator's answer: **Chen**
Correct?: **yes**

94. Andrea considers assassinating the Governor , but Milton knows that his second - in - command , Martinez , will follow through on the Governor 's plans . Instead , [MASK] urges Andrea to escape and warn Rick and the others .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Milton**
Correct?: **yes**

95. Patterson took the communiqué to the White House , where Truman and Attlee signed it on 16 November 1945 . The next meeting of the Combined Policy Committee on 15 April 1946 produced no accord on collaboration , and resulted in an exchange of cables between Truman and [MASK] .
**ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **N/A**
Correct?: **N/A**

96. In any case , Baldwin 's other brother Philip of Namur remained as regent , and eventually both of [MASK] 's daughters , Joan and Margaret II , were to rule as countesses of Flanders .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Baldwin**
Correct?: **yes**

97. In the late 1990s , Luis Rossi , Ivan Fernandez , and Mercedes Fernandez purchased the Aragon . In September 2014 , [MASK] sold all her interests in the Aragon .
**not ambiguous**
Pronoun in place of [MASK]?: **no**

Annotator's answer: **Mercedes Fernandez**
Correct?: **yes**

98. Peschko also played chamber music ; best known are his projects with violinist Georg Kulenkampff and cellists Enrico Mainardi and Hans Adomeit . From 1953 to 1958 [MASK] was responsible for lieder , choir and church music at Radio Bremen .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Peschko**
Correct?: **yes**

99. Realising the war was lost , Himmler attempted to open peace talks with the western Allies without Hitler 's knowledge , shortly before the end of the war . Hearing of this , [MASK] dismissed him from all his posts in April 1945 and ordered his arrest .
**not ambiguous**
Pronoun in place of [MASK]?: **yes**
Annotator's answer: **Hitler**
Correct?: **yes**

100. Dein was behind the appointment of the then little known Arsène Wenger to the manager 's job in 1996 ; under Wenger , Arsenal have won the Premier League three times and the FA Cup seven times , and [MASK] strongly backed him and his transfer wishes throughout .
**not ambiguous**
Pronoun in place of [MASK]?: **no**
Annotator's answer: **Dein**
Correct?: **yes**

# Appendix B

# Full Knowledge Base Completion Results

This appendix contains detailed information on the best performance of all models. Table B.1 contains the detailed information on the performance of the best models both on validation and test sets of the datasets. Table B.2 includes information on the best hyperparameter setups and approximate training times. Note that the given times are approximate and are strongly affected by the selection of the hyperparameters as well as external factors.

| Model | Pre-trained? | OlpBench validation | | | | | | | | OlpBench test | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 |
| GRU_TuckER | no | **55.6K** | .058 | .0.033 | .060 | .076 | .104 | .163 | .195 | **57.2K** | .053 | .029 | .054 | .070 | .097 | .155 | .189 |
| GRU_ConvE | no | 57.6$K$ | .047 | .025 | .048 | .063 | .090 | .147 | .179 | **57.2K** | .045 | .022 | .047 | .060 | .086 | .140 | .173 |
| GRU_5⋆E | no | 60.1$K$ | **.060** | **.034** | **.061** | **.078** | **.109** | **.171** | **.205** | 59.9$K$ | **.055** | **.030** | **.056** | **.075** | **.101** | **.160** | **.194** |

| Model | Pre-trained? | ReVerb20k validation | | | | | | | | ReVerb20k test | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 |
| NoEncoder_TuckER | no | 2532 | .196 | .160 | .207 | .230 | .265 | .318 | .352 | 2611 | .196 | .157 | .212 | .236 | .267 | .332 | .362 |
| | yes | 354 | .367 | .283 | .400 | .457 | .529 | .646 | .702 | 303 | .367 | .295 | .412 | .466 | .540 | .659 | .714 |
| NoEncoder_ConvE | no | 1376 | .279 | .233 | .301 | .329 | .364 | .423 | .458 | 1419 | .282 | .227 | .308 | .338 | .380 | .442 | .475 |
| | yes | 292 | .392 | .312 | .423 | .477 | .546 | .658 | .706 | 227 | .400 | .313 | .440 | .491 | .568 | .680 | .729 |
| NoEncoder_5⋆E | no | 2213 | .230 | .180 | .243 | .279 | .325 | .421 | .468 | 2301 | .228 | .174 | .243 | .278 | .334 | .429 | .473 |
| | yes | 773 | .249 | .194 | .263 | .299 | .353 | .466 | .528 | 780 | .249 | .191 | .264 | .302 | .363 | .471 | .532 |
| GRU_TuckER | no | 646 | .344 | .277 | .366 | .409 | .470 | .564 | .609 | 598 | .357 | .283 | .391 | .434 | .493 | .594 | .639 |
| | yes | 283 | .386 | .305 | .416 | .465 | .543 | .660 | .707 | 245 | .397 | .315 | .429 | .485 | .558 | .674 | .720 |
| GRU_ConvE | no | 333 | .377 | .305 | .405 | .447 | .516 | .635 | .683 | 334 | .387 | .305 | .421 | .471 | .540 | .648 | .690 |
| | yes | **229** | .397 | .317 | .428 | .477 | .554 | .671 | .726 | **184** | .409 | .326 | .442 | .499 | .573 | .687 | .737 |
| GRU_5⋆E | no | 430 | .379 | .302 | .411 | .454 | .523 | .639 | .685 | 395 | .390 | .311 | .422 | .475 | .546 | .650 | .697 |
| | yes | 243 | **.404** | **.318** | **.440** | **.492** | **.569** | **.690** | **.747** | 202 | **.417** | **.330** | **.455** | **.512** | **.586** | **.701** | **.748** |

| Model | Pre-trained? | ReVerb45k validation | | | | | | | | ReVerb45k test | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 |
| NoEncoder_TuckER | no | 5097 | .094 | .077 | .097 | .109 | .126 | .160 | .177 | 5135 | .103 | .088 | .106 | .116 | .131 | .164 | .182 |
| | yes | 839 | .305 | .228 | .335 | .386 | .453 | .559 | .610 | 780 | .299 | .220 | .332 | .386 | .453 | .557 | .607 |
| NoEncoder_ConvE | no | 2741 | .235 | .180 | .262 | .293 | .336 | .406 | .436 | 2690 | .232 | .179 | .256 | .288 | .333 | .400 | .434 |
| | yes | 632 | .353 | .274 | .384 | .438 | .506 | .611 | .662 | 666 | .345 | .267 | .378 | .432 | .500 | .601 | .644 |
| NoEncoder_5⋆E | no | 3653 | .148 | .116 | .151 | .174 | .210 | .289 | .324 | 3460 | .152 | .123 | .152 | .174 | .212 | .289 | .327 |
| | yes | 3233 | .187 | .151 | .195 | .222 | .258 | .320 | .349 | 3279 | .189 | .153 | .198 | .224 | .261 | .325 | .357 |
| GRU_TuckER | no | 1386 | .304 | .242 | .330 | .369 | .423 | .504 | .545 | 1398 | .302 | .242 | .323 | .365 | .420 | .506 | .546 |
| | yes | 761 | .336 | .263 | .368 | .413 | .473 | .570 | .613 | 706 | .331 | .258 | .359 | .412 | .477 | .575 | .618 |
| GRU_ConvE | no | 776 | .351 | .273 | .387 | .437 | .501 | .593 | .631 | 824 | .343 | .268 | .374 | .424 | .488 | .584 | .626 |
| | yes | **584** | .364 | .285 | .400 | .452 | .518 | .620 | .665 | 600 | .357 | .277 | .393 | .444 | .509 | .607 | .651 |
| GRU_5⋆E | no | 843 | .362 | .287 | .394 | .446 | .511 | .607 | .649 | 836 | .357 | .278 | .394 | .443 | .508 | .605 | .647 |
| | yes | 603 | **.385** | **.305** | **.421** | **.474** | **.542** | **.642** | **.683** | **596** | **.382** | **.302** | **.416** | **.467** | **.537** | **.636** | **.676** |

| Model | Pre-trained? | Fb15k237 validation | | | | | | | | Fb15k237 test | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 |
| NoEncoder_TuckER | no | 160 | .366 | .276 | .399 | .464 | .547 | .675 | .727 | 166 | .358 | .265 | .393 | .458 | .545 | .669 | .725 |
| | yes | 142 | **.369** | **.277** | **.403** | **.468** | **.555** | **.680** | **.732** | 151 | **.363** | **.269** | **.398** | **.464** | **.550** | **.678** | **.733** |
| NoEncoder_ConvE | no | 200 | .326 | .238 | .356 | .419 | .508 | .634 | .690 | 212 | .320 | .230 | .351 | .417 | .504 | .634 | .690 |
| | yes | 189 | .332 | .242 | .363 | .428 | .513 | .647 | .703 | 200 | .325 | .233 | .355 | .421 | .510 | .645 | .703 |
| NoEncoder_5⋆E | no | 144 | .358 | .267 | .393 | .456 | .542 | .670 | .723 | 152 | .353 | .260 | .389 | .454 | .539 | .666 | .721 |
| | yes | **137** | .362 | .270 | .398 | .462 | .548 | .672 | .727 | **143** | .357 | .264 | .393 | .459 | .544 | .670 | .727 |
| GRU_TuckER | no | **137** | .355 | .262 | .391 | .454 | .539 | .671 | .724 | 144 | .350 | .256 | .386 | .451 | .538 | .666 | .723 |
| | yes | **137** | .357 | .264 | .393 | .454 | .542 | .670 | .726 | **143** | .354 | .260 | .391 | .453 | .538 | .668 | .724 |
| GRU_ConvE | no | 161 | .340 | .249 | .372 | .433 | .521 | .653 | .708 | 166 | .334 | .242 | .368 | .431 | .519 | .650 | .705 |
| | yes | 151 | .331 | .241 | .361 | .423 | .515 | .652 | .708 | 157 | .327 | .237 | .359 | .422 | .513 | .646 | .704 |
| GRU_5⋆E | no | 145 | .351 | .256 | .387 | .453 | .540 | .669 | .725 | 150 | .345 | .249 | .380 | .449 | .536 | .667 | .725 |
| | yes | 143 | .351 | .257 | .386 | .451 | .537 | .667 | .722 | 145 | .348 | .254 | .384 | .450 | .536 | .666 | .720 |

| Model | Pre-trained? | Wn18rr validation | | | | | | | | Wn18rr test | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 | MR | MRR | H@1 | H@3 | H@5 | H@10 | H@30 | H@50 |
| NoEncoder_TuckER | no | 3701 | .469 | .437 | .483 | .499 | .524 | .571 | .595 | 4097 | .468 | .435 | .483 | .502 | .528 | .576 | .595 |
| | yes | 3332 | .470 | .439 | .479 | .500 | .529 | .583 | .604 | 3456 | .467 | .434 | .480 | .500 | .529 | .583 | .604 |
| NoEncoder_ConvE | no | 6337 | .429 | .403 | .437 | .455 | .479 | .512 | .530 | 6455 | .429 | .404 | .440 | .455 | .479 | .512 | .530 |
| | yes | 5678 | .433 | .407 | .442 | .460 | .483 | .523 | .542 | 5793 | .435 | .408 | .444 | .461 | .486 | .527 | .545 |
| NoEncoder_5⋆E | no | 2527 | **.492** | **.450** | **.505** | **.531** | .574 | .649 | **.680** | **2450** | **.492** | .448 | **.507** | **.534** | **.583** | **.652** | .680 |
| | yes | 2576 | **.492** | **.450** | .501 | .528 | **.575** | **.651** | **.680** | 2636 | **.492** | **.450** | .504 | .533 | .582 | **.652** | **.683** |
| GRU_TuckER | no | 3201 | .456 | .428 | .464 | .484 | .506 | .553 | .574 | 3262 | .456 | .429 | .465 | .483 | .505 | .551 | .573 |
| | yes | 2637 | .457 | .422 | .469 | .490 | .523 | .579 | .609 | 2790 | .455 | .418 | .469 | .494 | .524 | .577 | .600 |
| GRU_ConvE | no | 4376 | .431 | .407 | .439 | .456 | .477 | .518 | .538 | 4474 | .428 | .402 | .438 | .455 | .474 | .512 | .533 |
| | yes | 5983 | .400 | .376 | .410 | .422 | .444 | .475 | .493 | 6128 | .399 | .375 | .409 | .423 | .441 | .473 | .491 |
| GRU_5⋆E | no | **2444** | .456 | .418 | .463 | .492 | .533 | .598 | .629 | 2545 | .452 | .413 | .462 | .489 | .527 | .595 | .629 |
| | yes | 2971 | .426 | .388 | .440 | .464 | .494 | .548 | .575 | 3068 | .420 | .379 | .437 | .459 | .488 | .547 | .573 |

Table B.1: Full results on both validation and test set of all datasets. In addition to the metrics reported in the paper, I also report H@N for $N \in \{1, 3, 5, 10, 30, 50\}$, which appeared in related work. The best value in each column is written in **bold**.

| Model | Pre-trained? | dimension | learning rate | batch | dropout | N3 weight | time |
|---|---|---|---|---|---|---|---|
| | | OlpBench | | | | | |
| GRU_TuckER | no | 300 | $1 \cdot 10^{-4}$ | 4096 | 0.3 | – | 5 days |
| GRU_ConvE | no | 500 | $1 \cdot 10^{-4}$ | 4096 | 0.2 | – | 5 days |
| GRU_5⋆E | no | 500 | $1 \cdot 10^{-4}$ | 4096 | – | 0.03 | 12 days |
| | | ReVerb20k | | | | | |
| Model | Pre-trained? | dimension | learning rate | batch | dropout | N3 weight | time |
| NoEncoder_TuckER | no | 300 | $3 \cdot 10^{-5}$ | 512 | 0.4 | – | 30 min |
| | yes | 300 | $3 \cdot 10^{-4}$ | 512 | 0.3 | – | 30 min |
| NoEncoder_ConvE | no | 300 | $3 \cdot 10^{-4}$ | 1024 | 0.3 | – | 30 min |
| | yes | 500 | $1 \cdot 10^{-4}$ | 512 | 0.2 | – | 30 min |
| NoEncoder_5⋆E | no | 200 | $1 \cdot 10^{-3}$ | 512 | – | 0.3 | 30 min |
| | yes | 500 | $3 \cdot 10^{-4}$ | 512 | – | 0.03 | 30 min |
| Gru_TuckER | no | 300 | $1 \cdot 10^{-3}$ | 1024 | 0.4 | – | 30 min |
| | yes | 300 | $3 \cdot 10^{-5}$ | 2048 | 0.4 | – | 30 min |
| Gru_ConvE | no | 300 | $3 \cdot 10^{-5}$ | 512 | 0.3 | – | 30 min |
| | yes | 500 | $3 \cdot 10^{-5}$ | 512 | 0.3 | – | 30 min |
| Gru_5⋆E | no | 200 | $3 \cdot 10^{-4}$ | 2048 | – | 0.1 | 30 min |
| | yes | 500 | $1 \cdot 10^{-4}$ | 1024 | – | 0.1 | 30 min |
| | | ReVerb45k | | | | | |
| Model | Pre-trained? | dimension | learning rate | batch | dropout | N3 weight | time |
| NoEncoder_TuckER | no | 100 | $1 \cdot 10^{-3}$ | 512 | 0.4 | – | 2.5h |
| | yes | 300 | $3 \cdot 10^{-4}$ | 4096 | 0.3 | – | 2.5h |
| NoEncoder_ConvE | no | 300 | $3 \cdot 10^{-4}$ | 512 | 0.3 | – | 2h |
| | yes | 500 | $1 \cdot 10^{-4}$ | 2048 | 0.3 | – | 2.5h |
| NoEncoder_5⋆E | no | 200 | $1 \cdot 10^{-3}$ | 2048 | – | 0.3 | 3h |
| | yes | 200 | $1 \cdot 10^{-4}$ | 512 | – | 0.1 | 3h |
| Gru_TuckER | no | 300 | $1 \cdot 10^{-3}$ | 512 | 0.4 | – | 3h |
| | yes | 300 | $3 \cdot 10^{-4}$ | 2048 | 0.4 | – | 3h |
| Gru_ConvE | no | 300 | $1 \cdot 10^{-4}$ | 4096 | 0.3 | – | 2h |
| | yes | 500 | $3 \cdot 10^{-4}$ | 2048 | 0.3 | – | 2.5h |
| Gru_5⋆E | no | 500 | $3 \cdot 10^{-4}$ | 1024 | – | 0.03 | 3h |
| | yes | 500 | $3 \cdot 10^{-4}$ | 2048 | – | 0.1 | 3h |
| | | Fb15k237 | | | | | |
| Model | Pre-trained? | dimension | learning rate | batch | dropout | N3 weight | time |
| NoEncoder_TuckER | no | 200 | $3 \cdot 10^{-4}$ | 1024 | 0.4 | – | 9.5h |
| | yes | 300 | $3 \cdot 10^{-5}$ | 2048 | 0.4 | – | 9.5h |
| NoEncoder_ConvE | no | 300 | $3 \cdot 10^{-4}$ | 2048 | 0.3 | – | 8.5h |
| | yes | 500 | $3 \cdot 10^{-4}$ | 512 | 0.3 | – | 9h |
| NoEncoder_5⋆E | no | 500 | $1 \cdot 10^{-4}$ | 512 | – | 0.3 | 12h |
| | yes | 500 | $1 \cdot 10^{-4}$ | 512 | – | 0.3 | 12h |
| Gru_TuckER | no | 100 | $3 \cdot 10^{-4}$ | 512 | 0.4 | – | 13h |
| | yes | 200 | $1 \cdot 10^{-4}$ | 1024 | 0.4 | – | 13h |
| Gru_ConvE | no | 300 | $3 \cdot 10^{-4}$ | 512 | 0.3 | – | 11h |
| | yes | 500 | $3 \cdot 10^{-4}$ | 512 | 0.3 | – | 12h |
| Gru_5⋆E | no | 500 | $1 \cdot 10^{-4}$ | 1024 | – | 0.1 | 24h |
| | yes | 500 | $3 \cdot 10^{-4}$ | 512 | – | 0.1 | 24h |
| | | Wn18rr | | | | | |
| Model | Pre-trained? | dimension | learning rate | batch | dropout | N3 weight | time |
| NoEncoder_TuckER | no | 100 | $1 \cdot 10^{-3}$ | 512 | 0.3 | – | 7.5h |
| | yes | 100 | $1 \cdot 10^{-3}$ | 512 | 0.3 | – | 7.5h |
| NoEncoder_ConvE | no | 300 | $1 \cdot 10^{-3}$ | 512 | 0.3 | – | 8h |
| | yes | 300 | $1 \cdot 10^{-3}$ | 512 | 0.3 | – | 8h |
| NoEncoder_5⋆E | no | 200 | $1 \cdot 10^{-3}$ | 512 | – | 0.3 | 8h |
| | yes | 500 | $1 \cdot 10^{-3}$ | 1024 | – | 0.3 | 8h |
| Gru_TuckER | no | 100 | $1 \cdot 10^{-3}$ | 512 | 0.3 | – | 6h |
| | yes | 100 | $1 \cdot 10^{-3}$ | 1024 | 0.4 | – | 6h |
| Gru_ConvE | no | 300 | $1 \cdot 10^{-3}$ | 1024 | 0.3 | – | 8.5h |
| | yes | 500 | $1 \cdot 10^{-3}$ | 2048 | 0.3 | – | 8.5h |
| Gru_5⋆E | no | 500 | $3 \cdot 10^{-4}$ | 512 | – | 0.1 | 8.5h |
| | yes | 500 | $1 \cdot 10^{-3}$ | 1024 | – | 0.3 | 8.5h |

Table B.2: Best hyperparameter setups and training time of best models for all datasets.

# Bibliography


Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. In *Advances in Neural Information Processing Systems*, volume 33, pages 9649–9661. Curran Associates, Inc.

Lauren Ackerman. 2019. Syntactic and cognitive issues in investigating gendered coreference. *Glossa: A Journal of General Linguistics*, 4(1):117.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. 2020. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *CoRR*, abs/2006.13365.

Andrew Altman. 2020. Discrimination. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Winter 2020 edition. Metaphysics Research Lab, Stanford University.

Pascal Amsili and Olga Seminck. 2017. A Google-proof collection of French Winograd schemas. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 24–29, Valencia, Spain. Association for Computational Linguistics.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853.

Alya Asarina. 2009. Gender and adjective agreement in russian ∗. The 4th Annual Meeting of the Slavic Linguistics Society, Zadar, Croatia.

Tobias Baer. 2019. *How Real-World Biases Are Mirrored by Algorithms*, pages 53–57. Apress, Berkeley, CA.

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. 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 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Timothée Bernard and Ting Han. 2020. Mandarinograd: A Chinese collection of Winograd schemas. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. 2020. Can we predict new facts with open knowledge graph embeddings? a benchmark for open link prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2308, Online. Association for Computational Linguistics.

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. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. Spectral networks and locally connected networks on graphs. In *Proceedings of the 3rd ICLR*.

Chandrahas and Partha Pratim Talukdar. 2021. OKGIT: open knowledge graph link prediction with implicit types. In *Proceedings of the 59th ACL*.

Eugene Charniak. 1972. *Toward a model of children's story comprehension*. Ph.D. thesis, Massachusetts Institute of Technology.

Nancy A Chinchor. 1998. Overview of muc-7/met-2. Technical report, Science Applications International Corp.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. 2018. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 134–148, New York, NY, USA. PMLR.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. 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 1405–1415, Beijing, China. Association for Computational Linguistics.

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

E. Davis and X. Pan. 2015. A corpus of challenging pronoun disambiguation problems, adapted from children's books.

Ernest Davis, Leora Morgenstern, and Charles L Ortiz. 2017. The first winograd schema challenge at ijcai-16. *AI Magazine*, 38(3):97–98.

Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Proceedings of the 32th AAAI*, pages 1811–1818, New Orleans, USA.

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

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, Seattle, Washington, USA. Association for Computational Linguistics.

Ali Emami, Noelia De La Cruz, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. 2018. A knowledge hunting framework for common sense reasoning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1958, Brussels, Belgium. Association for Computational Linguistics.

Ali Emami, Paul Trichelair, Adam Trischler, Kaheer Suleman, Hannes Schulz, and Jackie Chi Kit Cheung. 2019. The KnowRef coreference corpus: Removing gender and number cues for difficult pronominal anaphora resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3952–3961, Florence, Italy. Association for Computational Linguistics.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

J. Fähndrich, S. Weber, and H. Kanthak. 2018. A marker passing approach to Winograd schemas. In *Semantic Technology*. Springer.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, page 1679–1688, New York, NY, USA. Association for Computing Machinery.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. 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 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA, USA. `http://www.deeplearningbook.org`.

Ralph Grishman and Beth Sundheim. 1996. Design of the muc-6 evaluation. Technical report, New York University, Dept. of Computer Science.

Barbara J Grosz. 1977. The representation and use of focus in a system for understanding dialogs. In *IJCAI*, pages 67–76.

Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. 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 84–94, Beijing, China. Association for Computational Linguistics.

Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. 2019. CaRe: Open knowledge graph embeddings. 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 378–388, Hong Kong, China. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. 2019. A hybrid neural network model for commonsense reasoning. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 13–21, Hong Kong, China. Association for Computational Linguistics.

Jerry R Hobbs. 1979. Coherence and coreference. *Cognitive science*, 3(1):67–90.

Jerry R Hobbs, Mark E Stickel, Douglas E Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial intelligence*, 63(1-2):69–142.

Patrick Hohenecker, Frank Mtumbuka, Vid Kocijan, and Thomas Lukasiewicz. 2020. Systematic comparison of neural architectures and training approaches for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8554–8565, Online. Association for Computational Linguistics.

Suk Joon Hong and Brandon Bennett. 2020. Tackling domain-specific winograd schemas with knowledge-based reasoning and machine learning. *arXiv preprint arXiv:2011.12081*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Wenhao Huang, Ge Li, and Zhi Jin. 2016. Improved knowledge base completion by path-augmented transr model. *CoRR*, abs/1610.04073.

N. Isaak and L. Michael. 2016. Tackling the winograd schema challenge through machine logical inferences. In *STAIRS*.

N. Isaak and L. Michael. 2019. Winoflexi: A crowdsourcing platform for the development of winograd schemas. In *Proc. AI*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. 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 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

Megumi Kameyama. 1986. A property-sharing constraint in centering. In *24th Annual Meeting of the Association for Computational Linguistics*, pages 200–206.

Andrew Kehler, Laura Kertz, Hannah Rohde, and Jeffrey L Elman. 2008. Coherence and coreference revisited. *Journal of semantics*, 25(1):1–44.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

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

Tassilo Klein and Moin Nabi. 2019. Attention is (not) all you need for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4831–4836, Florence, Italy. Association for Computational Linguistics.

Jon Kleinberg. 2018. Inherent trade-offs in algorithmic fairness. *SIGMETRICS Perform. Eval. Rev.*, 46(1):40.

Vid Kocijan, Oana-Maria Camburu, Ana-Maria Cretu, Yordan Yordanov, Phil Blunsom, and Thomas Lukasiewicz. 2019a. WikiCREM: A large unsupervised corpus for coreference resolution. 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 4303–4312, Hong Kong, China. Association for Computational Linguistics.

Vid Kocijan, Oana-Maria Camburu, and Thomas Lukasiewicz. 2021. The gap on gap: Tackling the problem of differing data distributions in bias-measuring datasets. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, February 2–9, 2021*. AAAI Press.

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019b. A surprisingly robust trick for the Winograd schema challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4837–4842, Florence, Italy. Association for Computational Linguistics.

Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th ICML*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

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

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of the 2nd SysML Conference*, Palo Alto, CA, USA.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2011. The winograd schema challenge. *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 46.

Hector J. Levesque. 2011. The Winograd Schema Challenge. *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Tttttackling winogrande schemas. *arXiv preprint arXiv:2003.08380*.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 705–714, Lisbon, Portugal. Association for Computational Linguistics.

Fei Liu, Luke Zettlemoyer, and Jacob Eisenstein. 2019a. The referential reader: A recurrent entity network for anaphora resolution. *CoRR*, abs/1902.01541.

Haokun Liu, William Huang, Dhara Mungra, and Samuel R. Bowman. 2020. Precise task formalization matters in Winograd schema evaluations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8275–8280, Online. Association for Computational Linguistics.

Q. Liu, H. Jiang, Z.-H. Ling, X. Zhu, S. Wei, and Y. Hu. 2017a. Combing context and commonsense knowledge through neural networks for solving Winograd Schema Problems. *AAAI Spring Symposium Series*.

Quan Liu, Hui Jiang, Andrew Evdokimov, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. 2017b. Cause-effect knowledge acquisition and neural association model for solving a set of winograd schema problems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2344–2350.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

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

N. Lourie, R. Le Bras, C. Bhagavatula, and Y. Choi. 2021. UNICORN on RAINBOW: A universal commonsense reasoning model on a new multitask benchmark. *Proceedings of the AAAI Conference on Artificial Intelligence*.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*.

Drew McDermott. 1976. Artificial intelligence meets natural stupidity. *Acm Sigart Bulletin*, pages 4–9.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. *CoRR*, abs/1908.09635.

G. Melo, V. Imaizumi, and F. Cozman. 2020. Esquemas de Winograd em português. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*.

VMojtaba Nayyeri, Sahar Vahdati, Can Aykul, and Jens Lehmann. 2021. 5* knowledge graph embeddings with projective transformations. In *Proceedings of the 35th AAAI*.

J. Opitz and A. Frank. 2018. Addressing the Winograd Schema Challenge as a sequence ranking task. In *Proc. 1st International Workshop on Language Cognition and Computational Models*. ACL.

Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving hard coreference problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819, Denver, Colorado. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088v2*.

Massimo Poesio, Simone Ponzetto, and Yannick Versley. 2011. Computational models of anaphora resolution: A survey.

Massimo Poesio, Roland Stuckardt, and Yannick Versley. 2016. *Anaphora resolution*. Springer.

Ashok Prakash, Arpit Sharma, Arindam Mitra, and Chitta Baral. 2019. Combining knowledge hunting and neural language models to solve the Winograd schema challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6110–6119, Florence, Italy. Association for Computational Linguistics.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. 2019. Language models are unsupervised multitask learners. *pre-print*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *pre-print*.

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

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, Jeju Island, Korea. Association for Computational Linguistics.

Y.-P. Ruan, X. Zhu, Z.-H. Ling, Z. Shi, Q. Liu, and S. Wei. 2019. Exploring unsupervised pretraining and sentence structure modelling for Winograd Schema Challenge. *arXiv:1904.09705*.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

Roger C Schank and Robert P Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum Associates.

A. Sharma. 2019. Using answer set programming for commonsense reasoning in the winograd schema challenge. *arXiv:1907.11112*.

Arpit Sharma, Nguyen H. Vo, Somak Aditya, and Chitta Baral. 2015. Towards addressing the winograd schema challenge: Building and using a semantic parser and a knowledge hunting module. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1319–1325. AAAI Press.

Ming Shen, Pratyay Banerjee, and Chitta Baral. 2021. Unsupervised pronoun resolution via masked noun-phrase prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Candace Sidner. 1979. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*. Ph.D. thesis, Massachusetts Institute of Technology.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. 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 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Sanjay Subramanian and Dan Roth. 2019. Improving generalization in coreference resolution via adversarial training. In *Proceedings of the Joint Conference on Lexical and Computational Sematics*.

W. L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism & Mass Communication Quarterly*, 30:415 – 433.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal. Association for Computational Linguistics.

P. Trichelair, A. Emami, J. C. K. Cheung, A. Trischler, K. Suleman, and F. Diaz. 2018. On the evaluation of common-sense reasoning in natural language understanding. In *Proceedings of NeurIPS Workshop on Critiquing and Correcting Trends in Machine Learning*.

T. H. Trinh and Q. V. Le. 2018. A simple method for commonsense reasoning. *arXiv:1806.02847*.

Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. 2017. Knowledge graph completion via complex tensor factorization. *J. Mach. Learn. Res.*, 18(1):4735–4772.

Yuval Varkel and Amir Globerson. 2020. Pre-training mention representations in coreference models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8534–8540, Online. Association for Computational Linguistics.

Shikhar Vashishth, Prince Jain, and Partha Talukdar. 2018. Cesi: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1317–1327, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *Proceedings of the 6th ICLR.*

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Shuohang Wang, Sheng Zhang, Yelong Shen, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Jing Jiang. 2019b. Unsupervised deep structured semantic models for commonsense reasoning. 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 882–891, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI*.

Zhigang Wang and Juanzi Li. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 1293–1299. AAAI Press.

Kellie Webster, Marta R. Costa-jussà, Christian Hardmeier, and Will Radford. 2019. Gendered ambiguous pronoun (GAP) shared task at the gender bias in NLP workshop 2019. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 1–7, Florence, Italy. Association for Computational Linguistics.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Yorick Wilks. 1975. An intelligent analyzer and understander of english. *Communications of the ACM*, 18(5):264–274.

T. Winograd. 1972. *Understanding Natural Language*. Academic Press, Inc., Orlando, FL, USA.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016a. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2659–2665. AAAI Press.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016b. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2965–2971. AAAI Press.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao,

Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Z.-X. Ye, Q. Chen, W. Wang, and Z.-H. Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *arXiv:1908.06725*.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Yordan Yordanov, Oana-Maria Camburu, Vid Kocijan, and Thomas Lukasiewicz. 2020. Does the Objective Matter? Comparing Training Objectives for Pronoun Resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4963–4969, Online. Association for Computational Linguistics.

H. Zhang and Y. Song. 2018. A distributed solution for winograd schema challenge. In *ACM International Conference Proceeding Series*. ACM.

Zhanqiu Zhang, Jianyu Cai, and Jie Wang. 2020. Duality-induced regularizer for tensor factorization based knowledge graph completion. In *Advances in Neural Information Processing Systems*, volume 33, pages 21604–21615. Curran Associates, Inc.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.