Karlsruhe Institute of Technology

Cross-lingual Coreference Resolution and Neural Machine Translation

Master's Thesis of

Jannik Fabian Hetzer

at the Interactive System Lab
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT)

Reviewer:	Prof. Dr. Alexander Waibel
Second reviewer:	Prof. Dr. Jan Niehues
Advisor:	M.Sc. Ngoc Quan Pham

01. February 2022 – 29. July 2022

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 29 July 2022

. . . .

(Jannik Fabian Hetzer)

Abstract

Like many other Natural Language Processing (NLP) tasks, coreference resolution has benefited massively from new developments in recent years. Deeper and more complex architectures enabled coreference resolution systems that work end-to-end, needing no additional input generated by separate models. Contextual word representations from pre-trained language models such as BERT further enhanced their performance. On the other hand, many of these new techniques can also be applied directly to downstream tasks like Neural Machine Translation (NMT), questioning the use of additional coreference information for these tasks.

In this thesis, I evaluated different variants of BERT as a possible foundation for coreference resolution systems. Some of them promised to increase the performance, while others could reduce the models' complexity and cut down the time needed to train the systems.

Since most of the recent developments in the field of coreference resolution took place in the English language, I applied one of the best-performing models to the German language. Due to the cross-lingual capabilities of multilingual BERT variants such as M-BERT or XLM-RoBERTa, I was able to directly apply models trained exclusively on English data to German texts. To show the benefits transfer learning can bring to coreference resolution in languages with limited annotated data, I analyzed the cross-lingual models on datasets of different sizes.

I augmented context-agnostic and context-aware NMT systems with coreference information and compared their handling of pronouns to prove that coreference information, and thus coreference resolution, can still play a valuable role for specific downstream tasks.

As part of this thesis, I reimplemented the most common state-of-the-art models in PyTorch and made them publicly available to give more people the opportunity to participate in the field of coreference resolution.

Zusammenfassung

Wie viele andere Natural Language Processing (NLP) Tasks hat auch die Coreference Resolution in den letzten Jahren massiv von neuen Entwicklungen profitiert. Tiefere und komplexere Architekturen ermöglichten Coreference Resolution Systeme, die Ende-zu-Ende arbeiten und keinen zusätzlichen Daten von separaten Modellen benötigen. Contextual Word Representations von vortrainierten Sprachmodellen wie BERT steigerten ihre Performance noch weiter. Andererseits können viele dieser neuen Techniken auch direkt auf Aufgaben wie die Neural Machine Translation (NMT) angewendet werden, wodurch die Verwendung von zusätzlichen Koreferenz-Informationen für diese Aufgaben infrage gestellt wird.

In dieser Arbeit habe ich verschiedene vortrainierte Sprachmodelle als mögliches Fundament von Coreference Resolution Systemen untersucht. Einige von ihnen versprachen, die Performance zu steigern, andere waren in der Lage, die Komplexität der Modelle zu reduzieren und die für das Training der Systeme benötigte Zeit zu verkürzen.

Da die meisten der Entwicklungen im Bereich der Coreference Resolution vorwiegend für die englische Sprache stattfanden, habe ich eines der besten Systeme in die deutsche Sprache übertragen. Aufgrund der sprachübergreifenden Fähigkeiten von mehrsprachigen Sprachmodellen wie M-BERT oder XLM-RoBERTa konnte ich Modelle, die ausschließlich auf englischen Daten trainiert wurden, direkt auf deutsche Texte anwenden. Um zu zeigen, welche Vorteile Transfer-Learning für die Coreference Resolution in Sprachen haben kann, die nur über wenig annotierte Daten verfügen, habe ich die sprachübergreifenden Modelle auf deutschen Korpora unterschiedlicher Größe analysiert.

Ich habe NMT-Systeme mit und ohne Kontext, mit zusätzlichen Koreferenz-Informationen trainiert und ihren Umgang mit Pronomen verglichen, um zu beweisen, dass Koreferenz Informationen und damit Coreference Resolution noch immer für ausgewählte Aufgaben eine wertvolle Rolle spielen können.

Im Rahmen dieser Arbeit habe ich die gängigsten State-of-the-Art-Modelle mit PyTorch neu implementiert und öffentlich zugänglich gemacht, um mehr Menschen die Möglichkeit zu geben, sich an der Forschung zu Coreference Resolution zu beteiligen.

Contents

List of Figures

List of Tables

1. Introduction

Coreference resolution is the task of clustering mentions in texts based on the entities they refer to. Identifying mentions referring to the same real-world entity can benefit various downstream tasks like Neural Machine Translation (NMT) [60]. Coreference resolution also represents an essential part of higher-level natural language processing (NLP) tasks involving natural language understanding (NLU). It is also part of the Winograd Schema Challenge, which was proposed by Terry Winograd as an improvement of the Turing test. In order to pass the challenge, the correct linking between noun phrases and ambiguous pronouns is crucial [40, Chapter 21].

In recent years neural systems emerged, outperforming existing models based on a broad range of linguistic features. End-to-end neural coreference resolution systems not only link mentions but also detect mention boundaries themselves, avoiding cascading errors from a distinct mention detection system [49]. Besides architectural advances, coreference resolution systems also profit from the steady improvement of pre-trained word embeddings in recent years [38][39].

1.1. Motivation

While the benefit of using explicit coreference information to improve context-agnostic NMT systems was widely approved in the past, the rise of context-aware systems cast doubt on this benefit [89].

Although context-aware systems indisputably learn some coreference resolution on their own, they do not stack up with the best, specialized systems, as shown in this thesis. Therefore coreference resolution is still a critical NLP task, and providing downstream tasks with coreference information can lead to better results than letting those tasks resolve the coreferences themselves.

Good coreference information from today's neural end-to-end systems enables us to build better-performing systems to solve downstream tasks and reduce the complexity resulting in less resource-hungry and faster to train systems.

1.2. Objectives

This thesis consists of three main contributions to the coreference resolution problem and the application of coreference information in the field of NMT.

The first contribution is the implementation of three end-to-end coreference resolution models with PyTorch. These three models build upon each other, with the third model being one of the best-performing systems today. The original TensorFlow implementations and pre-trained models are publicly available, but it requires good knowledge of the TensorFlow framework to make changes to them. With the provided PyTorch implementations from this work, these models get approachable to a broader range of researchers and facilitate the induction into the coreference resolution problem.

Secondly, I investigate the cross-lingual capabilities of neural end-to-end coreference systems. This involves the application of coreference systems on German datasets, although they were originally designed and tested on English coreference tasks. Besides training the models entirely on German data, I also analyze the benefits of transfer learning for the domain of coreference resolution. Since creating datasets with manually annotated coreference information is time-consuming and costly, those kinds of datasets, big enough to achieve competitive results, are available in a few languages only. Pre-training or even training entirely on English data before applying the system to another language might help to improve the performance in low-resource languages. In this work, I study and compare multiple ways to transfer the knowledge gained by pre-training on large English datasets to the problem of German coreference resolution.

In the final part of this work, I discuss the application of coreference information in NMT systems. To show the impact of such additional information, I augment context-agnostic and context-aware NMT systems with coreference information obtained by a state-of-the-art coreference resolution system. Furthermore, I demonstrate that context-agnostic systems can surpass even more complex context-aware systems on coreference-based tasks when provided with explicit coreference information.

2. Fundamentals

In this chapter, basic deep learning concepts and specific neural networks important for this thesis are briefly explained. In addition, coreference resolution terms are defined, different approaches of coreference resolution systems are outlined, and metrics for evaluating these systems are described.

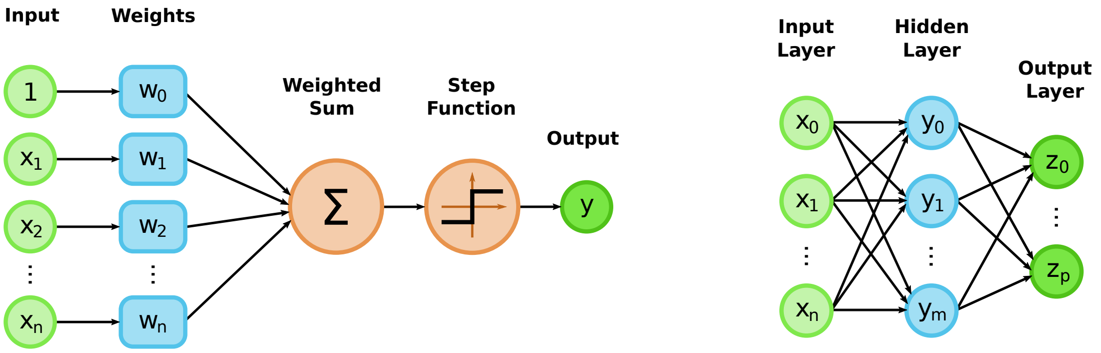
2.1. Neural Networks

The Perceptron by Rosenblatt [75] was the first linear model of a neuron that was able to learn the weights to categorize input examples [27]. Figure 2.1a shows its simple structure. Given the activation function f, which is a step function in this case, the output y for the inputs x_i can be calculated as follows:

$$y = f(\sum_{i=1}^{n} x_i w_i); \quad x_0 = 1$$

To use w_0 as a bias, the input x_0 is fixed to 1 and appended to the actual input data. The perceptron is only applicable to binary, linear classification problems. Early on, it was known that multiple layers of perceptrons with non-linear activation functions like the Multilayer Perceptron (MLP) in Figure 2.1 were necessary to solve more complex problems [78]. However, only with the introduction of the backpropagation algorithm [79] it became possible to train these models using gradient descent efficiently. They defined the sum of the squared distances between the predictions and desired outputs as an error measure. To update each weight in the network with the errors' gradient regarding that specific weight, its gradient is iteratively propagated back from the output to the input layer by applying the chain rule.

Cybenko [18] could finally prove in 1989 that every continuous function on a closed and bounded set could be modeled by a network with a single hidden layer and continuous



(a) Rosenblatt Perceptron

(b) Multilayer Perceptron

Figure 2.1.: Note that the view on the MLP structure is from a higher-level perspective than for the Perceptron. The nodes of the Perceptron refer to data, weights, and functions, whereas each node in the MLP denotes a complete neuron.

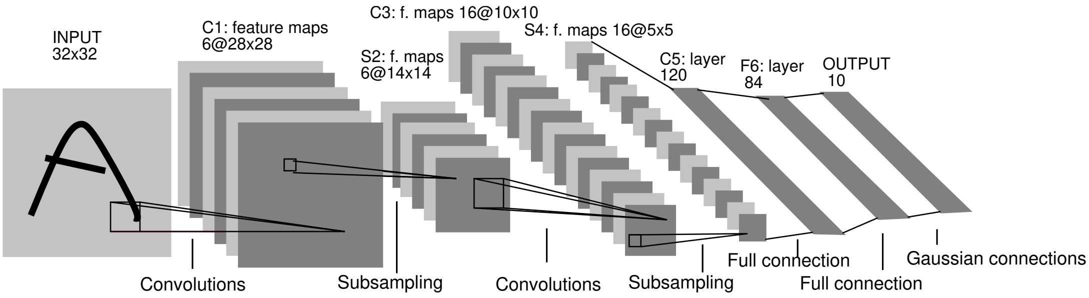


Figure 2.2.: The LeNet-5 architecture by LeCun et al. [46]

sigmoidal activation functions [27]. Today's deep neural networks demonstrate the superiority of a multitude of layers and rectified linear activation functions in praxis [26].

Feed-Forward Neural Network Every neural network whose connections between its nodes do not form a circle is a Feed-Forward Neural Network (FFNN). Examples of simple FFNNs are the single and multi-layer perceptrons.

Recursive Neural Network The term Recursive Neural Network (RNN) describes a category of ANNs that use the output of hidden layers or the network itself as an input for the same or lower layers in the next step. RNNs reach from simple Elman [23] or Jordan networks [37] to complex networks such as the Long Short-Term Memory [34].

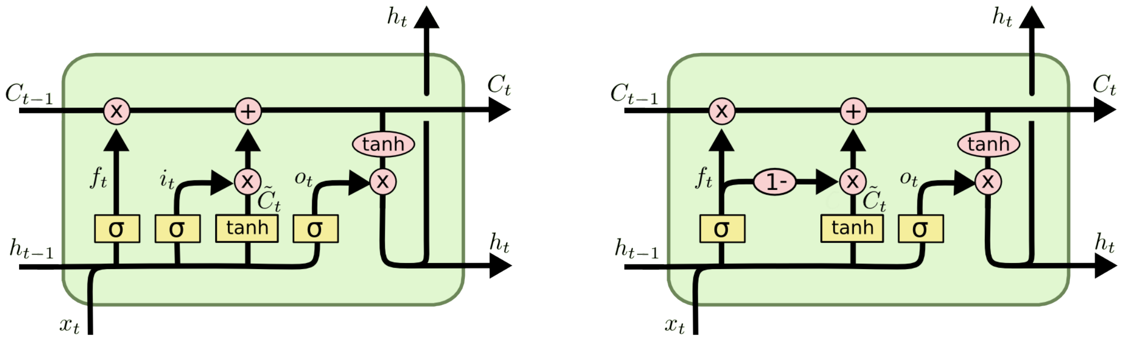
Convolutional Neural Network Waibel et al. introduced the Time Delay Neural Network (TDNN) as one of the first convolutional networks in 1987 [91]. TDNNs are shift-invariant in the time dimension by sharing the weights across that dimension and averaging over the gradients for each timestep of a context window before updating the network.

The idea of shift-invariance of TDNNs was also applied to computer vision to be invariant against translations. Unlike time invariance, handling translations in 2D images requires two-dimensional convolutional neural networks. Today the term CNN primarily refers to the two-dimensional convolutional neural networks, which benefits have long gone beyond the field of computer vision.

The convolutional network LeNet-5 was the first of its kind when proposed in 1998 [46]. However, its structure shown in Figure 2.2 can still be found in many CNNs today. Typically, a CNN consists of an alternating series of convolution and pooling layers followed by a fully connected FFNN that provides the network's final output. In the convolution layers, kernels slide over each input channel, creating the so-called feature maps. The convolution does not necessarily reduce the input's size in terms of width and height but might add more channels when multiple kernels are used. The pooling layers reduce the size of the feature maps by applying strategies like max pooling or average pooling. Before passing the data into the FFNN, it is flattened into a one-dimensional vector.

2.1.1. Long Short-Term Memory

The Long Short-Term Memories (LSTM) tackled the shortcomings of previous RNNs [34]. The main problem of these RNNs was their inability to store information for long and therefore were unable to learn long-term dependencies. LSTMs approach that issue by not only carrying



(a) Standard LSTM Cell

(b) LSTM Cell with tied Gates

Figure 2.3.: A standard LSTM cell and a variation with tied forget and input gates. Modified graphics from Olah [62].

a hidden state h_t through time but also a cell state C_t, which can be interacted with exclusively through three gates.

The input for all gates is the concatenation of the hidden state h_{t-1} and the current input x_t. A visualization of an LSTM cell is given in Figure 2.3a. The forget gate is a sigmoid layer that creates a vector f_t where values close to 0 indicate to forget the corresponding values in the cell state C_{t-1} and values close to 1 to keep the values.

To update the cell state C_{i-1} a new intermediate cell state $\tilde{C}_t$ is calculated by an tanh layer in the input gate. Similar to the forget gate a vector i_t is created by an sigmoid layer. i_t indicates what parts of $\tilde{C}_t$ are written to C_{t-1}.

The output gate consists of another sigmoid layer. The resulting vector o_t defines what parts of the new cell state C_t become part of the new hidden state h_t.

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
\tilde{C}_t &= tanh(W_C \cdot [h_{t-1}, x_t] + b_C)
\end{aligned}
$$

In order to update to old cell state C_{t-1} to the new cell state C_t the resulting vector of the forget gate f_t is multiplied element wise with C_{t-1} and the intermediate cell state $\tilde{C}_t$ is multiplied with i_t and added. To create the new hidden state h_t the output gate vector o_t is multiplied with the tanh of C_t.

$$
\begin{aligned}
C_t &= f_t \odot C_{t_1} + i_t \odot \tilde{C}_t \\
h_t &= o_t \odot \tanh C_t
\end{aligned}
$$

A common variation of the LSTM also used in some models in this thesis combines the forget gate and input gate, as shown in Figure 2.3b. Therefore, forgetting and adding information to the cell state are mutually dependent. The combination of C_{t-1} and ${}_t$ into the new cell state C_t only depends on f_t.

$$
C_t = f_t \odot C_{t_1} + (1 - f_t) \odot \tilde{C}_t
$$

2.1.2. Transformer Network

The transformer model was proposed in 2017 [87] and facilitated a faster, highly parallel computation in comparison with previous RNNs and handles long-range dependencies even better than the LSTM.

Attention Although LSTMs improved upon simple RNNs in handling long-range dependencies and avoiding vanishing or exploding gradients, when used in a typical encoder-decoder setup for tasks like NMT, all information of the source sentence has to be squeezed into a single fix-sized vector. That harms performance, especially on longer input sentences [12]. Bahdanau et al. introduced the concept of an attention mechanism [2] that should overcome this shortcoming. They not only pass a single state through the decoder but also provide a context vector c_i for every decoding step i, which is a weighted sum of the encoder's hidden states h_j.

$$c_i = \sum_j \alpha_{ij} h_j$$

The attention weights α_{ij} are obtained by applying the softmax function to the attention scores e_{ij}. In this case, scoring is done by a feed-forward neural network with the previous decoder state s_{i-1} and the corresponding hidden state h_j as inputs.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

$$e_{ij} = \textsc{ffnn}(s_{i-1}, h_j)$$

Even though the attention mechanism was widely adopted, in most cases, it was used in conjuncture with LSTMs or other types of RNNs until the introduction of the transformer model, which relied entirely on the attention mechanism and allowed highly parallel computing resulting in fast computation [87].

Transformer Network The transformer network not only uses the attention mechanism to transfer information between the encoder and decoder components but also within the components, called self-attention. Figure 2.4 shows a complete overview of the transformer network.

Since every input word is processed in parallel and there is no state anymore, which is passed through the time, an additional positional encoding must be added to the encoder's and decoder's input to enable decisions based on the position of the words. A function of interfering sine and cosine functions is used as positional encoding. The encoder and decoder consist of six stacked blocks, of which each includes multi-head attention and feed-forward layers. Residual connections skipping each layer allow the gradient to flow directly through the network and thereby helping to train the model. A layer normalization is subsequently applied to the sum of the outputs and inputs provided by the residual connections.

The first layer of an encoder block is a multi-head self-attention layer processing the input data. The multi-head attention shown in Figure 2.5a has three input matrices - the value V, key K, and query Q consisting of the input vectors. In the case of a self-attention layer, V, K, and Q are identical. All three are multiplied with the corresponding weight matrices W^k, W^q, or W^v, independently learned for each attention-head. The products are used for the scaled-dot-product attention, and the results for each head are concatenated. In the end, the

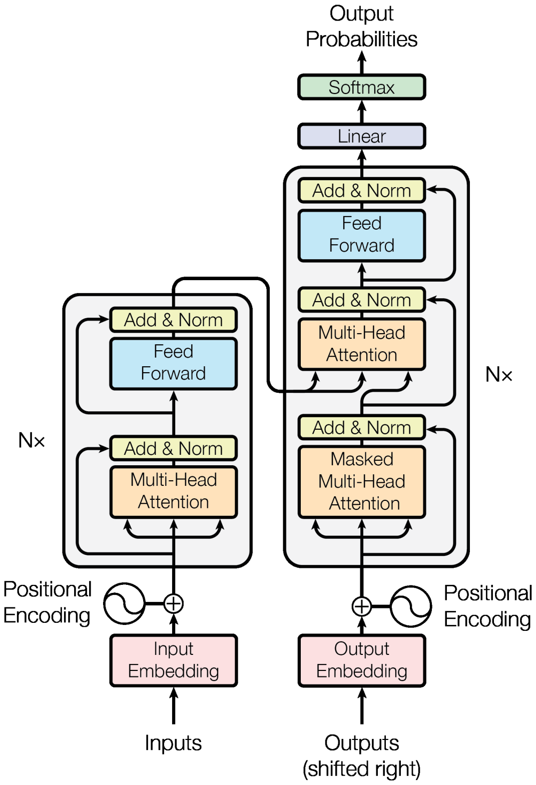


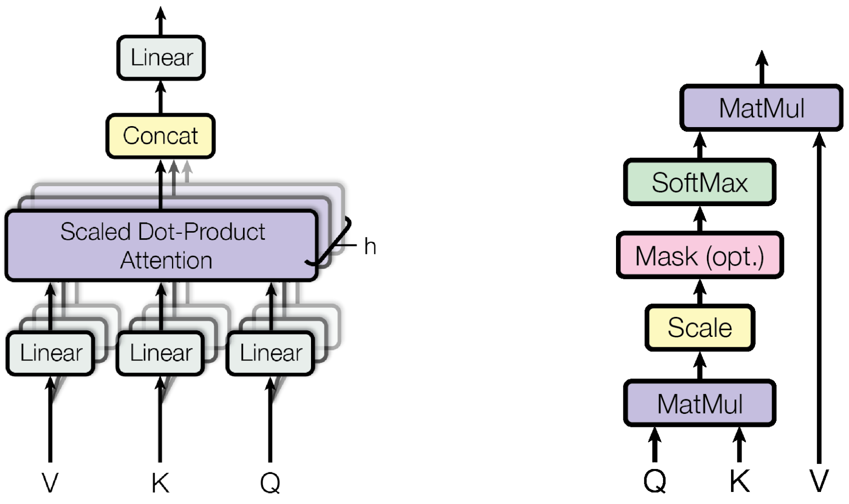
Figure 2.4.: A high-level overview of the complete transformer network [87].

concatenated results are passed into another linear layer in order to squeeze the result into a matrix with the same dimensionality as the value input matrix.

Figure 2.5b shows the schematic structure of the scaled dot-product attention used in each attention head. The inputs Q, K, and V can be logically split up into the query vectors q_i, the key vectors k_i, and the value vectors v_i corresponding to the sequence position i. The attention score a_{ij} is the dot product of q_i and k_j and defines how much attention is given to v_j for the resulting vector z_i. For stability reasons, the score is divided by the square root of the size of the key vector d_k, and the softmax is applied to get the probability distribution. The final vector z_i is then defined as the sum of v_j weighted by the scaled scores a_{ij} for all j. In praxis, K, Q, and V are not split up, and all steps can be done by matrix operations:

$$\text{Attention}(K, Q, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V$$

The self-attention layer in the encoder block is followed by a feed-forward layer. In the decoder, the self-attention layer is a masked multi-head layer. Except for the right-shifted decoder output, all future positions are masked by setting them to negative infinity before applying the softmax of the scaled-dot-product attention. The decoder block contains an additional attention layer that uses the output of the encoder as key and value. The last layer of the block is a feed-forward layer, just as with the encoder's block. After the repeating decoder blocks, a linear layer outputs a score vector, and the final probability distribution over the dictionary is given by a softmax layer.



(a) Multi-Head Attention

(b) Scaled-Dot-Product Attention

Figure 2.5.: Visualization of the multi-head attention mechanism on the left. Detailed view on the scaled-dot-product attention on the right [87].

2.2. Word Embeddings

To make text digestible for neural networks, words are represented by vectors of real numbers. Besides general approaches to map categorical features to vectors, word embeddings aim to reduce the dimensionality and encode semantic information into the vector embeddings.

Vector Encodings One of the simplest ways to represent categorical features as a vector are vector encodings like the one-hot-encoding or the dummy-encoding. Besides every kind of categorical data, vector encodings are applicable to text as well. For a dictionary of N words w the one-hot-encoding results in a vector v of size N with all elements being zero except for v_i representing the actual word w_i. Dummy-encoding is very similar but additionally also assigns the zero vector to a word, resulting in a vector size of only $N - 1$. However, these vector encodings are impractical for most problems with text input since dictionaries are large, and the model is suffering from the curse of dimensionality [5].

2.2.1. Static Word Embeddings

Vector encodings like the one-hot-encoding enforce equally distanced representations. However, it is evident that some words are semantically closer than others. Word embeddings aim to learn vector representations encoding the semantics of the words rather than just mapping them to equally distanced unique identifiers. Therefore the embeddings of similar words or words used in the same context are closer than others. Static word embeddings are trained once on vast amounts of text data and assign a single fixed-sized vector to each word in the training corpus. Two of the best-known representatives of these kinds of embeddings are Word2Vec [57] and GloVe [63].

Word2Vec was proposed in two different flavors: The skip-gram model and the continuous bag of words model (CBOW). While the CBOW model tries to predict a target word based on the surrounding context, the skip-gram model predicts context words for the given target word. A simple neural network is used for prediction for both models and simultaneously learns the word representations.

In contrast to the predictive Word2Vec model, GloVe is a count-based model using global statistics. The objective of GloVe is to learn a factorization of the global word-to-word co-occurrence matrix into two embedding matrices, the target, and the context matrix. Since the role of target and context is arbitrary in this model, either matrix represents a valid word embedding, and due to performance reasons, the final embedding is given by the sum of both.

2.2.2. Contextual Word Embeddings

One significant downside of static word embeddings is that each word is mapped to a single vector no matter its context. Especially words with different meanings based on the context suffer from this limitation. Contextualized word embeddings overcome this problem by providing an encoding model taking the actual context into account rather than a static vector representation. That model is applied during interference and outputs an embedding based on the word itself as well as its context.

ELMo The Embeddings from Language Model (ELMo) are based on a multi-layered bi-directional LSTM and trained on the language modeling (LM) objective [64]. The forward LSTM predicts the next word in a word sequence based only on the previous words, while the backward LSTM conversely predicts the previous word based on the following sequence. The LM objective maximizes the log-likelihood of the predicted sequence in both directions:

$$\sum_{k=1}^{N}\left[\log p(t_k \mid t_1,\ldots,t_{k-1};\Theta_x,\overrightarrow{\Theta}_{LSTM},\Theta_s)+\log p(t_k \mid t_{k+1},\ldots,t_N;\Theta_x,\overleftarrow{\Theta}_{LSTM},\Theta_s)\right]$$

The parameters of the token representation Θ_x and the softmax layer Θ_s are shared between the forward and backward LSTMs while the parameters of both are independent.

The context-independent initial embedding x_k for token k is derived from other neural language models either by token embeddings or by a character-based CNN helping the model to handle unknown words. To form the ELMo embedding, the outputs of the hidden LSTM layers of both LSTMs and the initial embeddings are combined. Let $\overrightarrow{\mathbf{h}}_{k,j}^{LM}$ be the output of layer j of the forward LSTM for the token k and let $\overleftarrow{\mathbf{h}}_{k,j}^{LM}$ the corresponding output in the backward LSTM, then the outputs of the hidden layers of the bi-directional LSTM are defined as the concatenation $\mathbf{h}_{k,j}^{LM} = [\overrightarrow{\mathbf{h}}_{k,j}^{LM}; \overleftarrow{\mathbf{h}}_{k,j}^{LM}]$ with $\mathbf{h}_{k,0}^{LM} = x_k$. The combination of these outputs into the final ELMo embedding of token k is defined as follows:

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

With $\mathbf{s}^{task}$ being task-specific softmax weights to collapse the representations from each layer into a single weighted sum and γ^{task} being a task-specific scalar scaling the whole embedding.

BERT In contrast to ELMo, Devlin et al. [20] utilized the transformer model [87] to pre-train their language model. The Bidirectional Encoder Representations from Transformer (BERT) are derived from the hidden states from the last of multiple stacked encoder blocks identical to those used in the transformer model. While ELMo obtains its bidirectionality by concatenating the layer outputs of a forward and backward LSTM, BERT inherits that property from the transformer models encoder. However, since the encoder processes the whole sentence at once, the LM objective of ELMo is not suitable. The encoder's self-attention is not masked, and all input tokens are known to the network.

While ELMo and other approaches like the OpenAI GPT [70], which relied on the transformers decoders, trained on the LM objective simply predicting the next word in a text, BERT introduces two new pre-training tasks. The Masked Language Modeling (MLM) objective requires to predict the original tokens, given tokens masked in the input. 15% of the tokens are chosen to be masked before being fed into the model. However, only 80% of these are actually replaced with the `[MASK]` token. The remaining tokens are replaced by other random tokens or not replaced at all to solve the mismatch between the pre-training and the fine-tuning on the downstream task during which no masked tokens are involved.

Many downstream tasks require an understanding of the relation of multiple sentences. That is why BERT is pre-train on the Next Sentence Prediction (NSP) task as well. Two sentences separated by a `[SEP]` token are fed into the BERT model. The encoder's output for the `[CLS]` token, which is added in front of every input sequence, is used to decide whether or not the two sentences are next to each other in the original document.

2.3. Coreference Resolution Terminology

For the sake of simplicity and comprehensibility, coreference resolution and many other linguistic terms are often used ambiguously, interchangeably, or inaccurately, especially in the context of NLP. In the following, these terminologies are put into the linguistical context, and subsequently, rules for their meaning in this thesis are defined.

Mentions Linguistic expressions referring to a discourse entity are called mentions [40]. Mentions can reach from single words to longer spans of text containing multiple mentions themselves.

Coreference and Coreference Resolution Mentions that refer to the same discourse entity are coreferent [40]. In Example (1) *the trophy* and *it* refer to the same entity while *the suitecase* describes a different entity. Therefore the former two mentions corefer.

(1) **The trophy** does not fit into the suitcase because **it** is too big.

Coreference Resolution describes the task of determining which mentions in a text are coreferent [60]. It can also be perceived as a clustering task, grouping the mentions according to the entities they refer to.

Singletons An entity referenced by only a single mention is called a singleton [40]. Singletons often form an edge case in the build of up coreference clusters. Reported performances of coreference resolution systems depend heavily on whether or not singletons were annotated in the corpora and how they were handled during evaluation.

Anaphora and Antecedent Anaphora describes the reference to a precedent expression, the so-called antecedent. The referring anaphor or anaphoric expression can be a pro-form or any kind of deictic [40]. In Example (2), the anaphor *it* refers back to the antecedent *the trophy*.

(2) $\underline{\text{The trophy}}$ does not fit into the suitcase because **it** is too big.

The term anaphora is often used in a broader sense, including references to succeeding expressions.

Cataphora and Postcedent When used in a narrower role, cataphora is the opposite of the anaphoric reference. The cataphoric expression refers to a succeeding expression that introduces the entity to the discourse. In that case, the antecedent is sometimes called postcedent [83]. In Example (3) the postcedent *his* refers to the cataphor *Peter* later in the sentence.

(3) With **his** speech, Peter tried to convince the people.

Due to its rare use in written and spoken language, most approaches do not consider cataphora or handle it by supposing the broader sense of anaphora.

Anaphora and Coreference resolution Often anaphora resolution and coreference resolution are used synonymously; other times, the former is referred to as a subcategory of the latter or vice versa. Sukthanker et al. argue that neither assumption is correct, albeit there is a reasonable larger intersection of both concepts [83]. While coreference partially includes cataphora and therefore is no subset of anaphora, Example (4) gives an example of an anaphoric reference that is not coreferent. In the case of coreference, *every scientist* would have to do his own research and the research for every other scientist.

(4) Every scientist does **his** research.

In this thesis, I focus on coreference resolution rather than anaphora resolution. However, as Sukthanker et al. mention, many publications are not precise on the terminology and what kind of references they consider coreferent. This includes the CoNLL-2012 shared task, the standard benchmark for coreference resolution [83].

In line with recent work on coreference resolution like the end-to-end model by Lee et al. [49], I will term every preceding mention a possible antecedent or antecedent candidate, whether it is an anaphoric or cataphoric reference.

2.4. Coreference Resolution Approaches

Over the years, a large variety of different coreference resolution systems have been introduced, yet many of them share the same basic concepts. Therefore, many surveys about coreference resolution try to work out a systematic overview of the different approaches. While some authors break them down into several subcategories [40], others display various aspects in which these systems differ in a single list [60]. In the following, I classify the approaches into three different categories:

- The method describes in which way knowledge about coreference resolution is gathered and integrated into the model. This reaches from entirely manually created rules to self-learning systems.

- The model type summarizes the ideas on how to evaluate single coreference candidates and how to cluster the coreferent mentions.

- The aspect of mention detection divides systems into the ones depending on dedicated systems and those that combine mention detection and coreference resolution in a single system.

2.4.1. Methods

Rule-based Coreference Resolution Rule-based coreference resolution systems were the starting point for automated coreference detection. Hobbs algorithm [33] in 1978 was one of the first approaches to resolving pronoun references. A breadth-first search through the syntactic parse tree of a sentence, led by simple rules regarding the number, gender, and person of the found nouns. Nevertheless, it was not too long ago that rule-based systems played a competitive role in coreference resolution. One of the best-known examples is the Stanford resolver consisting of 12 sieves, which consecutively search for coreferences, starting with the sieves with the most robust rules [47]. Since other approaches outperform rule-based systems today, they only play a relevant role in languages with no large coreference annotated corpora [83].

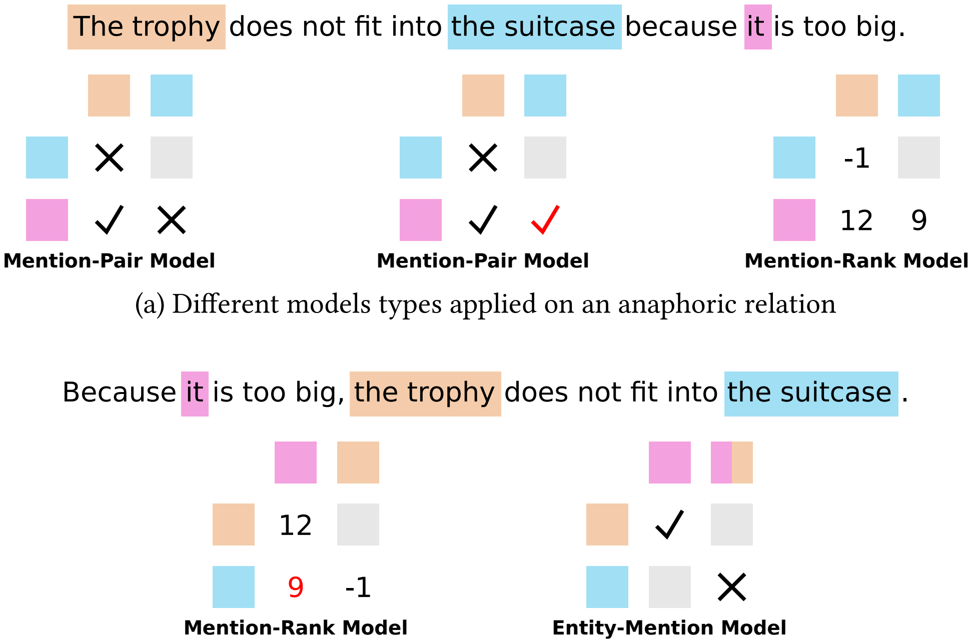
Feature-based Coreference Resolution Also called sometimes statistical or machine learning approach. Particularly earlier feature-based systems did not rely solely on statistical or learning models but still incorporated hand-crafted rules in many cases. Learning systems were enabled by emerging annotated coreference corpora, which were not needed for rule-based systems [83]. The performance of these systems is heavily dependent on the quality of the manually created features. Different types of features are supposed to reflect the essential characteristics of the anaphor, the candidate antecedent, and their relationship. Additional types were used for entity-based models or to describe the document genre [40]. Feature-based coreference resolution systems were the leading systems up to the emergence of deep learning models making use of pre-trained word embeddings.

Deep Neural Coreference Resolution Today's best-performing coreference systems are based on deep neural networks that depend on at most a few manually engineered features such as mention distance or the document genre. Powerful contextual word embeddings and big annotated corpora enable these systems to find coreferences directly from the text.

2.4.2. Model Types

Model types differ in the way they decide on mentions being coreferent or not and how they build up clusters of coreferent mentions. Although models can find possible coreference candidates in different manners, for the sake of clarity, I assume a uniform strategy common today. The system goes through the mentions from the beginning of the text to the end and considers every preceding mention being possibly coreferent or an antecedent candidate. Figure 2.6 visualizes how the three most common model types decide over the coreference of the antecedent candidates found this way. Besides the correct solution process for each model type, common problems of different types are also demonstrated, motivating more advanced approaches.

Mention-Pair Model The mention-pair model is one of the most straightforward approaches, yet it is also one of the most widely used models. It is based on a binary classifier that classifies each mention pair as coreferent or not, independently of other mentions, mention pairs, or


The trophy does not fit into the suitcase because it is too big.
× ✓ ×
Mention-Pair Model
× ✓ ✓
Mention-Pair Model
-1 12 9
Mention-Rank Model
Because it is too big, the trophy does not fit into the suitcase .
12 9 -1
Mention-Rank Model
✓ ×
Entity-Mention Model

(a) Different models types applied on an anaphoric relation

(b) Different models types applied on a cataphoric relation

Figure 2.6.: Visualization of the decision making of different coreference resolution model types. The models go through the mentions in the sentence (rows) and evaluate possible coreferences towards the preceding mentions (columns). Mistakes made by the models are highlighted in red.

earlier decisions only given local information about both mentions and their relation. Many mention-pair approaches use explicit clustering algorithms to build up coreference chains from the independent decisions on the individual mention pairs. Well-known clustering algorithms are the closest-first clustering, which simply uses the first antecedent found, or the best-first clustering, which introduces simple rules to decide which of the found antecedent is chosen to continue the coreference chain [40]. Since coreference is an equivalence relation and therefore transitive, the final clusters are given by the transitive closer of relations found by the mention-pair model or the clustering algorithms.

Mention-Ranking Model Due to its independent evaluation of each antecedent candidate the mention-pair model is prone to wrongly resolve coreferences as shown in Figure 2.6a. Mention-ranking models on the other hand do not evaluate mention pairs independently but assign a score to all candidate antecedents that indicate how likely a coreference relation between the candidate and the given mention is. Only the highest ranked candidate is considered to be the correct antecedent, making a dedicated clustering algorithm obsolete. The final clusters are formed by the transitive closure of the coreferences found as described above. However, the ranking algorithm does not inherently cover non-anaphoric mentions, which are not coreferent to any of the preceding mentions. One solution to solve this problem is the introduction of a dummy antecedent or a lower bound for the score, which prevents the mention from being assigned to an antecedent candidate despite a low score [40]. Another option is to apply an independently trained anaphoricity classifier before ranking [60].

Entity-Based Model Entity-based models aim to overcome the shortcomings of evaluating every mention pair individually, like the mention-pair model, as well as selecting the best antecedent without consideration of earlier decisions like the mention-ranking model. Therefore, it does not try to find coreferences between a mention and a single antecedent but to link every mention to a discourse entity - a cluster of antecedents. The current discourse entities are given by the transitive closures of the coreferences found so far. In Figure 2.6b the model decides that the first two mentions are coreferent. Hence, a possible coreference between the third mention and the first or second is not considered individually. Instead, the third mention is considered as part of the discourse entity formed by the first and second mentions. Entity-based models based on binary classifiers like mention-pair models are called entity-mention models, whereas entity-based models that act like mention-ranking models are called entity-ranking or cluster-ranking models [60][40].

2.4.3. Mention Detection

Pipeline Approach Especially older feature-based approaches rely on dedicated mention detection systems since it is not feasible to consider every span of text as a possible mention, and they are unable to find good mention candidates by themselves. These coreference systems can be trained with manually annotated or automatically detected mention boundaries but have to be evaluated on the latter ones to be comparable with the end-to-end models.

End-To-End Model In contrast to approaches based on pipelines, end-to-end systems combine mention detection and coreference resolution in a single model. That eliminates the risk of cascading errors introduced by the previously trained mention detection mistakes. Instead, mention detection and coreference resolution are trained simultaneously, and mistakes made by either of the two functions should lead to weight changes at the responsible layers. Even though not the first, one of the most common end-to-end models is the end-to-end neural coreference system by Lee et al. [49].

2.5. Coreference Resolution Metrics

Despite newer metrics like BLANC [72] and LEA [58, 54], which overcome some shortcomings of their predecessors, the scorer used in the CoNLL-2012 Shared Task [69] is still the most used and widely accepted benchmark to compare coreference resolution systems. Therefore I will apply this very scorer for the evaluation of my implementations. The final score is defined as the mean of the F1 scores of the three metrics MUC [88], B-CUBED [1], and CEAF [53]. This approach is based on the MELA metric [19] and only differs from MELA by using the entity-based version of CEAF instead of the mention-based version.

In the context of these coreference resolution scoring schemes, the gold clusters and the predicted clusters are referred to as key K and response R. Often a total order of the mentions in the entities K_i or R_i is assumed. Therefore the clusters are also called key-chain and response-chain. However, since the following metrics only take the number of links or mentions within a key entity, response entity, or an intersection of both into account, no total order is required.

2.5.1. MUC

The MUC scoring scheme was proposed as a scoring metric for the shared task of the Sixth Message Understanding Conference in 1995 [88]. It is a link-based metric and therefore defines the precision and recall as the ratio between correct links found by the system and the minimum number of links to recreate the key chain, respectively the total number of links predicted by the system:

$$Precision = \frac{\sum_i |R_i| - |p'(R_i)|}{\sum_i |R_i| - 1}$$

$$Recall = \frac{\sum_i |K_i| - |p(K_i)|}{\sum_i |K_i| - 1}$$

The key and response entities are given by K_i and R_i. Obviously, their cardinality equals the number of links in that very entity plus one. The function $p(K_i)$ maps a key entity to a set of partitions induced by the intersection of K_i with the response entities. Conversely, $p'(R_i)$ maps a response entity to a set of partitions induced by the intersection of R_i with the key entities. The cardinality $|p(K_i)| - 1$ corresponds to the number of missing links in the response whereas $|p(R_i)| - 1$ equals the number of false links in it.

2.5.2. B-CUBED

The mention-based B-CUBED overcame some flaws of MUC, like the inability to handle singletons or the uniform penalization of linking errors without considering the impact on the resulting clusters [1]. The idea of B-CUBED is to average over the precision and recall calculated for each mention. Therefore, the number of common mentions $|K_{m_i} \cap R_{m_i}|$ between the key entity K_{m_i} and the response entity R_{m_i}, which contain the mention m_i, is put in relation with the total number of mentions in K_{m_i} and R_{m_i}. When precision and recall are calculated over all mentions, the resulting averages can be simplified as follows:

$$Precision = \frac{\sum_i \sum_j \frac{|K_i \cap R_j|^2}{|R_j|}}{\sum_j |R_j|}$$

$$Recall = \frac{\sum_i \sum_j \frac{|K_i \cap R_j|^2}{|K_i|}}{\sum_i |K_i|}$$

2.5.3. CEAF

The CEAF metric aligns the key and response entities before calculating precision and recall so that each entity is only used once since Luo's main criticism of B-CUBED was its multiple use of single entities [53]. The alignment is based on a similarity metric for two entities ϕ in which the two variants of CEAF differ. For the entity-based version of CEAF, used for the CoNLL-2012 shared task [68] and in this thesis, that metric is given by:

$$\phi(K_i, R_j) = \frac{2|K_i \cap R_i|}{|K_i| + |R_j|}$$

To find the actual mapping between key and response entities the Kuhn-Munkres algorithm is applied with the goal of maximizing the similarity over all entity pairs. According to this mapping, $g^*(K_i)$ maps K_i to an response entity R_j in the following:

$$Precision = \frac{\sum_i \phi(K_i, g^*(K_i))}{\sum_i \phi(R_i, R_i)}$$

$$Recall = \frac{\sum_i \phi(K_i, g^*(K_i))}{\sum_i \phi(K_i, K_i)}$$

2.5.4. BLANC

The BiLateral Assessment of Noun-phrase Coreference metric (BLANC) is an adaptation of the RAND index to the coreference resolution problem [72]. RAND is a similarity measure for general clusters. Since it is related to the accuracy, it is not suitable for highly imbalanced data and, therefore, not applicable to this problem. The precision and recall are calculated by averaging over the independent scores of coreferent and non-coreferent mention-pairs:

$$Precision = \frac{1}{2}\left[\frac{rc}{rc+wc} + \frac{rn}{rn+wn}\right]$$

$$Recall = \frac{1}{2}\left[\frac{rc}{rc+wn} + \frac{rn}{rn+wc}\right]$$

with rc and rn as the number of correctly predicted coreferent and non-coreferent pairs as well as wc and wn as the number of mention pairs wrongly predicted as coreferent or non-coreferent.

2.5.5. LEA

Moosavi and Strube showed flaws in the interpretability of B-CUBED, CEAF, and BLANC [58]. By manipulating the key and response in various ways, the evaluation score of those metrics changed counterintuitively. Since MUC, the only metric robust against these flaws, lacks discriminative power, they introduced the interpretable and discriminative Link-based Entity-Aware evaluation metric (LEA).

LEA assigns a score to each key and response entity based on how well it is resolved and weighted by its importance. The importance is given by the size $|K_i|$ or $|R_j|$ of each entity. The second part of the numerator describes the resolution by counting the number of unique links in the intersection with a corresponding key or response entity. The precision is defined as the sum of the scores for all response entities, whereas the recall is derived from the scores of all key entities.

$$Precision = \frac{\sum_i \sum_j |R_j| \times \frac{link(R_j \cap K_i)}{link(R_j)}}{\sum_j |R_j|}$$

$$Recall = \frac{\sum_i \sum_j |K_i| \times \frac{link(R_j \cap K_i)}{link(K_i)}}{\sum_i |K_i|}$$

With $link(e) = n \times (n-1)/2$ being the number of unique coreference links in the entity e consisting of n mentions.

3. Literature Review

3.1. Coreference Resolution

Coreference resolution is a long-established task in the field of machine learning. Comprehensive summaries of the last decades of coreference resolution research can be found in [61] and [60]. Fundamental terminology and concepts of coreference resolution are defined in [40, Chapter 21], which additionally gives a brief overview of the linguistic background. More linguistic context and a discussion about the differences between anaphora and coreference resolution, which details are often suppressed for the sake of simplicity, are provided in [83]. Its outline of different coreference systems ranges from very early approaches to the first neural end-to-end coreference resolution system by Lee et al. [49], which is the starting point for this thesis.

The end-to-end model by Lee et al. [49] and the models based on it - the higher-order model [48] and the BERT-based model [38] - form the basis of the experimental part of this thesis. To the best of my knowledge, there is one PyTorch implementation of the BERT-based model with corresponding scientific work [99]. Other PyTorch implementations of that model and its two predecessors lack a sufficient reporting of their performances and are not described in the scientific context.[1]

3.2. German Coreference Resolution

As for English, there are also large German corpora with annotated coreference information [84], and therefore the problem of coreference resolution has been treated for a long time in German as well. Historically German coreference resolution systems were oftentimes adapted versions of English systems. A brief overview of German coreference resolution is described in [82].

More recent German models are the IMS HotCoref DE model [76] and the model proposed by Schröder et al. [81], a version of the BERT-based model [38] with multilingual and German embeddings similar to what is done in this work.

3.3. Multi- and Cross-lingual Coreference Resolution

While multilingual coreference resolution primarily refers to training the same model on multiple languages, cross-lingual coreference resolution describes the problem of applying a model on a language different from the language it was initially trained on. Although both approaches lead to a model suitable for multiple languages, they serve different purposes. The multilingual approach aims to improve the performance of a model trained on a single language only, whereas the cross-lingual approach is used to apply knowledge learned on a resource-rich language onto other languages with fewer or no annotated training data. Various

[1] `https://github.com/search?q=coref+pytorch&type=Repositories`

past tasks emphasized multilingual coreference resolution like the SemEval-2010 Task 1 [73] as well as the CoNLL-2012 Shared Task [68]. However, in this work, I will focus on cross-lingual coreference resolution.

In general cross-lingual coreference resolution can be divided into two kinds of approaches [24]. On the one hand, the projection-based approaches, which need to be trained on parallel corpora to transfer their knowledge into another language, on the other hand, approaches relying on a common multilingual feature space like my proposal in this thesis.

Using the multilingual, static word embedding FastText [28], Cruz et al. create a neural end-to-end coreference system for Spanish, which also delivers compelling results on a Portuguese test set [17]. FastText is utilized in [86] as well to create a coreference resolution system for Basque by training on larger English corpora. Kundu et al. [44] use word2vec [57] to achieve competitive results on Spanish and Chinese with their entity-mention model trained on English data.

Similar to this work a "first study on cross-lingual transfer learning for event coreference resolution" leveraging XLM-RoBERTa [15] and domain-adversarial training [25] is conducted in [65].

3.4. Machine Translation and Coreference Resolution

Coreference information was undoubtedly valuable in order to improve context agnostic NMT systems. However, with the upcoming of modern context-aware systems, the benefit of augmenting these systems with coreference information is in question. Multiple works try to enhance context-agnostic and context-aware MT systems with the help of coreference information, while others try to show that context-aware systems implicitly consider coreferences.

In [94] a re-rank and post-edit algorithm was introduced in order to improve the accuracy of the pronoun translation of a phrase-based statistical machine translation system. Only the post-edit approach shows a significant improvement, however. For each mention, multiple translation hypotheses are created. The combination of translation hypotheses that is most likely to be a coreference cluster on the target side is chosen as the translation for each cluster on the source side.

Hwang et al. use automatically detected coreferences to create contrastive examples by corrupting the coreferences [35]. By leveraging contrastive learning, they create multiple Transformer-based, context-aware NMT systems with better sentence representations regarding coreferences. Their systems improve over their counterparts not trained on contrastive examples in terms of BLEU score as well as in accuracy on the ContraPro test set [59].

Besides explicitly incorporating coreference information into MT systems, much work is done to push context-aware systems to make better use of the contextual information provided without focusing on a single discourse phenomenon. One example is the hierarchical attention network [95], which includes document-level contextual information and is reported to improve the noun and pronoun translation compared to a context-aware NMT Transformer. The CADec model proposed by Voita et al. [90], which I use to show the impact explicit coreference information can still have, is another example of such a system. The authors claim to improve translation consistency by implicitly considering different phenomena like deixis, ellipsis, and lexical cohesion - problems multiple other context-aware systems struggle with.

Different LSTM-based, context-aware architectures are compared with a context-agnostic baseline in [3]. Though most of their models performed close to the baseline, two context-aware models clearly outperformed the baseline. Since these two models could disambiguate pronouns, the authors conclude that these models must be able to use linguistic context. To show what context-aware systems are capable of, Voita et al. created a system, which they used to analyze the flow of information from the extended context to the context-aware translation model [89]. They were able to improve the BLEU score on sentences containing ambiguous pronouns over a context-agnostic and a simple context-aware baseline and showed that the model made use of anaphora relations.

4. Coreference Resolution with Pre-trained Language Models

In this chapter, I describe three of the most important models for coreference resolution in recent years. They build upon each other, starting with the first competitive neural end-to-end model, over the proposal of advanced pruning strategies, and finally leveraging pre-trained language models to further improve its performance. I provide a PyTorch implementation for each of these models and conduct experiments with a series of different language models as the foundation of the coreference system.

4.1. English Coreference Resolution

In 1995 the Sixth Message Understanding Conference (MUC-6) offered the first shared task tackling coreference resolution in order to standardize the evaluation of coreference systems [30]. The MUC evaluation metric introduced for that very first task is still used today. A second coreference task was presented for MUC-7. As part of the ACE program, multiple coreference datasets were released in the early 2000s [22]. They contained various languages and were more extensive than the MUC datasets. Therefore they were the de facto standard for coreference resolution in the 2000s, even though they restricted the task to specific entity types [60]. Today almost all coreference systems for English are trained and evaluated on the OntoNotes 5.0 corpus [93]. The CoNLL 2012 shared task defined the splitting into train, test, and validation data [68]. Its predecessor, the CoNLL 2011 shared task, introduced the CoNLL scoring scheme consisting of the MUC, B3, and CAFE metrics [67].

4.1.1. End-to-end Neural Coreference Resolution

The end-to-end model (*e2e-model*) by Lee et al. [49] was proposed as the first state-of-the-art neural coreference resolution model, which was trained in an end-to-end manner. Instead of relying on predefined mention boundaries, the idea is to consider every span of text up to a certain length as a possible mention. However, due to computational limitations, pruning has to be applied to the spans and the span pairs. The model can be divided roughly into two parts. The first one deals with single spans or mentions and the second one with pairs of mentions, trying to find the best antecedent.

Figure 4.1 shows the first part of the model, which generates an encoding for all spans that are potential mentions, and assigns a mention score to all of them. Depending on that score, the spans are ranked, and the top M spans are considered to be mentions in the following.

The initial word embedding x is a concatenation of a 300-dimensional GloVe with a windows size of 10 and a 50-dimensional Turian embedding, as well as a small character embedding generated by a CNN trained simultaneously with the coreference resolver. The character embedder is a simple convolutional network that is jointly trained with the *e2e-model*. In order to contextualize x, a bidirectional LSTM is applied, which outputs the contextual word

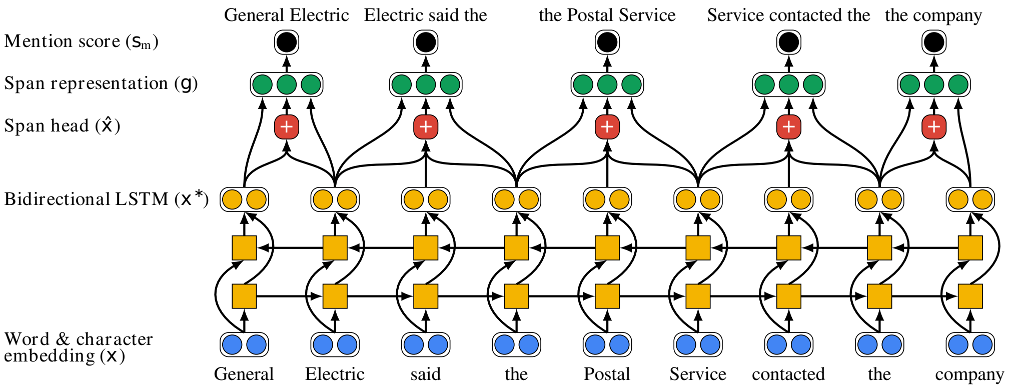


Figure 4.1.: Lower layers of the *e2e-model* encoding span representations and scoring mentions.

embedding x^*. This specific LSTM version works without an explicit input gate but derives its output from the forget gate so that no information is forgotten without replacing it with new data:

$$\begin{aligned}
f_{t,\delta} &= \sigma(W_f[x_t, h_{t+\delta,\delta}] + b_i) \\
o_{t,\delta} &= \sigma(W_o[x_t, h_{t+\delta,\delta}] + b_o) \\
\tilde{C}_{t,\delta} &= \tanh(W_c[x_t, h_{t+\delta,\delta}] + b_C) \\
C_{t,\delta} &= f_{t,\delta} \odot \tilde{C}_{t,\delta} + (1 - f_{t,\delta}) \odot C_{t+\delta,\delta} \\
h_{t,\delta} &= o_{t,\delta} \odot \tanh(C_{t,\delta}) \\
x_t^* &= [h_{t,1}, h_{t,-1}]
\end{aligned}$$

The direction of each LSTM is given by $\delta = \{-1, 1\}$. By using this bidirectional LSTM, the newly computed word representation x^* contains contextual information from previous words in the first and contextual information of the following words in the second half. The final span representation g_i of span i is given by:

$$g_i = [x^*_{\text{START}(i)}, x^*_{\text{END}(i)}, \hat{x}_i, \phi(i)]$$

It is a concatenation of the contextual embedding of the first word $x^*_{\text{START}(i)}$ and the last word oft the span $x^*_{\text{END}(i)}$, a head word representation $\hat{x}_i$ and $\phi(i)$, which encodes the width of the span. Many traditional coreference resolution systems rely on syntactical head words to represent mentions. Lee et al. introduce an attention mechanism to circumvent the need for a syntactical parser:

$$\begin{aligned}
\alpha_t &= w_\alpha \cdot \text{FFNN}_\alpha(x_t^*) \\
a_{i,t} &= \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp(\alpha_k)} \\
\hat{x}_i &= \sum_{t=\text{START}(i)}^{\text{END}(i)} a_{i,t} \cdot x_t
\end{aligned}$$

Their experiments show that the mechanism assigns high attention weights to words, such a parser would choose as head words [49]. To assign the mention score $s_m(i)$ to span i, a standard feed-forward neural network FFNN_m is applied to the span representation g_i:

$$\begin{aligned}
s_m(i) &= w_m \cdot \text{FFNN}_m(g_i) \\
s_a(i, j) &= w_a \cdot \text{FFNN}_a([g_i, g_j, g_i \odot g_j, \phi(i, j)])
\end{aligned}$$

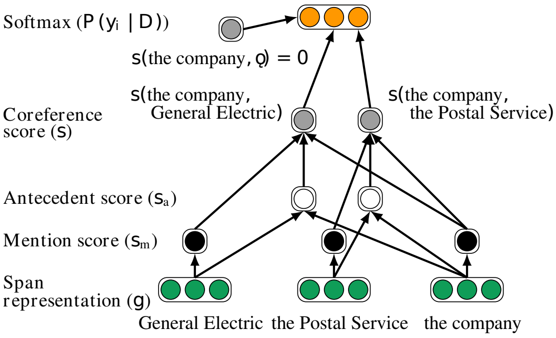


Figure 4.2.: Antecedent score between the anaphor and all antecedent candidates is together with the mention score combined into the final coreference score.

Before calculating the antecedent scores s_a between each mention and its preceding antecedent candidates, due to computational constraints, the spans are pruned considering s_m, and the number of antecedent candidates per mention is limited as well. For a document of length D, the number of spans considered as mentions in the following computations is pruned to λD with $\lambda \in [0, 1]$. The maximal number of antecedents candidates for each span is limited to K. For the antecedent score $s_a(i, j)$ between the spans i and j the span representations of both g_i and g_j, the element-wise similarity $g_i \odot g_j$ and an additional vector $\phi(i, j)$ encoding the genre, the distance between both spans and if they were expressed by the same speaker are fed into a feed-forward neural network $\textsc{ffnn}_a$.

Figure 4.2 shows how the mention and antecedent scores are combined into a probability distribution over all antecedent candidates y_i for span i. In praxis, the softmax is dismissed, and the antecedent candidate with the highest coreference score is chosen. This final score $s(i, j)$ for each span-pair is defined as the sum of the mention scores $s_m(i)$ and $s_m(j)$ of both spans as well as the antecedent score $s_a(i, j)$ between both. In order to be able to handle the first mention of an entity, non-anaphoric mentions and spans that are no mentions at all, a dummy antecedent ϵ with a fixed score of zero is introduced.

$$s(i, j) = \begin{cases} 0 & j = \epsilon \\ s_m(i) + s_m(j) + s_a(i, j) & j \neq \epsilon \end{cases}$$

To train the model, the log-likelihood of all antecedent candidates y_i, which are also in the same gold cluster as i, is optimized. The probability distribution $P(y_i)$ is given by the softmax over the coreference scores of between i and y_i:

$$\log \prod_{i=1}^{N} \sum_{\hat{y} \in \mathcal{Y}(i) \cap \mathrm{GOLD}(i)} P(\hat{y})$$

The model achieved a record-breaking average F1 score on the CoNLL-2012 metric of 67.2, leading to the breakthrough of neural end-to-end coreference resolvers.

4.1.2. Higher-order Coreference Resolution with Coarse-to-fine Inference

With their higher-order model with coarse-to-fine inference (*c2f-model*) Lee et al. [48] improved their own *e2e-model* in mainly two points. First, they reduce the errors due to the independent scoring of each span pair by introducing a higher-order architecture and secondly by allowing to link spans of any distance thanks to the replacement of the simple antecedent pruning by a more advanced pruning strategy utilizing a coarse but fast antecedent scoring.

(1) **[It]** is too big. **[The trophy]** does not fit into **[the suitcase]**.

Higher-order Coreference Resolution Sentence (1) shows an example first-order coreference resolvers can struggle with. That system might find the links [*it*, *the trophy*] and [*it*, *the suitcase*]. Even though it is clear for a human that the second one is wrong, it might be hard for a machine system to decide that *the suitcase* is non-anaphoric. On the other hand, it should be obvious to the system that the mention pair [*the trophy*, *the suitcase*] is not coreferent. By defining the coreference clusters as the transitive hull of its initial decisions, however, *the trophy* and *the suitcase* would end up in the same cluster.

This problem can be overcome by entity-mention models, as shown in Section 2.4.2. Lee et al. solve that problem by introducing a higher-order inference to the mention-ranking *e2e-model.* Therefore the span representation g_i is being refined in N iterations with g_i^n denoting the representation of span i at iteration n. As a result, the probability distribution $P(y_i)$ differs in each iteration:

$$P_n(y_i) = \frac{\exp(s(g_i^n, g_{y_i}^n))}{\sum_{y \in \mathcal{Y}(i)} \exp(s(g_i^n, g_y^n))}$$

While g_i^1 is identical to g_i in the *e2e-system*, the span representations g_i^{n+1} for the following iterations are defined as the interpolation of the previous representation g_i^n and the expected antecedent embedding a_i^n. The expected antecedent embedding is defined as the sum of the embeddings g_{y_i} of all antecedent candidates y_i for span i weighted by the probability $P_n(y_i)$.

$$a_i^n = \sum_{y_i \in \mathcal{Y}(i)} P_n(y_i) \cdot g_{y_i}^n$$

For the element-wise interpolation, a layer-independent forget gate W_f is trained. The concatenation of the representations of the span g_y^n and its expected antecedent a_i^n is used as input to the forget layer.

$$\begin{aligned} f_i^n &= \sigma(W_f[g_i^n, a_i^n]) \\ g_i^{n+1} &= f_i^n \odot g_i^n + (1 - f_i^n) \odot a_i^n \end{aligned}$$

Coarse-to-fine Inference In the *e2e-model* at most K previous mentions were considered as candidate antecedents for each mention, yet the distance between coreferent mentions in a text can become fairly large. However, considering all M mentions as possible antecedents is not feasible. Consequently, Lee et al. introduce the coarse-to-fine inference for pruning the candidate antecedent without relying on the distance and, therefore, theoretically enabling the system to handle coreference over an infinite distance. Before calculating the actual antecedent scores $s_a(i, j)$ between the span i and j, a less accurate but much faster to compute coarse score $s_c(i, j)$ is calculated for all span-pairs:

$$s_c(i, j) = g_i^\top W_c g_j$$

W_c is a learned weight matrix of the size $|g| \times |g|$. During the computation of the coarse scores, two matrices are manipulated. The interim result of $g^\top W_c$ is of the dimension $M \times |g|$ for M being the number of all possible mentions, and the final matrix of coarse scores is of the size $M \times M$. This is a significant improvement in comparison to the $M \times M \times (3 * |g| + |\phi|)$ input tensor which would be needed to calculate the antecedent score $s_a(i, j)$ for all span-pairs right away. Together with the mention scores $s_m(i)$ and $s_m(j)$ the coarse score sums up to the fast score $s_f(i, j)$ by which the antecedent candidates are ranked for each mention and subsequently pruned. The final score $s(i, j)$ of the *e2e-model* is made up of $s_f(i, j)$ and the more accurate antecedent score $s_a(i, j)$ calculated after pruning:

$$\begin{aligned} s_f(i, j) &= s_m(i) + s_m(j) + s_c(i, j) \\ s(i, j) &= s_m(i) + s_m(j) + s_c(i, j) + s_a(i, j) \end{aligned}$$

Even though the scoring in this higher-order model is computed in every iteration based on the changing span embeddings, the coarse score is only computed once and is fixed during the iteration, just like the mention scores. Therefore only $s_a(i, j)$ has to be calculated again in each iteration and added to the fixed fast score $s_f(i, j)$ in order to update the overall score.

Besides the higher-order inference and the coarse-to-fine antecedent pruning, a few more changes were added to the original implementation of the *e2e-model*. The Turian embedding as one part of the concatenated word embedding used as input to the LSTM is replaced by an extracted ELMo embedding. For the embedding of the head word of each span, a different, much smaller embedding is now used. Besides the same character embedding used in the LSTM input embedding, it contains a GloVe embedding with a smaller window size of two. Instead of a single-layer bidirectional LSTM, the *c2f-model* is based on a bidirectional highway LSTM with three layers and coupled gates. Due to the advanced pruning strategy, the number of antecedent candidates per span can be reduced from 250 down to only 50 without making sacrifices in performance. On the other hand, the maximal length for a span can be increased from 10 to 30.

The *c2f-model* achieves an F1 score on the Onto Notes 5.0 test set of 73 and outperforms its predecessor by almost six percentage points.

4.1.3. BERT for Coreference Resolution

For their BERT-based model (*bert-model*), Joshi et al. [38] reused large parts of the *e2e-model* as well as the higher-order approach and the coarse-to-fine inference of the *c2f-model* but replaced the initial embedding and the complete LSTM structure with a BERT-encoder which is fine-tuned on the coreference task during training.

The text must be split up during the preprocessing to match the chosen segment size, which is limited by the BERT model's configuration. Because of the possible lack of context at the end and the beginning of those segments, Joshi et al. introduce two different approaches to obtaining the contextual embeddings. The independent version uses non-overlapping segments and accepts the drawbacks of the lacking context. To tackle that problem, the overlapping version creates segments that overlap with each other, resulting in two embeddings for each token. A jointly trained feed-forward neural network combines both embeddings into one. However, experiments showed that the overlapping model showed no improvement over the independent.

Results are reported for the cased versions of BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$. Even though both language models are able to digest segments of up to 512 tokens, they perform best for segments of 128 or 384 tokens. To limit GPU memory usage, the training examples are pruned to 11 segments for BERT$_{\text{BASE}}$ and three segments for BERT$_{\text{LARGE}}$. The *bert-model* achieves an F1 score on the Onto Notes 5.0 test set of 73.9 with the base sized language model and 76.9 with the large version.

SpanBERT By replacing the BERT language models with SpanBERT$_{\text{BASE}}$ and SpanBERT$_{\text{LARGE}}$ the performance increases to 77.7 F1 for the base[1] and 79.6 F1 for the large version. SpanBERT is pre-trained on masked spans of text rather than single words and, therefore, is better suited for the task of coreference resolution than other BERT variants [39].

During the pre-training, continuous randomly sampled input tokens $(x_s, ..., x_e)$ are masked. The objective function consists of the MLM objective and a newly introduced span boundary objective (SBO), which predicts the span tokens x_i given the contextualized embeddings x^*_{s-1} and x^*_{e+1} of the tokens before and after the span as well as a positional embedding p_{i-s+1} of the token relative within the span:

$$\begin{aligned}\mathcal{L}(x_i) &= \mathcal{L}_{\text{MLM}}(x_i) + \mathcal{L}_{\text{SBO}}(x_i)\\ &= -\log P(x_i|x^*_i) - \log P(x_i|x^*_{s-1}, x^*_{e+1}, p_{i-s+1})\end{aligned}$$

The SBO encourages the transformer model to include as much information about the span as possible into the embeddings of the tokens before and after the span. Since the span representations g_i of the previously described coreference systems are partially defined by the first and last tokens of the spans, it is plausible to ascribe the performance gains on the coreference task to the SBO objective.

4.1.4. Recent Coreference Resolution Approaches

Xu and Choi achieved with their implementation without the higher-order inference slightly better results for BERT$_{\text{LARGE}}$ and SpanBERT$_{\text{LARGE}}$ [99]. Their best model with cluster merging, which makes the *bert-model* a truly entity-ranking model, reaches an F1 score on the Onto Notes 5.0 test set of 80.2 and 79.9 (±0.2) as a mean average over five runs.

In order to create a lightweight coreference system without handcrafted features and with less pruning, Kirstain et al. replace the span representations of the *bert-model* by solely relying on representations of the start and end token of each span [43]. Lightweight bilinear functions are applied to the start and end token representations to calculate the scores for the mention and antecedent candidates. They use the implementation of Xu and Choi [99] without the higher-order inference. The Longformer [4] is used as the pre-trained language model to avoid the segmentation described in [38]. Besides reducing the memory footprint of the *bert-model* the start-to-end model (*s2e*) also accomplishes a higher performance of 80.3 F1.

Another completely different approach for a lightweight model is the word-level coreference resolution system (*wl-coref*) by Dobrovolskii [21]. Instead of evaluating coreferences on a pruned subset of all possible mentions, the *wl-coref* evaluates coreferences between words and creates spans for coreferent words subsequently. Even though the author provides their own implementation, the model does not rely on higher-order inference as well. The *wl-coref* achieves an F1 score of 81 on the Onto Notes 5.0 test set.

CorefQA represents the current state-of-the-art system with its score of 83.1 F1 [97]. However, it is a very compute-intensive model, even in comparison to the *bert-model* with SpanBERT$_{\text{LARGE}}$, and uses additional data. The coreference resolution is formulated as a question answering problem using a single sentence with a highlighted mention as the question and expecting all coreferent mentions in the input text as the answer. The transformation into a question answering problem enables the model to be pre-trained on question answering corpora which are typically larger than coreference annotated corpora.

[1] The performance of SpanBERT$_{\text{BASE}}$ is not evaluated in [39]. In the SpanBERT repository a score of 77.4 F1 is reported (`https://github.com/facebookresearch/SpanBERT`) in the repository of [38] the score is stated to be 77.7 F1 (`https://github.com/mandarjoshi90/coref`).

4.2. PyTorch Implementation

All three models explained in detail above were originally implemented with TensorFlow [55]. With PyTorch [96] becoming the standard deep learning framework for research, the need for PyTorch implementations of the most important models regarding coreference resolution is inevitable.[2] Providing models implemented with different frameworks can also enable more people to step into the field of coreference resolution.

I implemented the *e2e-model*, *c2f-model*, and *bert-model* building up on each other and with a common structure.[3] In the following implementation details are given and the performance in comparison with the original implementations is reported.

4.2.1. Other Implementations

Multiple PyTorch implementations of the three models exist without corresponding scientific work or properly reported performances.[4] Xu and Choi [99] provide an implementation of the *bert-model* with different configurations of which they report the performances. One of them is equivalent to the original *bert-model* with SpanBERT$_{\text{LARGE}}$.

4.2.2. Implementation Details

Noticeable changes from the original implementation are a different ELMo embedding and the use of the HuggingFace transformer API [96]. Instead of the ELMo embedding from the TensorFlow Hub, I use another embedding from AllenNLP for my implementation of *c2f-model*.[5] However, further experiments showed no effect on the performance regarding the used embedding. Building the *bert-model* on top of the HuggingFace transformer API enables a quick exchange of the underlying pre-trained language model and makes it far easier to conduct experiments with several different language models.

In order to reduce the GPU memory footprint of all three models, the implementations include the option for mixed precision training [56]. Half-precision is used for operations that do not need the precision of a 32 bit floating point number, reducing the memory consumption and speeding up the computation without sacrificing performance. The models can also be used with gradient checkpointing [10] to further reduce the GPU memory requirements. However, in contrast to mixed precision training, gradient checkpointing is a trade-off between computation time and memory consumption. During the forward pass, activation values are only saved at specific checkpoints in the computational graph. Omitting most activation values saves memory, but since they are needed to calculate the gradient, the forward-pass has to be performed again on the model's segments between the checkpoints during backpropagation.

4.2.3. Comparison to the Original Implementation

Table 4.1 shows the performance of my reimplementation of the *e2e-model*, the *c2f-model* and the *bert-model* variants in comparison with the original reported CoNLL-2012 scores on the OntoNotes 5.0 test set. All hyper-parameters are identical to the original implementation. For the *e2e-model* and the *bert-model* with BERT$_{\text{LARGE}}$ and SpanBERT$_{\text{LARGE}}$ I was able to match the

[2] `http://horace.io/pytorch-vs-TensorFlow/`

[3] The code is available under: `https://github.com/jfhetzer/e2e-coref`

[4] `https://github.com/search?q=coref+pytorch&type=Repositories`

[5] `https://allenai.org/allennlp/software/elmo`

Model	TensorFlow	PyTorch
e2e-coref	67.2	67.2
c2f-coref	73.0	70.9
BERT$_{\text{BASE}}$	73.9	73.4 / 74.2†
BERT$_{\text{LARGE}}$	76.9	76.9
SpanBERT$_{\text{BASE}}$	77.7[1]	77.1 / 77.6†
SpanBERT$_{\text{LARGE}}$	79.6	79.7

Table 4.1.: Comparison of my PyTorch implementation to the original TensorFlow implementation on the OntoNotes 5.0 test set. † denotes results obtained by training for 30 epochs instead of 20. A more detailed breakdown of the PyTorch implementations scores can be found in Appendix A.1.

reported results. For the base versions of the *bert-model*, I could not quite reach the results but managed to do better by training for 30 epochs instead of 20. Despite many efforts, I was not able to match the *c2f-models* performance and clearly lack behind with my implementation.

4.3. Experiments and Analysis

Besides BERT and SpanBERT, there is a vast amount of other language models and BERT variations. These modifications of the original BERT embedding should either lead to performance gains or be computational more efficient. The embedding used in the *bert-model* model can be easily switched thanks to the HuggingFace transformer framework. In this section, I evaluate the *bert-model* using different BERT variations on the English portion of the OntoNotes 5.0 dataset.

4.3.1. Pre-trained Language Models

RoBERTa The Robustly optimized BERT approach (RoBERTa) was introduced by Liu et al. in 2019 [52]. RoBERTa brings multiple modifications to the pre-training of BERT in order to improve its performance in various downstream tasks. Dynamic masking ensures that a new randomly selected mask is applied to the input sequence in every training step. In contrast, BERT creates the masks during data preprocessing and tries to limit that downside by duplicating the training data multiple times beforehand. The Next Sentence Prediction (NSP) loss is dropped so that the MLM is the only objective during pre-training. Consequently, the input format changes from a pair of segments to contiguously complete sentences of one or more documents. RoBERTa also leverages larger batches and a more extensive BPE vocabulary.

Besides the improvements above, RoBERTa is trained on a vastly larger set of training data than BERT (16GB to 160GB) and trained for much longer (100K training steps to 500K). As a result, RoBERTa outperforms BERT clearly on the GLUE [92] and SQuAD [71] tasks.

DistilBERT and DistilRoBERTa DistilBERT by Sanh et al. is a version of BERT distilled into a smaller model to reduce its size by 40% compared to the original BERT transformer [80]. The authors also claim the pre-trained model to be 60% faster at inference while preserving 97% of its language understanding capabilities.

For distilling a larger teacher model m_t into a smaller student model m_s, the student model is trained to behave like the teacher model. In the case of DistilBERT a triple loss is used for training the student model consisting of the masked language modelling loss $\mathcal{L}_{MLM}$, a cosine embedding loss $\mathcal{L}_{cos}$ between the students' and teachers' hidden states, and the distillation loss $\mathcal{L}_{ce} = \sum_i t_i \cdot \log s_i$ with t_i being the probability given by m_t and s_i the probability given by m_s.

Besides DistilBERT, Hugging Face published a distilled version of RoBERTa as well, dubbed DistilRobERTa.[6] It is reported to outperform the cased DistilBERT on all GLUE tasks and the uncased version on all but one.

TinyBERT TinyBERT is another language model distilled from BERT$_{\text{BASE}}$ [36]. In contrast to DistilBERT, not only the number of layers is reduced but also their size. Besides the general distillation, the authors also propose a distillation and data augmentation step during fine-tuning on the downstream task. This second distillation step requires the larger teacher model to be previously fine-tuned on the downstream task as well.

The distillation loss is composed of the MSE losses between the embeddings, attention matrices and hidden states of the student and teacher model. To compare the embeddings and hidden states of different sizes, learned matrices scale up the students' embeddings and hidden states before calculating the loss. The prediction loss on the resulting logits vectors, which is similar to the distillation loss of DistilBERT, is used only for the distillation during the fine-tuning.

The model comes in two different sizes. The bigger TinyBERT$_6$ has the same size as DistilBERT with 6 transformer layers, an embedding size of 768, and a hidden size of 3,072. The smaller TinyBERT$_4$ makes use of its capability to scale down the teachers' layer sizes. It contains four layers and an embedding and hidden size of 312. With both steps of distillation and data augmentation, the authors claim to reach 96.7% of the performance of BERT$_{\text{BASE}}$ on the GLUE tasks while being 7.5 times smaller and 9.4 times faster. They also report to match the performance BERT$_{\text{BASE}}$ with TinyBERT$_6$.

ELECTRA Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) differs from all other language models covered in this thesis by proposing a new pre-training task called replaced token detection [13].

Instead of a single model, two transformer models, a generator and a discriminator, are trained in a non-adversarial way. The generator is a small language model trained on the MLM task that creates corrupted input tokens by predicting the previously masked tokens. The discriminator has to decide for each token if it was corrupted. Since it learns from the decision on every token and not just from the masked ones, the training is computationally more efficient than the MLM-based training.

After pre-training, the generator is discarded, and the discriminator is fine-tuned on the downstream tasks. Due to its efficient training, the authors claim to outperform BERT-based models on the GLUE tasks given the same model size, data, and compute.

4.3.2. Results and Analysis

Table 4.2 shows the results of the comparison on the English part of the OntoNotes 5.0 dataset. The results for BERT$_{\text{BASE}}$ and SpanBERT$_{\text{BASE}}$ are the scores of the original implementation reported by the authors and discussed in more detail in Section 4.1.3. While evaluation of RoBERTa$_{\text{BASE}}$ and ELECTRA$_{\text{BASE}}$ aims to find language models which can match or outperform

[6] `https://huggingface.co/distilroberta-base`

the ones originally used, the goal for the other language models examined in this experiment is to create a more lightweight coreference resolution system.

Most hyperparameters used in this experiment are the same as in [38]. The learning rates, segment size, and number of segments during the training can be found in the Appendix A.3 for each model. Since the goal of using TinyBERT is the reduction of GPU memory consumption and execution time during the training and not the pursuance of coreference systems for inference on low-resource devices, the used TinyBERT models are distilled from a general BERT model and the second distillation step during fine-tuning is omitted.

	MUC			B^3			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
BERT$_{\textbf{BASE}}$	80.2	82.4	81.3	69.6	73.8	71.6	69.0	68.6	68.8	73.9
SpanBERT$_{\textbf{BASE}}$	-	-	-	-	-	-	-	-	-	77.7[1]
RoBERTa$_{\textbf{BASE}}$	82.2	83.4	82.8	72.9	75.7	74.3	72.4	70.8	71.6	76.2
ELECTRA$_{\textbf{BASE}}$	83.0	85.0	84.0	73.6	78.1	75.8	73.6	72.5	73.0	77.6
ELECTRA$_{\textbf{SMALL}}$	82.2	76.6	79.3	72.2	65.7	68.8	69.3	61.0	64.9	71.0
DistilBERT (cased)	82.1	76.2	79.0	72.2	64.8	68.3	69.0	62.1	65.4	70.9
DistilBERT (uncased)	82.4	76.9	79.6	72.9	66.5	69.5	70.0	62.8	66.2	71.8
DistilRoBERTa	80.6	79.7	80.1	70.5	69.9	70.2	69.2	65.0	67.0	72.4
TinyBERT$_{\textbf{4}}$	79.8	70.8	75.1	69.5	58.4	63.4	65.3	53.2	58.7	65.7
TinyBERT$_{\textbf{6}}$	81.5	77.5	79.5	71.9	67.3	69.5	69.6	63.6	66.5	71.8

Table 4.2.: Comparison of the *bert-model* using different language models on the English part of the Onto Notes 5.0 dataset.

Performance Gains The *bert-model* with RoBERTa$_{\text{BASE}}$ clearly outperforms its counterpart with BERT$_{\text{BASE}}$, while ELECTRA$_{\text{BASE}}$ even matches the performance of the SpanBERT$_{\text{BASE}}$, a language model specialized for problems like coreference resolution and only available for the English language. On the other hand, are far more pre-trained versions of RoBERTa and ELECTRA publicly available on the HugginFace model hub.[7] Therefore, both language models might be great candidates to improve coreference resolution on languages other than English.

	MUC			B^3			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
BERT$_{\textbf{LARGE}}$ [38]	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
SpanBERT$_{\textbf{LARGE}}$ [39]	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
SpanBERT w/o HOI + CM [99]	85.9	85.5	85.7	79.0	78.9	79.0	76.7	75.2	75.9	80.2
s2e + **Longformer** [43]	86.5	85.1	85.8	80.3	77.9	79.1	76.8	75.4	76.1	80.3
wl-coref + **RoBERTa** [21]	84.9	87.9	86.3	77.4	82.6	79.9	76.1	77.1	76.6	81.0
CorefQA [97]	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	83.1
ELECTRA$_{\textbf{LARGE}}$	85.6	87.0	86.3	78.2	81.1	79.6	77.6	76.3	77.0	81.0

Table 4.3.: My implementation of the *bert-model* with ELECTRA Large in comparison with the current state-of-the-art coreference resolution systems briefly described in 4.1.4

For the best performing pre-trained model ELECTRA$_{\text{BASE}}$ I also trained the coreference system with its larger counterpart - the ELECTRA$_{\text{LARGE}}$ - for 40 epochs. Table 4.3 shows

[7] https://huggingface.co/models

its performance in comparison with the original *bert-model* as well as with more recent work, including the current state-of-the-art. While it can not match the performance of the computationally much more expansive and with additional question answering data trained CorefQA [97], it matches the word-level coreference approach [21] and outperforms any other model based on the *bert-model*. This result reassures the capabilities of ELECTRA and indicates that it might be an even better pre-trained model for coreference resolution than the Longformer and RoBERTa which were used in recent work.

LM	Steps / Sec	#Parameters
BERT$_{\textbf{BASE}}$	1.42	153m
SpanBERT$_{\textbf{BASE}}$	1.75	153m
RoBERTa$_{\textbf{BASE}}$	1.66	169m
ELECTRA$_{\textbf{BASE}}$	1.83	153m
ELECTRA$_{\textbf{SMALL}}$	2.88	25m
DistilBERT (cased)	2.06	110m
DistilBERT (uncased)	2.08	111m
DistilRoBERTa	1.74	127m
TinyBERT$_4$	3.19	29m
TinyBERT$_6$	1.88	111m

Table 4.4.: Efficiency of various pre-trained language models used in the *bert-model*.

Efficiency Enhancement Table 4.4 compares the distilled and smaller language models with their bigger counterparts in terms of training time measured in steps per second and the number of parameters of the complete coreference resolver. While the *bert-model* has about the same number of parameters for DistilBERT and TinyBERT$_6$ and only slightly more parameters for RoBERTa$_{\text{BASE}}$, the much smaller TinyBERT$_4$ and ELECTRA$_{\text{SMALL}}$ stand out with only a quarter of the parameters. The training speed partially depends on the segment size and the number of segments used during training. Since the segments are sequentially processed, longer and fewer segments, which also add up to slightly fewer tokens overall, tend to be faster. Despite being larger, the RoBERTa$_{\text{BASE}}$, for example, is faster than the BERT$_{\text{BASE}}$ for that reason. The hyperparameters for each model can be found in Appendix A.3.

Table 4.2 shows that the smaller models suffer in terms of performance compared to their larger counterparts. The DistilRoBERTa, which is the best performing smaller model, is also the slowest and biggest model among the smaller ones. It is also slower than the ELECTRA$_{\text{BASE}}$, which makes it an option only if the focus is on fewer parameters. A slight increase in speed is provided by the DistilBERT model. However, the only model not distilled - the ELECTRA$_{\text{SMALL}}$ - seems to be the best trade-off. While being the smallest model and the second fastest, it maintains an F1 score of 71, which is more than five F1 points above the slightly faster TinyBERT$_4$.

5. German and Cross-lingual Coreference Resolution

In this chapter, I cover German coreference resolution as well as cross-lingual coreference resolution. The latter is evaluated on German as well, yet additionally to the German coreference annotated data, it leverages knowledge from English coreference resolution systems described in Chapter 4 in order to improve its performance.

5.1. German Coreference Resolution

Besides English, which was the foundation for most of the developments in coreference resolution, German coreference resolution is also an active field of research. The SemEval-2010 Shared Task 1 [73] and the CORBON-2017 Shared Task [29] involve German coreference resolution. Furthermore, various German coreference annotated corpora exist - up to a similar size of the English corpora used in the CoNLL-2012 Shared Task.

5.1.1. German Corpora

While I conduct all experiments for German coreference resolution on the large Tüba-D/Z corpus, I use two smaller datasets as well to evaluate the capabilities of cross-lingual coreference resolution on lower resource languages. A detailed comparison regarding the size of the three German corpora and the English OntoNotes 5.0 dataset can be found in Appendix A.2.

TüBa-D/Z The "Tübinger Baumbank des Deutschen / Zeitungskorpus" (TüBa-D/Z) is a German syntax annotated corpus consisting of almost four thousand newspaper articles in its latest version [84]. Multiple versions were released over the years. In this thesis, the 10th version of the TüBa-D/Z is used for training and evaluating German coreference resolution systems. The newer 11th version is slightly larger, but there are no coreference scores reported on that version yet.

SemEval-2010 The SemEval-2010 shared task on coreference resolution defined a standard for training and evaluating coreference systems on many languages [73]. The German portion of the provided coreference annotated dataset is the 8th version of the TüBa-D/Z corpus. To make it easier to distinguish between the corpora used in this thesis, this dataset is referred to as SemEval-2010 in the following. With just above 1,200 documents and is the second largest German corpus used in this thesis.

DIRNDL The Discourse Information Radio News Database for Linguistic analysis (DIRNDL) [7] is a corpus of German news broadcasts with coreference and prosodic annotations. In the following DIRNDL refers to the version used by Rösinger et al. [77]. That version consists of just under 500 documents, which are much shorter than the documents of the two corpora described above. That makes DIRNDL the smallest German dataset in this thesis by far.

5.1.2. German Coreference Resolution Systems

Most German coreference resolvers are rule-based or feature-based systems leveraging simple machine learning frameworks to learn coreference resolution in a data-driven manner. Two of the most common systems are the *IMS HotCoref DE* and the *CorZu* system.

Even though the research in coreference resolution progressed unceasingly for the English language, developments of German systems seemed to stall in the last years until the recent adaptation of deep learning models to the German language [81].

IMS HotCoref DE The *IMS HotCoref DE* [76] is an adaptation of the IMS Higher-Order Tree Coreference system (*IMS HOTCoref*) that was proposed as a resolver for Arabic, Chinese, and English [6]. The *IMS HOTCoref* is a feature-based system that models coreferences at the document level as a directed rooted tree. Every mention refers to a node in that tree with a dummy node at the root. Arcs reaching from the antecedent nodes to the anaphor nodes indicate the coreferences. All subtrees under the root node correspond with entity clusters.

A structured perceptron [14] is trained by updating the predicted tree against a latent tree that is also inferred by the perceptron given constrained antecedent candidates. The candidates are restricted so that only trees can be found, which subtrees indicate the gold coreference clusters. The latent tree should help the model to build correct trees in terms of its induced coreference clusters by pushing it to structure the root's subtrees in a simple way to learn. Making a left to right pass and incrementally building up the tree gives the possibility to include not only local features like mention type, distance, syntax, and lexical features but also non-local features like the shape or size of the clusters defined by the partially built tree.

Rösiger and Kuhn adapted the *IMS HOTCoref* to German [76]. Therefore, they tackled some problems specific to the German language, including grammatical gender, richer inflections, and compounds that are single words in German, by modifying and extending data handling and the model itself. They also tried to incorporate world knowledge by leveraging GermaNet [32]. Besides the features inherit

Besides the features inherited by the *IMS HOTCoref* system and features specific to the German language, the *IMS HotCoref DE* was also used to augment coreference resolution with prosodic information on the DIRNDL dataset [77]. For this purpose, a CNN acts as a prosodic event detector. The detected events can then be used as additional features to benefit the coreference resolution.

CorZu The Coreference Resolver for German from Zurich (*CorZu*) [85] is an incremental entity-mention system that does not validate the relation between single mentions and entity clusters like other entity-mention models, but between two mentions, one of which represents its entity cluster. Each mention is sequentially evaluated to be coreferent with all previous, not yet matched, antecedent candidates as well as with the last mention of each already formed entity cluster. To prevent morphological disagreements with other mentions of the cluster, morphological properties of former mentions are projected onto the last mention.

CorZu is a hybrid between a rule-based and a feature-based system. For selecting antecedent candidates, deterministic rules are applied. To select the best antecedent among those candidates, a Markov Logic Network [74] is used to learn weights assigned to constraints regarding the relation between the two mention pairs like distance, syntax, or properties of the possible antecedent.

German c2f Recent, concurrent work by Schröder et al. tackled the application of the *bert-model* on German data [81]. Referring to the coarse-to-fine pruning approach of the *c2f-model* they dub their model *German c2f*. By replacing the English BERT model with German or multilingual language models, they follow the same approach used in this chapter, and their work partially overlaps with the experiments I conducted and described in the following. Their best-performing base and large model are both based on pre-trained German ELECTRA models.

While I use my own implementation exactly replicating the original *bert-model* for all experiments, the *German c2f* is based on the implementation of Xu and Choi [99]. Consequently, the authors omit the higher-order inference as originally proposed and reduce the size of the feed-forward layers in order to further reduce the memory consumption as well.

5.2. Cross-lingual Coreference Resolution

Many supervised NLP systems require large, manually annotated corpora. These corpora are only available for a few languages in which research is mainly done. However, real-world applications derived from this research should often times serve more languages, but creating corpora for all of them is just not feasible. Cross-lingual language understanding can circumvent this problem by building systems trained on a single language and applying it on many others [16]. Transfer learning, which describes the transfer of knowledge learned in one domain to another, is essential for this task. In the context of two different languages, cross-lingual transfer learning aims to transfer a system trained on a source language to a target language on which the final evaluation takes place.

Today's deep learning coreference resolution systems rely on large coreference annotated corpora as well. These are available just for a couple of languages, and research is mainly done for the English language. Since coreference resolution can be an important factor for various downstream tasks, including machine translation, as shown in Chapter 6, its application on a wide variety of languages is essential. This emphasizes the importance of the experiments described in the following, evaluating cross-lingual coreference resolution between English and German for multiple settings with different sized target language corpora.

5.2.1. Multilingual Embeddings

In recent years with new, advanced word embeddings changing NLP in many aspects, multilingual word embeddings have become a common way to build multilingual and cross-lingual systems. Even though many of them are not trained with the objective of mapping different languages to a common feature space and only focus on representing each language on its own as good as possible, they work surprisingly well for cross-lingual transfer learning and even for zero-shot learning.

Multilingual BERT Although BERT was originally proposed as a single language model [20], a multilingual model (M-BERT) trained on monolingual corpora of 104 languages was released alongside other variants.[1] The M-BERT model is of the size of BERT$_{\mathrm{BASE}}$ and case-sensitive,

[1] https://github.com/google-research/bert

replacing an earlier uncased version trained on 102 languages. Even though M-BERT is pre-trained without any objective assuring generalization across multiple languages, it was found to perform surprisingly well on zero-shot cross-lingual model transfers [66].

XLM Since M-BERT was initially not designed with the idea in mind of serving as a multilingual embedding, the cross-lingual language model (XLM) was specifically adapted for multilingual use cases [45]. The original vocabularies are replaced with a single shared vocabulary created through Byte-Pair Encoding (BPE). Besides the MLM objective used by BERT, XLM introduces two additional objectives. The Casual Language Modelling (CLM) task challenges the model to predict the probabilities for the next word given the previous ones. The Translation Language Modelling (TLM) uses parallel data and allows for training the embedding in a supervised fashion. Instead of a single sentence, a pair of the same sentence in both languages is used as input. Words in both sentences are selected and masked randomly. In order to predict the masked words, the model can leverage not just the context in the same language but also the translation. This should enforce the model to align the embeddings in both languages.

Embeddings are pre-trained with the MLM and CLM objectives for multiple language pairs, including German to English. Additional one embedding is pre-trained with MLM and TLM together.

XLM-Roberta In contrast to XLM, the XLM-RoBERTa (XLM-R) embedding relies solely on the MLM objective [15]. It is argued that using only data obtained from Wikipedia results in poor performance, especially on low-resource languages. Instead, the XLM-R is trained on a substantially larger dataset of 2.5TB of CommonCrawl data. They observe a trade-off between the cross-lingual performance for low-resource languages and the overall monolingual and cross-lingual performance when adding more languages to the embeddings but can overcome that trade-off by increasing the model size. The final model covers 100 languages. Similar to what RoBERTa did to improve BERT, Conneau et al. can further boost the performance of their embedding by tuning training parameters. They state that XLM-R does not just outperform M-BERT and XLM but can also compete with monolingual models on the XNLI and GLUE benchmarks.

5.2.2. Cross-lingual Learning Settings

Given labeled source and target language data at training time, various training strategies can be applied. Three basic settings are used in the following experiments to assess the benefits of cross-lingual coreference resolution. Additionally, a more advanced, adversarial method aiming to improve the language models' mapping between the source and target language is evaluated.

Training on the Target Language A baseline is trained directly on the target language data, not using the source language data at all. This baseline is used for comparison with systems leveraging the source language data and, therefore, to evaluate the benefit cross-lingual training can bring to coreference resolution. The system's performance depends on the quantity and quality of the target language corpora.

Zero-Shot Learning In the zero-shot learning setting, no target language data is available during training. The system is trained on the source language and evaluated on the target language without seeing a single example in that language before. In the context of coreference

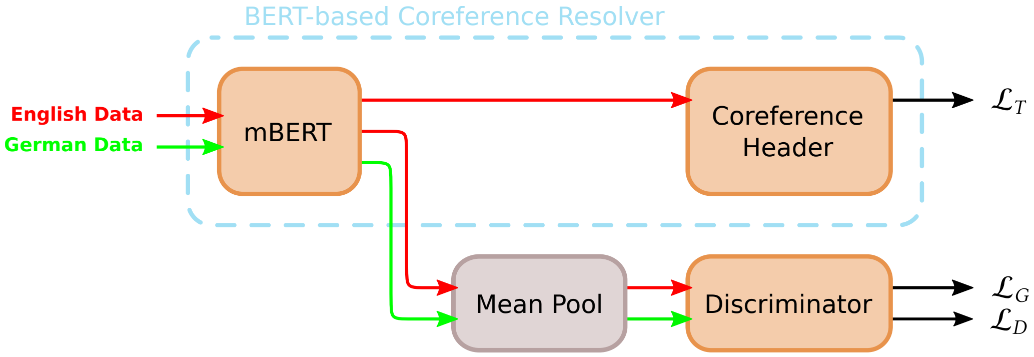


Figure 5.1.: BERT-based coreference resolver and the discriminator layer for adversarial learning. Possible flows of English and German data through the network are indicated in red and green.

resolution, this setting is especially relevant for languages without any coreference annotated corpora.

Fine-Tuning on Target Language Equally to the zero-shot learning setting, the model is first trained on the source language. Subsequently, however, it is further fine-tuned on the target language data. The hyperparameters differ between training on the source and target language. In particular, the number of training steps is less during fine-tuning. Fine-tuning the coreference resolution system is vital for many languages with an only very limited amount of coreference annotated data.

Adversarial Cross-lingual Learning Language adversarial techniques aim to map both languages, the source and target language, to a common feature space in order to improve the performance on the target language despite training mostly or even explicitly on the source language. The adversarial cross-lingual learning approach in this thesis is based on the proposal for text classification and named entity recognition (NER) by Keung et al. [42].

Besides training the system on a downstream task for a high resource source language, they introduced another adversarial task. For that task, the language model acts as the generator creating the output embeddings. An additional layer, projecting the mean pooled output embedding to a single score, acts as the discriminator trying to distinguish between documents of the source and target language. Figure 5.1 shows the different modules of such a setup applied to the BERT-based coreference resolver and the use of English and German training data in the network. While the coreference task is only trained on English data, the generator and discriminator losses rely on documents of both languages. The generator loss $\mathcal{L}_G$ and the discriminator loss $\mathcal{L}_D$ are defined as follows:

$$\mathcal{L}_G(y^A; x) = -(1 - y^A) \log p(E = 1|x) - y^A \log p(E = 0|x)$$
$$\mathcal{L}_D(y^A; x) = -(1 - y^A) \log p(E = 0|x) - y^A \log p(E = 1|x)$$

The binary label y^A is 1 for English documents and 0 for German documents, while $p(E = 1|x)$ is the probability that the input document x is in English, estimated by the *bert-model* with the discriminator head. The coreference loss, as well as the generator and discriminator losses, are calculated alternatingly, and the corresponding parts of the model are updated immediately. The batch size to compute all three losses is one document. For a more detailed explanation of the algorithm, see [42].

5.3. Experiments and Analysis

For all experiments, I use my implementation of the *bert-model* by Joshi et al. [38]. To adapt that coreference system to German, the underlying pre-trained language model is replaced by German or multilingual language models. The same CoNLL-2012 metric and scorer used in Section 4.3.2 is used for the evaluation on German as well.

As a contribution to the German coreference resolution, I analyze the performance of the *bert-model* with various German versions of BERT and ELECTRA on the large TüBa-D/Z v10 dataset. To explore how cross-lingual coreference resolution can benefit languages with different amounts of coreference annotated data, I evaluate the settings described in Section 5.2.2 on the TüBa-D/Z v10, the SemEval-2010, and the DIRNDL dataset.

5.3.1. German Coreference Resolution with Pre-trained Language Models

Various German adaptions of BERT and ELECTRA are available via the Hugging Face model hub[2]. If available, I choose the cased version in the base size of each model. For the most promising model, I evaluated the larger version as well. The following gives a detailed overview of the German pre-trained language models used in this experiment.

deepset BERT The deepset BERT has the same architecture and size as the original BERT$_{\text{BASE}}$ model. It is case-sensitive and trained on 12GB of German text data.[3]

DBMDZ BERT The later released DBMDZ BERT was trained on 16GB of data and is said to slightly outperform the deepset BERT on downstream tasks. A cased and an uncased base-sized version were released.[4]

GBERT The GBERT model is a joint work of the authors of the deepset BERT and DBMDZ BERT [9]. In contrast to those two models, it uses whole word masking rather than masking single tokens. Besides a cased base-sized model, a larger model of the size of BERT$_{\text{LARGE}}$ was released as well.

GELECTRA The GELECTRA was proposed alongside the GBERT in [9]. A base and large version of the same size as the GBERT variants were made publicly available. While the authors report the GELECTRA$_{\text{LARGE}}$ to outperform its BERT-based counterpart on all tested downstream tasks, the ELECTRA$_{\text{BASE}}$ lags behind the GBERT$_{\text{BASE}}$.

GNG-ELECTRA Another ELECTRA model pre-trained on German was published by the German NLP group.[5] It is dubbed GNG-ELECTRA in the following and is only available in a base-sized, uncased version.

All language models were used with a segment size of 512 as I find that longer segments result in better performances, congruent with [81]. To match the number of steps trained for in Section 4.3.2, the number of epochs is increased to 26 except for the ELECTRA-based models,

[2] https://huggingface.co/models

[3] https://deepset.ai/german-bert

[4] https://github.com/dbmdz/berts

[5] https://huggingface.co/german-nlp-group/electra-base-german-uncased

which further improve by training for even 40 epochs. The remaining hyperparameters are identical to those used for BERT in [38].

Table 5.1 shows the coreference resolution performance on the TüBa-D/Z v10 corpus using the different German pre-trained language models and compares them to the *IMS HotCoref DE* and the recently proposed *German c2f* systems.

	MUC			B^3			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
IMS HotCoref DE [76]	-	-	52.57	-	-	45.13	-	-	64.79	48.54
German c2f Base [81]	81.92	79.90	80.90	77.41	73.52	75.41	75.16	75.50	75.33	77.21
German c2f Large [81]	82.85	81.61	82.23	78.41	75.73	77.05	76.75	77.44	77.09	78.79
deepset BERT	71.23	76.13	73.60	63.94	68.97	66.36	65.08	70.85	67.84	69.27
DBMDZ BERT	76.46	73.35	74.87	69.66	66.76	68.18	71.12	67.34	69.18	70.74
GBERT$_{\text{BASE}}$	77.47	80.28	78.85	70.09	74.74	72.34	73.82	72.78	73.30	74.83
GNG-ELECTRA$_{\text{BASE}}$ (40 Epochs)	78.99	83.88	81.36	72.27	79.53	75.73	75.78	76.85	76.31	77.80
GELECTRA$_{\text{BASE}}$ (40 Epochs)	77.04	83.77	80.26	69.61	79.32	74.15	73.52	75.46	74.48	76.30
GELECTRA$_{\text{LARGE}}$ (40 Epochs)	80.77	85.82	83.22	75.01	81.78	78.25	77.71	79.84	78.76	80.08

Table 5.1.: Performance of German Coreference Systems on the TüBa-D/Z v10 test data.

As expected, all BERT-based coreference systems were able to clearly outperform the feature-based *IMS HotCoref DE* by up to over 30 points. The *German c2f Base* and *German c2f Large* which use the exact same pre-trained GNG-ELECTRA and GELECTRA$_{\text{LARGE}}$, fall short of the performance of my own implementation with both language models. This might be due to the different implementation of the *bert-model* used or the omission of the higher-order coreference. Another reason could be the adjustment of hyperparameters in order to save GPU memory usage and computation time as well as the number of epochs which is not reported in [81]. It is noticeable that the performance differences narrow down on the development set. Whether the parameter tuning on the development set helped to limit the differences on the test set or hindered generalization is questionable considering these results.

The DMBDZ BERT slightly improves upon the deepset BERT. This result is consistent with previously reported results on other downstream tasks like NER and POS tagging.[6] The GBERT outperforms both by wide margin as suggested in [9]. Contrary to the results on offensive language detection and NER [9], the GELECTRA$_{\text{BASE}}$ achieves better results than the GBERT$_{\text{BASE}}$. The uncased GNG-ELECTRA performs even better as reported in [81].

5.3.2. Cross-lingual Coreference Resolution with Pre-trained Language Models

To emulate the problem of training a coreference resolution system on a low-resource language or a language without any coreference annotated corpus, I use the three German corpora from Section 5.1.1 to train or fine-tune the system on. The extensive TüBa D/Z v10 dataset is mainly used as a reference, whereas the much smaller DIRNDL and the midsize SemEval-2010 corpus are used to evaluate the benefits cross-lingual learning can bring to coreference resolution and for which amount of target language data it is useful.

All cross-lingual experiments are conducted with the base versions of the pre-trained language models and a segment size of 128 to the reduced GPU memory consumption and because just a moderate drop in performance for shorter segments.

Table 5.2 shows the performance of the multilingual language models trained and evaluated on the English portion of the OntoNotes 5.0 corpus. All hyperparameters are identical to

[6] https://github.com/stefan-it/fine-tuned-berts-seq

	MUC			*B*^{3}			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Multilingual BERT	82.6	79.1	80.8	73.4	69.2	71.3	71.2	65.2	68.1	73.4
XLM-RoBERTa	79.3	79.2	79.3	67.0	69.4	68.1	69.1	62.0	65.3	70.9

Table 5.2.: Evaluation of multilingual pre-trained language models on the OntoNotes 5.0 dataset.

those used for BERT$_{\text{BASE}}$ in [38]. These trained models are applied to German datasets for zero-shot evaluation and fine-tuned with additional German annotated data in the following experiments.

Zero-Shot Learning and Fine-tuning on German On all of the three German corpora, the *bert-model* with M-BERT and XLM-R is evaluated after training directly on the training set of the same corpus, after training on the English OntoNotes 5.0 dataset, and after training on English and fine-tuning on German data. For comparison, a setting with the best performing German base model GNG-ELECTRA is directly trained on each dataset as well. Since that language model suffers significantly from shorter segments, I report results for segment sizes of 128 and 512. For training directly on the German datasets, the number of epochs is chosen to match the number of steps in 4.3.2. However, it must be taken into account that the update of the model for each step is based on a different number of coreference examples. While the TüBa-D/Z v10 contains almost as many tokens as the OntoNotes 5.0 dataset, even though it has significantly fewer documents, the SemEval-2010 and especially the DIRNDL dataset consist of much shorter documents. For a detailed comparison, see Appendix A.2.

	MUC			*B*^{3}			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
IMS HotCoref DE	-	-	-	-	-	-	-	-	-	47.93
+ prosodic information	-	-	-	-	-	-	-	-	-	48.88
GNG-ELECTRA 128	66.60	47.85	55.69	66.55	44.97	53.67	67.35	45.50	54.31	54.56
GNG-ELECTRA 512	68.90	51.28	58.80	68.76	48.80	57.08	73.22	47.34	57.50	57.79
Multilingual BERT	66.51	42.85	52.12	66.18	40.41	50.18	67.70	38.77	49.30	50.53
zero-shot (from English)	68.19	51.14	58.44	67.05	48.47	56.27	67.05	51.00	57.93	57.55
+ fine-tuning (1 Epoch)	74.26	61.00	66.98	73.46	57.73	64.66	76.27	56.82	65.12	65.59
+ fine-tuning (5 Epochs)	73.35	65.28	69.08	71.66	62.38	66.70	72.97	62.20	67.15	67.64
+ fine-tuning (10 Epochs)	76.85	63.57	69.58	76.24	60.43	67.42	75.67	60.26	67.09	68.03
+ fine-tuning (20 Epochs)	77.75	66.42	71.64	76.53	63.13	69.19	74.96	62.55	68.20	69.68
XLM-RoBERTa	56.87	52.00	54.32	54.26	50.19	52.15	59.70	45.27	51.49	52.65
zero-shot (from English)	67.48	55.14	60.69	65.92	52.46	58.43	70.00	52.14	59.77	59.63
+ fine-tuning (1 Epoch)	71.87	62.42	66.81	70.84	60.33	65.16	74.13	57.71	64.90	65.62
+ fine-tuning (5 Epochs)	73.39	65.42	69.18	71.13	62.20	66.37	73.03	59.96	65.85	67.13
+ fine-tuning (10 Epochs)	72.63	63.71	67.88	70.77	60.74	65.38	73.65	57.67	64.69	65.98
+ fine-tuning (20 Epochs)	71.47	63.00	66.97	70.83	60.92	65.50	72.77	59.91	65.72	66.06

Table 5.3.: Evaluation of the *bert-model* for various cross-lingual settings on the DIRNDL corpus.

Table 5.3 shows the evaluation on the tiny DIRNDL dataset. The *bert-model* is able to outperform the *IMS HotCoref DE* as expected for every language model. Although the GNG-ELECTRA performs better than the multilingual language models when directly trained on DIRNDL, the latter profit enormously from pre-training on English data. Even the zero-shot setting with a model, which has never seen a German sentence during training, matches the GNG-ELECTRA with a segment size of 512 in the case of M-BERT or performs even better in the case of XLM-R. Both multilingual language models further improve after fine-tuning the zero-shot model on the German dataset. Most of the gains are made during the first five epochs of fine-tuning, and the XLM-R cannot improve further. The M-BERT, however, reaches a score of 69.68 F1 by fine-tuning for 20 epochs and outperforms the GNG-ELECTRA by over 12 percentage points. The gap to the directly on DIRNDL trained M-BERT is even larger with almost 20 percentage points. These results prove that cross-lingual coreference resolution can massively benefit low-resource languages.

	MUC			B^3			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
IMS HotCoref DE	-	-	52.11	-	-	45.55	-	-	48.17	48.61
CorZu	-	-	-	-	-	-	-	-	-	45.82
GNG-ELECTRA 128	73.11	71.60	72.34	66.48	64.92	65.69	67.90	67.62	67.76	68.60
GNG-ELECTRA 512	74.65	76.30	75.46	68.27	70.41	69.32	69.66	72.79	71.19	71.99
Multilingual BERT	73.23	67.80	70.41	66.76	61.05	63.78	67.92	64.73	66.29	66.83
zero-shot (from English)	61.91	51.09	55.98	53.51	41.53	46.76	54.05	41.96	47.24	49.99
+ fine-tuning (1 Epoch)	71.18	64.74	67.81	64.08	56.51	60.06	63.62	58.94	61.19	63.02
+ fine-tuning (5 Epochs)	71.62	71.84	71.73	64.95	64.74	64.85	66.14	67.20	66.67	67.75
+ fine-tuning (10 Epochs)	72.38	72.93	72.66	64.62	66.40	65.50	67.23	66.76	67.00	68.39
XLM-RoBERTa	66.78	66.32	66.55	58.61	60.07	59.33	63.61	61.07	62.31	62.73
zero-shot (from English)	57.67	50.63	53.92	47.76	41.30	44.30	51.41	38.32	43.91	47.38
+ fine-tuning (1 Epoch)	68.56	63.07	65.7	62.52	55.46	58.78	63.09	60.44	61.73	62.07
+ fine-tuning (5 Epochs)	69.20	69.87	69.53	61.46	63.21	62.33	64.50	64.46	64.48	65.45
+ fine-tuning (10 Epochs)	69.64	70.44	70.04	61.97	63.74	62.85	64.51	65.46	64.99	65.96

Table 5.4.: Evaluation of the *bert-model* for various cross-lingual settings on the SemEval-2010 corpus with excluded singletons.

On the larger SemEval-2010 dataset, the directly trained BERT-based systems significantly improve upon their performance on the DIRNDL dataset, as shown in Table 5.4. The zero-shot models are clearly lagging behind but can still match or slightly exceed the performance of the *IMS HotCoref DE* and *CorZu*. Fine-tuning the zero-shot model leads to better results than directly training on the German dataset with the same language model. However, unlike on the DIRNDL dataset, the fine-tuned multilingual language models cannot outperform the GNG-ELECTRA directly trained on the SemEval-2010 dataset. Nevertheless, the results of the cross-lingual settings are still encouraging for languages with similar-sized coreference annotated corpora and without language-specific language models comparable to the GNG-ELECTRA.

The trend observed from the small DIRNDL corpus to the bigger SemEval-2010 continues on the comparatively large TüBa-D/Z v10 dataset. Table 5.5 shows that the performance gap between the deep learning models directly trained on the German dataset and the zero-shot setting widens. However, the zero-shot models are able to outperform the *IMS HotCoref DE*, whose performance is reported in Table 5.1. The *IMS HotCoref DE* performs similarly on all three datasets despite their enormous difference in size, suggesting that the system is not able to utilize the additional data. On the other hand, the score of the zero-shot models increased from the SemEval-2010 to the TüBa-D/Z v10 dataset, although the very same checkpoints of those models were used for evaluation on all German corpora. That might result from the similarity between the TüBa-D/Z v10 and the OntoNotes 5.0 dataset. Even though the SemEval-2010 dataset is a predecessor of the TüBa-D/Z v10 dataset, the document sizes and sentence lengths of the latter are closer to the English dataset. Also, the number of mentions per entity and entities per document are similar according to [81]. The zero-shot models further fine-tuned on German can improve upon their counterparts directly trained on the TüBa-D/Z v10, although the improvement for the M-BERT is marginal. The fine-tuned M-BERT can match the performance of the GNG-ELECTRA for the shorter segments, but both multilingual language models cannot keep up with the longer segments. The experiments on the TüBa-D/Z v10 dataset show that for languages with large coreference annotated corpora, cross-lingual coreference can lead only to a minimal improvement. At the same time, however, they also indicate that for languages without any coreference data, a zero-shot transfer from English promises a performance that matches shallow feature-based models or is even slightly higher.

	MUC			B^3			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
GNG-ELECTRA 128	76.75	80.28	78.48	69.62	74.79	72.11	73.19	72.45	72.82	74.47
GNG-ELECTRA 512	78.99	83.88	81.36	72.27	79.53	75.73	75.78	76.85	76.31	77.80
Multilingual BERT	79.76	75.86	77.76	74.03	69.57	71.73	74.63	71.09	72.82	74.32
zero-shot (from English)	63.97	56.64	60.08	55.44	47.51	51.17	53.90	46.06	49.67	53.64
+ fine-tuning (1 Epoch)	79.19	74.67	76.87	74.22	67.46	70.68	72.00	71.93	71.97	73.17
+ fine-tuning (5 Epochs)	78.15	78.65	78.40	72.44	72.58	72.51	72.86	74.31	73.58	74.83
+ fine-tuning (10 Epochs)	78.02	78.27	78.15	72.32	72.22	72.27	72.79	73.65	73.22	74.55
XLM-RoBERTa	76.33	70.90	73.51	69.50	63.82	66.54	71.33	64.79	67.90	69.32
zero-shot (from English)	58.49	56.79	57.63	47.18	47.97	47.57	51.30	41.13	45.65	50.28
+ fine-tuning (1 Epoch)	75.48	72.20	73.80	67.82	65.26	66.51	69.29	65.55	67.37	69.23
+ fine-tuning (5 Epochs)	75.18	76.27	75.72	67.51	70.05	68.76	70.73	68.80	69.75	71.41
+ fine-tuning (10 Epochs)	73.45	78.07	75.69	65.54	72.40	68.80	70.17	69.11	69.64	71.38

Table 5.5.: Evaluation of the *bert-model* for various cross-lingual settings on the TüBa-D/Z v10.

Adversarial Cross-lingual Learning To increase the zero-shot capabilities of the *bert-model* even further, I apply the adversarial cross-lingual learning approach described in Section 5.2.2. For this experiment, I use the M-BERT since it outperforms the XLM-R throughout the previous experiments. The model is trained on the English portion of the OntoNotes 5.0 corpus and evaluated on the TüBa-D/Z v10 test set for the coreference task. To train the language model acting as the generator and the discriminator layer on the adversarial task, the model is provided with English and German data from the training sets of the OntoNotes 5.0 and TüBa-D/Z v10 datasets. The hyperparameters of the model and the coreference task are identical to the hyperparameters used for the M-BERT in Table 5.2. The learning rates for the adversarial task are set to 3e-7 for the generator and 2e-4 for the discriminator.

	MUC			B^3			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Performance on OntoNotes 5.0										
Multilingual BERT	82.56	79.14	80.81	73.42	69.2	71.25	71.23	65.19	68.08	73.38
+ Adversarial CL Learning	82.84	78.51	80.61	73.81	68.28	70.94	71.09	64.95	67.88	73.14
Zero-Shot Performance on TüBa-D/Z v10										
Multilingual BERT	63.97	56.64	60.08	55.44	47.51	51.17	53.90	46.06	49.67	53.64
+ Adversarial CL Learning	65.90	56.79	61.01	57.61	47.48	52.06	56.25	46.54	50.93	54.67

Table 5.6.: Evaluation of the adversarial cross-lingual learning approach on the English OntoNotes 5.0 and the German TüBa-D/Z v10 dataset via zero-shot transfer.

Table 5.6 shows the performance of the *bert-model* trained with and without the adversarial task on the English and German test sets. Despite being optimized against an additional objective, the model can almost retain its performance on the English test set. For the zero-shot application on German, the performance increases by 1 F1 point, suggesting that the adversarial cross-lingual learning does, indeed, benefit the cross-lingual coreference resolution capabilities.

Appendix A.5 shows that the adversarial task brought the English and German embeddings closer together as intended. However, it also indicates that the language model is not improved by learning a mapping between English and German but instead squeezes the embeddings into a smaller feature space.

6. Context-Aware Neural Machine Translation

One common field of application for coreference information is machine translation [60]. However, with recent developments in context-aware Neural Machine Translation (NMT), the benefit of incorporating coreference information into those systems is in question. Earlier work showed that the pronoun translation of context-agnostic models can be improved by finding the best combination of translation hypotheses regarding the induced coreference clusters [94]. Hwang et al. slightly improve two of the context-aware models originally evaluated on pronoun resolution in [59], but do not report results for training the best performing models with their contrastive learning approach [35]. Contrary, it is shown in [3] and [89] that context-aware models inherit coreference resolving capabilities and even learn some form of anaphora resolution.

6.1. Context-Aware Systems learn Discourse Phenomena

After demonstrating the coreference resolving capabilities earlier [89], Voita et al. proposed another context-aware model in 2019, achieving improvements over a context-agnostic baseline on deixis, ellipsis, and lexical cohesion [90]. They argue that context-aware models are generally developed with a metric not sensitive to these discourse phenomena, of which at least deixis and lexical cohesion are strongly tied to coreference resolution.

Their model is closely related to the deliberation networks [98]. As shown in Figure 6.1, it consists of two parts: A context-agnostic encoder-decoder model called base model and a context-aware decoder called CADec. However, for the sake of simplicity, the entire model consisting of the context-agnostic base model and the context-aware decoder is referred to as *CADec* in this thesis. Moreover, the context-agnostic base model is called *base-model* when used as a standalone model in the following experiments.

The base model is a reimplementation of the original transformer [87] and translates every source and context sentence independently. It is trained on single sentences and fixed before training the compound model, including the context-aware decoder. The objective is to maximize the sentence-level log-likelihood, where x_i denotes the source and y_i the target sentences.

$$\sum_{(x_i,y_i)\in D_{sent}} \log P(y_i|x_i,\Theta_B)$$

The context-aware decoder is a modification of the transformers decoder. It must take into account not only the encoder's output like in the standard encoder-decoder setting but also the output of the context-agnostic decoder, resulting in an additional multi-head attention layer. The translation of the source sentence by the base model is used as input to the very first self-attention layer of the context-aware decoder. The states from the last layer of the base model's encoder for the source and all source-side context sentences are fed together with a sentence distance embedding into the next attention layer. The output of the base model's decoder is concatenated with the embedding of the sampled, target-side tokens and

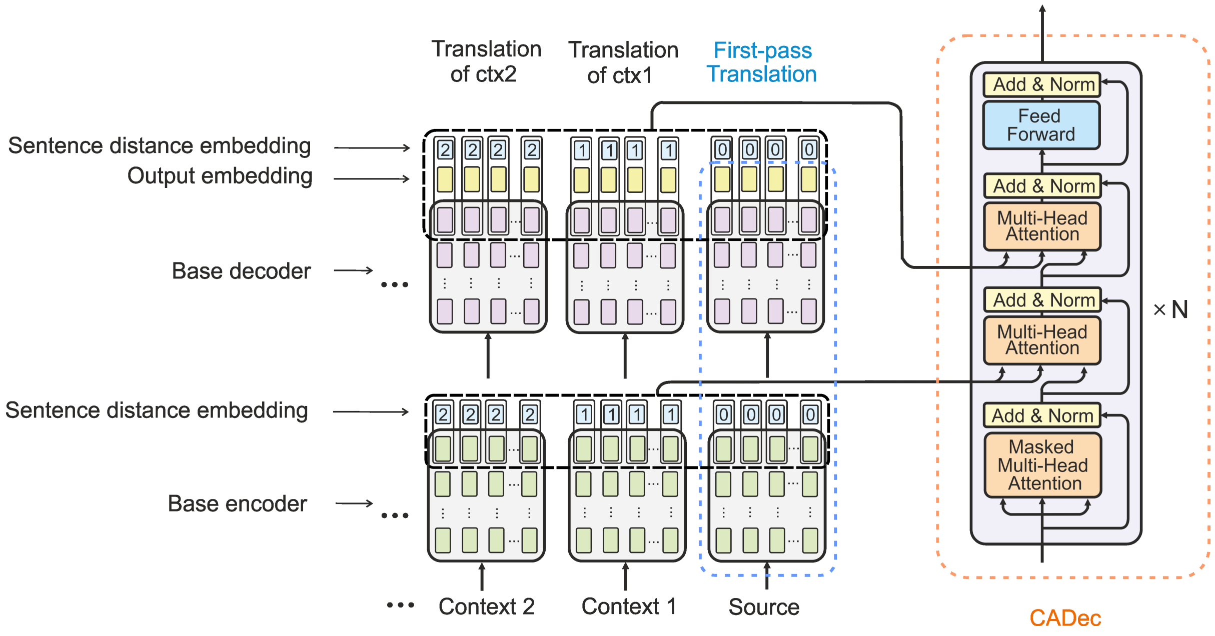


Figure 6.1.: Context-agnostic encoder-decoder baseline translates each sentence independently before the CADec-decoder corrects the source sentence translation [90].

an additional sentence distance embedding. The complete concatenation is fed into the last attention layer of the context-aware decoder. Dependent on the translation y_j^B sampled from the base, the context-aware decoder is trained to maximize the following log-likelihood and therefore correct the mistakes made by the base:

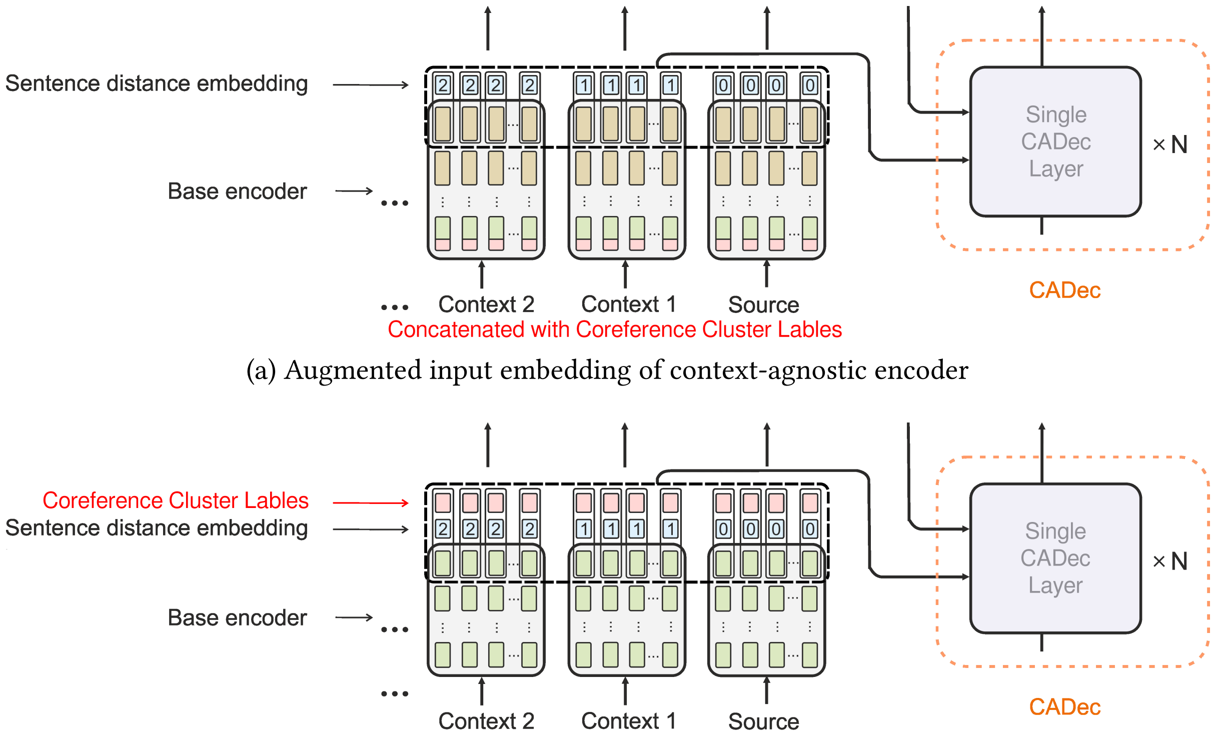
$$\sum_{(x_j,y_j)\in D_{doc}} \log E_{y_j^B \propto P(y|x_i,\Theta_B)} P(y_j|x_j, y_j^B, c_j, \Theta_C)$$

During training, either the translation of the source sentence by the base model or a corrupted version of the reference translation is used as input to the context-aware decoder. Moreover, reference translations are used for the context sentences.

6.2. Augmenting NMT Systems with Coreference Information

To prove that coreference information can still benefit today's NMT systems, I augment the context-aware *CADec* with coreference cluster information predicted by a dedicated coreference resolver. In the following, two different methods are proposed to provide the model with this information.

However, not only the *CADec* is a promising candidate for coreference augmentation - the context-agnostic *base-model* on which the *CADec* is built, can be trained with additional coreference information as well. Training the initial *base-model* is almost seven times faster per step than training the *CADec* itself. On the OpenSubtitles2018 dataset [51], however, the *base-model* is on par with the *CADec* regarding the BLEU score. Translating from English to Russian, both models reach a score of 32.4, according to Voita et al. [90]. Providing the *base-model* with explicit coreference information might lead to an improved pronoun handling like the *CADec* is capable of but without the significant increase in training time.



(a) Augmented input embedding of context-agnostic encoder

(b) Augmented encoder output before fed into the CADec-decoder

Figure 6.2.: Two different approaches to augmenting the context-agnostic encoder and CADec-decoder with coreference cluster labels.

6.2.1. Coreference Cluster Labels

Each source token t_j belonging to a coreferent word can be associated with one or more coreference clusters C_i. Coreference cluster labels c_i^j are one-hot encoded vectors in which each dimension i relates to a coreference cluster.

$$c_i^j = \begin{cases} 1 & \text{if } word(t_j) \in C_i \\ 0 & \text{otherwise} \end{cases}$$

The *base-model* and *CADec* can be augmented with coreference cluster labels at various places where an assignment with the source tokens is possible.

Baseline with Coreference Clusters Labels One way of augmenting the *base-model* and indirectly the *CADec* with coreference cluster labels is to add them to the input embedding of the encoder. This should lead to a better translation by the context-agnostic model regarding coreference-related phenomena. Since the translation of the *base-model* is used as the candidate translation, which is only refined by the context-aware part of the *CADec*, a better initial translation should benefit the *CADec* as well.

The input embedding of each token of the source sentence and each token of the source-context sentences, when used in the *CADec*, is augmented with a coreference cluster label as shown in Figure 6.2a. Due to the residual connections of the transformer network, the hidden state size is increased by the same amount. This further enlarges the key and value matrices in the *CADec*-decoder and also results in larger key and value matrices in the context-agnostic decoder. Overall the number of parameters is only slightly increased in comparison with the original *CADec*.

CTX1: My hand is quite steady and the story soon comes to an end.

CTX2: My friend suffered from her heart since her youth.

CTX3: All too often it would beat too passionately and sometimes she felt as if a merciless hand clutched at the twitching thing and squeezed it, dreadful and aching,

SRC: so that it may come to rest!

so that **[START]** her heart **[SPLIT]** the twitching thing **[SEPERATOR]** it **[END]** may come to rest!

(a) Simple example of text-level coreference annotation

CTX1: Alright, who is it?

CTX2: It’s *your* little brother Charlie.

CTX3: Well, for goodness’ sake!

SRC: It’s *my* brother Charlie.

It’s **[START]** your little brother charlie **[SEPERATOR]** **[START]** your **[SEPERATOR]** my **[END]** brother Charlie **[END]** .

(b) Recursive example of the text-level coreference annotation

Figure 6.3.: Two examples from the ContraPro test set [59] of different complexity with text-level coreference annotation provided by the *bert-model* using SpanBERT$_{\text{LARGE}}$

CADec with Coreference Cluster Labels Another option to specifically augment the *CADec*-decoder with coreference information is to add the coreference cluster labels to the context-agnostics encoder’s output along with the sentence distance embedding, as shown in Figure 6.2b. The combined embedding is fed into every layer of the *CADec*-decoder as key and value into a multi-head attention layer. Due to the additional information, the corresponding key and value weight matrices are slightly larger than in the original *CADec* leading to a marginal increase in the overall number of parameters.

6.2.2. Text-level Coreference Annotation

Instead of altering the *base-model* and *CADec*, coreference information can also be provided by incorporating it into the input sentences. Figure 6.3a shows a simple example of how the pronoun in the source sentence is augmented with coreferent mentions from the context sentences. The *[START]* and *[END]* tokens surround each mention and its coreferent mentions, while the *[SEPERATOR]* token separates the original mention from the provided coreferent mentions. In the case of multiple coreferent mentions, the *[SPLIT]* token separates the coreferent mentions from each other. Since mentions can contain other mentions and the *bert-model* is able to resolve those recursions, it has also to be modeled by the annotation logic. Figure 6.3b gives an example of the annotation of recursive mentions. The above examples show how a single sentence is augmented. This is sufficient for the context-agnostic *base-model*. However, for the

CADec the context sentences have to be augmented as well - not just with coreferent mentions from preceding context sentences, but also from succeeding sentences, including the source sentence to cover cataphoric relations.

6.3. Experiments

6.3.1. Datasets

Performance boosts in NMT due to context information in general and coreference information, particularly, are challenging to measure since the commonly used BLEU metric is unlikely to show any significant improvements [59]. Therefore various test suites emerged, focusing only on specific words or single phenomenons [31][41][90].

ContraPro The models used in this experiment are evaluated on the ContraPro test set to assess the impact of explicit coreference information [59]. ContraPro is a comparatively large contrastive set of English to German translations focusing on pronouns. It contains 12,000 samples automatically extracted from the OpenSubtitles corpus [50] by searching for the English pronoun *it*, aligned with the German pronouns *er*, *sie* and *es*. The test set consists of 4,000 samples for each German pronoun.

Each sample consists of the source sentence, source context sentences, the target sentence, and target context sentences. Additionally, contrastive target sentences are given, in which the German pronoun is replaced by a wrong pronoun. The NMT systems translate the English source sentence and calculate the score of that translation given the correct one or one of the contrastive German target sentences like it is done with the target sentence during training. If the best score is assigned to the target sentence with the correct pronoun, the sample is considered to be solved correctly.

WMT17 Like the models in the evaluation done by the authors of ContraPro, the augmented models are trained on the English to German part of the training corpus for the WMT 2017 shared task on news translation [8]. This includes the Europarl v7, the Common Crawl corpus, the News Commentary v12, and the Rapid corpus of EU press releases. Together they consist of ~5.85 million sentence pairs - enough to create ~1.46 million sentence pairs with three context sentences. All models based on the *base-model* are trained on all of the single sentence pairs, including the sentences used for context training the *CADec*-based models. These numbers also come close to the number of the English to Russian dataset used by the authors of the *CADec* [90].

For evaluating the BLEU score of the trained models I use a development set consisting of the NEWSTEST2014, NEWSTEST2015 and NEWSTEST2016 data provided by the WMT 2017 shared task. They consist of 8,171 sentence pairs resulting in 2,042 sentence pairs with context.

The already preprocessed data downloaded from the official website[1] is further prepared by the pipeline described for the *CADec*.[2] Subsequently the single sentence pairs are split into groups of four sentences to train the context-aware models. Since the training data does not contain any document boundaries, there is a small number of sentences with random context sentences. As discussed in [59], the models should be robust against these anomalies.

[1] http://statmt.org/wmt17/translation-task.html

[2] https://github.com/lena-voita/good-translation-wrong-in-context/blob/master/README.md

Even though the development data is provided with document boundaries, the contextualized samples are created in the same manner as the training data.

6.3.2. Coreference Resolution

The coreference clusters were created by my PyTorch implementation of the model by Joshi et al. [38] utilizing SpanBERT$_{\text{LARGE}}$ as the pre-trained language model [39]. It was trained on the Onto Notes 5.0 dataset as described in 4.3.2 and applied to the English portion of the WMT17 corpus. The data was split into the same documents of four sentences as it was done during the NMT preprocessing. Each document was tokenized by the SpanBERT tokenizer and split into segments of 512 tokens, including the classification and separator token at the start and end of each segment. The resulting coreference clusters were aligned with the data preprocessed by the NMT pipeline, translated into coreference cluster labels, and used to annotate the source-side text with the text-level coreference annotations explained in Section 6.2.

The number of dimensions of the coreference cluster labels is given by the maximal number of clusters in a single document of the corpus. For the WMT17 data, the coreference system resolved up to 18 coreference clusters for a single document. Since the vast majority of documents contain far less clusters, the one-hot encoded coreference cluster labels of each document are randomly permuted before training. Thereby all indices are equally likely to imply a coreferent relation. For the context-agnostic models, the labels are cleaned so that mentions in the source sentence are only assigned to a cluster if it contains another mention in the same sentence.

For the text-level coreference annotations, the coreferent mentions are sorted by their occurrence in the document from early to late. Each coreferent mention in the context and source sentence is then annotated with the other sorted mentions. Unlike the coreference cluster labels, the annotations are not cleaned for the context-agnostic models. The source sentences of the single sentence pairs do contain mentions from context sentences even though they are used with a context-agnostic model.

6.3.3. Models

Besides the *base-model* and *CADec* as described in [90], I train both of the models with multiple modifications. If not stated otherwise, all configurations are trained with the hyperparameters originally used.

Input Cluster Labels The input embedding of the context-agnostic encoder is augmented with coreference cluster labels as described in Section 6.2.1 and shown in Figure 6.2a.

Output Cluster Labels The output of the context-agnostic encoder is augmented with coreference cluster labels before being fed into the context-aware decoder as described in Section 6.2.1 and shown in Figure 6.2b.

Text-level Annotation Mentions in context and source sentences are annotated with other coreferent mentions as described in Section 6.2.2 and shown in Figure 6.3.

Larger Base Model #1 The number of layers in the context-agnostic transformer is increased from 8 to 12.

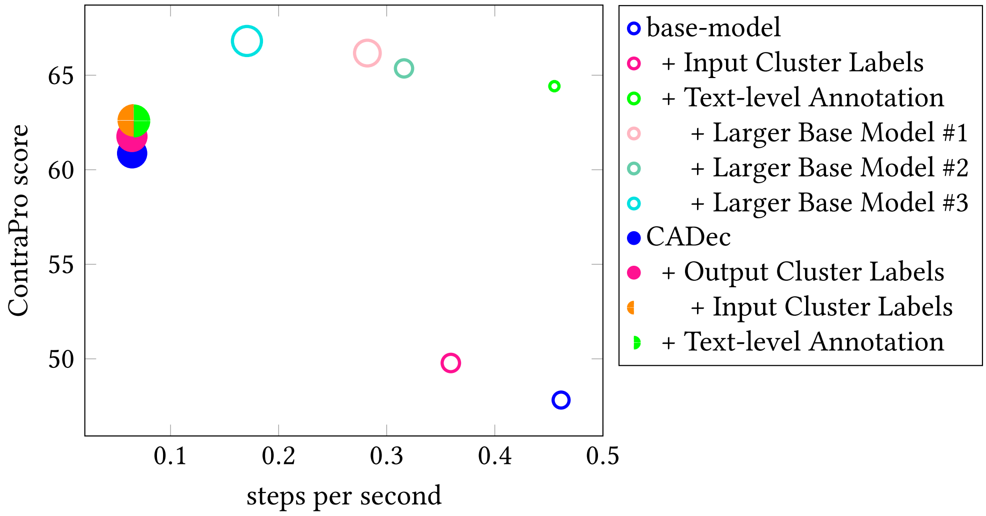


Figure 6.4.: Comparison of the different NMT approaches regarding their performance on the ContraPro test set and the speed measured in steps per second during the training. Additionally, the BLEU score on the WMT17 development set is visualized by the size of each mark. The exact numbers this plot is based on can be found in Appendix A.6.

Larger Base Model #2 The embedding and hidden state sizes of the context-agnostic transformer are increased from 512 to 768.

Larger Base Model #3 The embedding and hidden state sizes of the context-agnostic transformer are increased from 512 to 768. Additionally, the number of attention heads is increased from 6 to 12, and the dropout of the attention layers, residual layers, and fully-connected layers from 0.1 to 0.2.

6.4. Results and Analysis

The trained NMT systems are evaluated regarding three different metrics. The ContraPro score, the BLEU score, and the speed measured during training in steps per second. The *base-model* configurations were trained for about 150k steps except for the bigger models, which were trained for an additional 50k steps. The *CADec* configurations were trained for about 130k steps. The snapshots for evaluation were chosen by the best ContraPro scores.

With an accuracy of 60.87% on the ContraPro test set, the *CADec* is on par with the best performing model reported in [59] and clearly outperforms the previously proposed model by Voita et al. [89]. That makes the *CADec* an excellent candidate to measure the gains from explicit coreference information provided by a dedicated system.

Figure 6.4 draws the big picture, visualizing the performance of all configurations on all three metrics. It clearly shows the dominance of the *CADec*-based models over the *base-model* and how the *base-model* with more parameters and text-level coreference annotations can overtake the by design context-aware models in terms of pronoun resolution and speed while matching them regarding the BLEU score.

6.4.1. Analysis of Coreference Cluster Labels

Table 6.1 shows that the models augmented with coreference cluster labels achieve a slight but constant improvement over their baselines for the pronoun translation on the ContraPro test set. The *CADec* with additional coreference cluster labels in the context-agnostic encoder's input further improves upon the configuration with only output cluster labels.

	CONTRAPRO SCORE
base-model	47.82
+ Input Cluster Labels	49.78
+ Text-level Annotation	64.42
+ Larger Base Model #1	66.17
+ Larger Base Model #2	65.36
+ Larger Base Model #3	66.81
CADec	60.87
+ Output Cluster Labels	61.76
+ Input Cluster Labels	62.61
+ Text-level Annotation	62.58

Table 6.1.: Overall scores on the ContraPro dataset evaluated on all 12,000 contrastive samples.

The *base-model* with input cluster labels outperforms its baseline as well but is still far away from the performance of the *CADec* configurations. This could be expected since the input cluster labels only provide coreference information regarding the source sentence but no information about the context sentences. Unlike the *CADec* variants, it is also significantly slower than its baseline, as shown in 6.4.

6.4.2. Analysis of Inline Coreference Augmentation

	#Parameters	BLEU Score
base-model	128m	31.00
+ Input Cluster Labels	129m	31.09
+ Text-level Annotation	128m	30.58
+ Larger Base Model #1	172m	31.61
+ Larger Base Model #2	206m	31.08
+ Larger Base Model #3	287m	31.82
CADec	191m	31.70
+ Output Cluster Labels	191m	31.76
+ Input Cluster Labels	192m	31.84
+ Text-level Annotation	191m	31.87

Table 6.2.: Comparison of the NMT systems in terms of model size and BLEU score

The *base-model* with text-level coreference annotations and its bigger variants clearly outperform every other model regarding the overall accuracy on the ContraPro test set. In terms of the BLEU score, however, the standard-sized *base-model* falls short of the context-aware models. The result is contrary to the findings of the authors of the *CADec* [90]. They claim that the context-agnostic and context-aware models do not differ in BLEU score. This discrepancy might occur due to the different language pairs used. While this experiment was conducted on English to German, the original *CADec* was trained on an English to Russian corpus. Moreover, the genre of the data differs as well. The WMT17 data mainly contains news texts and transcripts from the European parliament, the originally used OpenSubtitles dataset, in contrast, consists entirely of movie subtitles.

Table 6.2 shows that the model size could be a limiting factor hindering the *base-model* configurations from matching the BLEU score of the *CADec*. The larger *base-model* #3 proves that increasing the number of parameters can help to close that gap, though it is even 50% larger than the *CADec* models. The larger *base-model* #1 comes close to those BLEU scores and shows in comparison with the larger *base-model* #2 that increasing the number of layers in the transformers encoder and decoder has a more significant impact than increasing the layer and embedding sizes. As shown in Figure 6.4, all context-agnostic models are faster than the context-aware models. That makes the larger *base-model* #3 with text-level coreference annotation a better performing, faster to train model without any sacrifices regarding the BLEU score.

Even though the configurations with text-level coreference annotations can clearly outperform the configurations with coreference cluster labels for the *base-model*, the *CADec* with text-level coreference annotations cannot improve the overall performance of its counterpart with coreference cluster labels on the ContraPro test set. In the case of the *CADec*, both approaches encode the same coreference information in different ways, whereas for the *base-model*, the text-level coreference annotations contain additional contextual information in comparison to the coreference cluster labels.

6.4.3. Analysis of Pronoun Resolution

The ContraPro test set does not only provide an overall score for pronoun translation but allows for analyzing the systems regarding different German pronouns and distances between pronoun and antecedent as well.

	PRONOUN ACCURACY		
	er	es	sie
base-model	21.45	87.63	34.38
+ Input Cluster Labels	23.90	88.93	36.53
+ Text-level Annotation	50.00	88.18	55.08
+ Larger Base Model #1	49.68	88.68	60.15
+ Larger Base Model #2	50.43	90.18	55.48
+ Larger Base Model #3	51.60	90.93	57.90
CADec	42.00	90.58	50.03
+ Output Cluster Labels	46.33	91.20	47.75
+ Input Cluster Labels	48.45	91.55	47.83
+ Text-level Annotation	45.63	92.53	49.60

Table 6.3.: Accuracy of the NMT systems on the ContraPro test set broken down by German pronouns. For each pronoun ContraPro contains 4,000 contrastive samples.

Pronouns The pronoun distribution in the WMT17 training data is clearly reflected by the models' performances on the different German pronouns, as shown in Table 6.2. About one third of the English pronouns *it* refer to the German pronoun *es* while only ~8% refer to *sie* and ~6% to *er* [59].

Since the *base-model* seems to translate *it* mostly to *es* leaving much room for improvement on the pronouns *er* and *sie*, its variants with explicit coreference information improve especially on these two pronouns. The *CADec* configurations on the other hand can not outperform their baseline for the pronoun *sie* but manage to increase accuracy on *er* and slightly on *es*.

Distances Table 6.4 shows the performance of all configurations regarding the sentence distance between the pronoun and its antecedent. The *base-model* does perform strongly for distance 0 (pronoun and antecedent are in the same sentence). Due to its lack of context information, the accuracy drops for inter-sentence coreference. However, it seems to suffer less the greater the sentence distance becomes. This phenomenon is explained in [59] by the greater share of the German pronoun *es* for longer distances.

	DISTANCE ACCURACY				
	0	1	2	3	>3
base-model	75.42	36.86	48.81	50.09	66.97
+ Input Cluster Labels	77.96	38.95	47.75	53.93	71.72
+ Text-level Annotation	78.29	59.84	62.58	64.57	68.33
+ Larger Base Model #1	80.96	61.27	65.17	65.45	68.55
+ Larger Base Model #2	79.83	60.38	64.83	63.18	71.04
+ Larger Base Model #3	83.17	61.12	65.76	68.24	70.81
CADec	68.17	55.90	63.64	67.36	82.81
+ Output Cluster Labels	71.50	56.57	62.91	67.71	80.32
+ Input Cluster Labels	71.50	58.06	62.52	67.54	81.00
+ Text-level Annotation	74.04	56.66	64.64	70.16	78.28

Table 6.4.: Accuracy of the NMT systems on the ContraPro test set broken down by sentence distance between the pronoun and its antecedent. ContraPro contains 2,400 contrastive samples of intra-sentence coreferences. With 7,075 samples most data exist for the distance of a single sentence. With longer distances the number of samples declines from 1,510 to 573 to 442.

The *base-model* with coreference cluster labels improves upon its baseline for intra-sentence coreferences. Although the labels do not provide any information about inter-sentence coreferences, the model does improve for most greater sentence distances as well. One reason could be that the labels prevent the model from mistakenly recognizing coreferences in the source sentence. Another possibility is that some coreferences in the source sentence were missed or not considered to be a nominal antecedent during the automatic creation of the ContraPro test set. Appendix A.7 provides an evaluation of those coreference relations.

As described in Section 6.4.2 the *base-model* with text-level coreference annotations and its bigger variants clearly outperform the *base-model* variants, which were not augmented with context information. This observation is particularly evident for shorter inter-sentence coreferences, while for distances over three sentences, the other models can catch up due to the skewed pronoun distribution previously mentioned. The bigger variants can also improve their intra-sentence pronoun translation.

All *CADec*-based models perform worse than the *base-model* configurations for inter-sentence coreferences. Whereas the latter are provided with a single sentence, which most probably contains all necessary information needed to correctly translate the pronouns, the former need to identify that information in an input of four sentences making the problem more difficult. For distances of one and two sentences, the *CADec* falls short of the bigger *base-models* with text-level coreference annotations but can outperform them for longer dependencies. Several factors could play a role here: For longer distances, the annotations might suffer from longer coreference chains resulting in more useless mentions in the augmented source sentence between the pronoun and the nominal antecedent. This problem could be overcome by reversing the annotated mentions' order or removing the non-nominal mentions from them. Alternatively, the coreference clusters automatically defined by the coreference resolver might worsen for wider coreference distances. This would be consistent with the observation that coreference resolution systems struggle with larger documents [38]. The fact that the *CADec* with output cluster labels and the configurations with additional input cluster labels improve upon their baseline only for short distances as well strengthens this thesis. Contrary to the results of the other models augmented with coreference information, the *CADec* with annotations outperforms its baseline for a distance of three sentences but falls short for longer distances of not only the *CADec* but also the configurations with coreference cluster labels. This might result from a generally greater impact on the translation of the text-level coreference annotation compared to the cluster labels and a systematic problem of the coreference resolver for distances greater than four sentences. For those pronouns, the nominal antecedent is not even part of the context sentences the coreference resolver considers. Since the resolver trains almost entirely on complete documents, it might expect a nominal antecedent for every pronoun to exist within the boundaries of the input text and therefore tends to assign the pronoun to an antecedent candidate, even though the correct antecedent is not part of the text excerpt.

7. Conclusion and Future Work

7.1. Conclusion

In the scope of this thesis, I published a PyTorch implementation of three of the most important models in the recent years, whose implementations were originally done with TensorFlow: The *e2e-model* and *c2f-model* by Lee et al. [49][48] and the *bert-model* by Joshi et al. [38]. I used the *bert-model* to analyze how different pre-trained language models other than BERT or SpanBERT can increase the performance of coreference systems or improve their time and memory efficiency. For German coreference resolution, I could confirm the latest results of German neural end-to-end coreference systems [81] and reported a new state-of-the-art performance. Further experiments with multilingual language models demonstrated that low-resource languages or languages without any coreference annotated corpora can immensely profit from cross-lingual coreference resolution. Finally, I could also show how today's context-aware NMT systems can still benefit from explicit coreference information, proving that coreference resolution is still an imported field of research.

By replacing the originally used language models with ELECTRA$_{\text{LARGE}}$, the *bert-model* reached a CoNLL-2012 score of 81.0 F1 on the OntoNotes 5.0 only beaten by the massive *CorefQA* [97] which utilizes additional question answering corpora. I reported the performance, size, and training speed for various smaller and distilled language models making an informed decision on the trade-off between them possible. With ELECTRA$_{\text{SMALL}}$, the number of trainable parameters of the *bert-model* is six times lower than with the originally used BERT$_{\text{BASE}}$ while still maintaining a score of 71 F1.

For the German coreference resolution, ELECTRA seems to be the best language model as well. By using an ELECTRA$_{\text{LARGE}}$ model pre-trained on German, I achieved an F1 score of 80.08 on the TüBa-D/Z v10, which is, to the best of my knowledge, the highest score ever reported on that dataset. The findings of the evaluation of multiple German BERT and ELECTRA variants were mainly congruent with the contemporary work of Schröder et al. confirming their results [81].

I evaluated cross-lingual coreference systems on three German datasets of different sizes to emulate low-resource languages or languages without any coreference annotated data. I successfully transferred knowledge learned on a large English coreference annotated corpus to German by using multilingual language models together with the *bert-model*. The evaluation on the small DIRNDL dataset showed that cross-lingual coreference resolution can enormously benefit low-resource languages by outperforming the best model only trained on DIRNDL by about 12 F1 points. The same experiment on two larger corpora demonstrated that the cross-lingual approach can also be valuable to languages with more coreference annotated data if no better pre-trained language model specifically for that very language exists. A zero-shot transfer from English to German could match or outperform the scores of older feature- and rule-based systems on all three datasets and thus represents a valid option for languages without any coreference annotated corpora. The application of adversarial cross-lingual learning to a coreference resolution system to improve its cross-lingual capabilities slightly increased the zero-shot performance.

I proved that today's context-aware NMT systems still benefit from providing explicit coreference information, even though they do have some coreference resolution capabilities inherently. By adding coreference cluster labels to each token or by annotating coreferent mentions in the input text, the context-aware *CADec* [90] did improve on the translation of pronouns. However, an even more significant improvement and even better translation of pronouns were achieved by the actually context-agnostic baseline when annotating the input sentence with coreferent mentions from the context sentences. The faster baseline with the same BLEU score and better pronoun translation shows that explicit coreference information can not only improve context-aware models but can also help to replace them with less complex and more efficient systems. I analyzed the pronoun translation in detail, showing the strengths and weaknesses of different NMT systems, which can be used as a starting point for future work.

7.2. Future Work

Combining the ELECTRA$_{\text{LARGE}}$, which performed best with the *bert-model* in my English coreference experiments, with the changes to the *bert-model* proposed in [99] and [43] is a promising option for future work and might further increase the *bert-models* performance. This could lead to a new state-of-the-art system for coreference resolution without relying on massive question answering corpora and an highly computational expensive training.

To obtain even more meaningful and general insights into the benefits of cross-lingual coreference resolution, further experiments on more languages not as closely related as English and German, are necessary. But not just experiments with a single source language are of interest. Pre-training a single BERT-based coreference system on many languages might take even greater advantage of the capabilities of the multilingual language model and might lead to an improved cross-lingual performance of the coreference system.

In this thesis, I used the M-BERT and XLM-R language models as the foundation for the cross-lingual experiments. However, many more multilingual word embeddings exist, which may perform even better. Given the good performance of the different ELECTRA models used for English and German coreference experiments, an ELECTRA-based, multilingual language model like the XLM-E [11] is an exciting option for further investigations. Another possibility to further increase the cross-lingual performance is tuning additional hyperparameters like the segment size or emphasizing regularization by increasing the dropout rates.

Future work on coreference information in NMT could further elaborate the inline coreference annotation. With longer context and more coreferent mentions, the annotated source sentence gets harder to translate due to longer annotations. One approach to circumvent these problems is to include only the mentions head words in the annotations or utilize the headword embeddings learned by the *bert-model* for the mention representation. Head words contain a lot of information like gender or the grammatical number necessary for coreference resolution. Omitting the remaining parts of the mention would lead to shorter annotations and make it easier for the model to focus on important information. Another problem of more coreferent mentions is the higher percentage of repeating pronouns, especially later in the document, which do not offer additional information. Trying to remove those pronouns from the annotation or reversing the order of the mentions in the annotation so that the annotated mention and more meaningful coreferent mentions from the beginning of the document are closer together could be interesting as well.

Exploring other approaches to augment an NMT system with coreference resolution could also be the subject of future work. Leveraging the mention embeddings learned by the coreference system rather than annotating the input on the text-level could provide more context information. The coreference cluster labels provide a linking between embeddings learned by the NMT system but are limited to the context that system uses for translation.

Further optimization of the models I used in my experiments could be worthwhile as well. The performance of the largest variant of the *base-model* trained with text-level coreference annotations might be possible with fewer parameters and shorter training time by not increasing the hidden layer sizes. Also, the BLEU score of the *CaDEC* can possibly be improved by using the larger base models for the context-agnostic agnostic part of the model.

Bibliography

[1] Amit Bagga and Breck Baldwin. “Entity-based cross-document coreferencing using the vector space model”. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1.* 1998, pp. 79–85.

[2] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *3rd International Conference on Learning Representations, ICLR 2015.* 2015.

[3] Rachel Bawden et al. “Evaluating discourse phenomena in neural machine translation”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* New Orleans, Louisiana, 2018, pp. 1304–1313.

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. “Longformer: The long-document transformer”. In: *arXiv preprint arXiv:2004.05150* (2020).

[5] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. “A neural probabilistic language model”. In: *Advances in Neural Information Processing Systems* 13 (2000).

[6] Anders Björkelund and Jonas Kuhn. “Learning structured perceptrons for coreference resolution with latent antecedents and non-local features”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 2014, pp. 47–57.

[7] Anders Björkelund et al. “The extended DIRNDL corpus as a resource for automatic coreference and bridging resolution”. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC.* 2014, pp. 3222–3228.

[8] Ondřej Bojar et al. “Findings of the 2017 Conference on Machine Translation (WMT17)”. In: *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers.* 2017, pp. 169–214.

[9] Branden Chan, Stefan Schweter, and Timo Möller. “German’s next language model”. In: *Proceedings of the 28th International Conference on Computational Linguistics.* 2020, pp. 6788–6796.

[10] Tianqi Chen et al. “Training deep nets with sublinear memory cost”. In: *arXiv preprint arXiv:1604.06174* (2016).

[11] Zewen Chi et al. “XLM-E: cross-lingual language model pre-training via ELECTRA”. In: *arXiv preprint arXiv:2106.16138* (2021).

[12] Kyunghyun Cho et al. “On the properties of neural machine translation: Encoder-decoder approaches”. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation.* 2014, pp. 103–111.

[13] Kevin Clark et al. “Electra: Pre-training text encoders as discriminators rather than generators”. In: *International Conference on Learning Representations.* 2020.

[14] Michael Collins. “Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms”. In: *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002).* 2002, pp. 1–8.

[15] Alexis Conneau et al. “Unsupervised cross-lingual representation learning at scale”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* 2020, pp. 8440–8451.

[16] Alexis Conneau et al. “XNLI: Evaluating Cross-lingual Sentence Representations”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* 2018, pp. 2475–2485.

[17] André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. “Exploring spanish corpora for portuguese coreference resolution”. In: *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS).* IEEE. 2018, pp. 290–295.

[18] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[19] Pascal Denis and Jason Baldridge. “Global joint models for coreference resolution and named entity classification”. In: *Procesamiento del lenguaje natural* 42 (2009).

[20] Jacob Devlin et al. “BERT: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* 2019, pp. 4171–4186.

[21] Vladimir Dobrovolskii. “Word-Level Coreference Resolution”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.* 2021, pp. 7670–7675.

[22] George R Doddington et al. “The automatic content extraction (ACE) program–tasks, data, and evaluation”. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04).* 2004.

[23] Jeffrey L Elman. “Finding structure in time”. In: *Cognitive science* 14.2 (1990), pp. 179–211.

[24] André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. “Coreference resolution: toward end-to-end and cross-lingual systems”. In: *Information* 11.2 (2020), p. 74.

[25] Yaroslav Ganin et al. “Domain-adversarial training of neural networks”. In: *The journal of machine learning research* 17.1 (2016), pp. 2096–2030.

[26] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep sparse rectifier neural networks”. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics.* JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[28] Édouard Grave et al. “Learning word vectors for 157 languages”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* 2018.

[29] Yulia Grishina. “CORBON 2017 Shared Task: Projection-Based Coreference Resolution”. In: *Second Workshop on Coreference Resolution beyond OntoNotes.* 2017, p. 51.

[30] Ralph Grishman and Beth M Sundheim. “Message understanding conference-6: A brief history”. In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics.* 1996.

[31] Liane Guillou and Christian Hardmeier. “Protest: A test suite for evaluating pronouns in machine translation”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 636–643.

[32] Birgit Hamp and Helmut Feldweg. “Germanet-a lexical-semantic net for german”. In: *Automatic information extraction and building of lexical semantic resources for NLP applications*. 1997.

[33] Jerry R Hobbs. “Resolving pronoun references”. In: *Lingua* 44.4 (1978), pp. 311–338.

[34] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[35] Yongkeun Hwang, Hyeongu Yun, and Kyomin Jung. “Contrastive learning for context-aware neural machine translation using coreference information”. In: *Proceedings of the Sixth Conference on Machine Translation*. 2021, pp. 1135–1144.

[36] Xiaoqi Jiao et al. “TinyBERT: Distilling BERT for natural language understanding”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 4163–4174.

[37] Michael I Jordan. “Attractor dynamics and parallelism in a connectionist sequential machine”. In: *Artificial neural networks: concept learning*. 1990, pp. 112–127.

[38] Mandar Joshi et al. “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)*. 2019, pp. 5803–5808.

[39] Mandar Joshi et al. “Spanbert: Improving pre-training by representing and predicting spans”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 64–77.

[40] Dan Jurafsky and James H Martin. *Speech and language processing*. 3rd ed. draft. `https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf`, Accessed 2022-05-12.

[41] Prathyusha Jwalapuram et al. “EEvaluating pronominal anaphora in machine translation: An evaluation measure and a test suite”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 2964–2975.

[42] Phillip Keung, Yichao Lu, and Vikas Bhardwaj. “Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and NER”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 1355–1360.

[43] Yuval Kirstain, Ori Ram, and Omer Levy. “Coreference resolution without span representations”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2021, pp. 14–19.

[44] Gourab Kundu et al. “Neural cross-lingual coreference resolution and its application to entity linking”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2018, pp. 395–400.

[45] Guillaume Lample and Alexis Conneau. “Cross-lingual language model pretraining”. In: *arXiv preprint arXiv:1901.07291* (2019).

[46] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[47] Heeyoung Lee et al. “Stanford’s multi-pass sieve coreference resolution system at the conll-2011 shared task”. In: *Proceedings of the fifteenth conference on computational natural language learning: Shared task*. 2011, pp. 28–34.

[48] Kenton Lee, Luheng He, and Luke Zettlemoyer. “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)*. 2018, pp. 687–692.

[49] Kenton Lee et al. “End-to-end neural coreference resolution”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 188–197.

[50] Pierre Lison and Jörg Tiedemann. “Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles”. In: (2016).

[51] Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. “Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[52] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[53] Xiaoqiang Luo. “On coreference resolution performance metrics”. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 2005, pp. 25–32.

[54] Xiaoqiang Luo et al. “An extension of BLANC to system mentions”. In: *Proceedings of the conference. Association for Computational Linguistics. Meeting*. Vol. 2014. NIH Public Access. 2014, p. 24.

[55] Martín Abadi et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[56] Paulius Micikevicius et al. “Mixed Precision Training”. In: *International Conference on Learning Representations*. 2018.

[57] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *Proceedings of ICLR Workshops Track*. 2013.

[58] Nafise Sadat Moosavi and Michael Strube. “Which coreference evaluation metric do you trust? A proposal for a link-based entity aware metric”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 632–642.

[59] Mathias Müller et al. “A Large-Scale Test Set for the Evaluation of Context-Aware Pronoun Translation in Neural Machine Translation”. In: *WMT 2018*. Brussels, Belgium: Association for Computational Linguistics, 2018.

[60] Vincent Ng. “Machine learning for entity coreference resolution: A retrospective look at two decades of research”. In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[61] Vincent Ng. “Supervised noun phrase coreference research: The first fifteen years”. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. 2010, pp. 1396–1411.

[62] Christopher Olah. *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Last accessed 2022-04-01. 2015.

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

[64] Matthew E. Peters et al. “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)*. 2018, pp. 2227–2237.

[65] Duy Phung et al. “Learning Cross-lingual Representations for Event Coreference Resolution with Multi-view Alignment and Optimal Transport”. In: *Proceedings of the 1st Workshop on Multilingual Representation Learning*. 2021, pp. 62–73.

[66] Telmo Pires, Eva Schlinger, and Dan Garrette. “How multilingual is multilingual BERT?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4996–5001.

[67] Sameer Pradhan et al. “Conll-2011 shared task: Modeling unrestricted coreference in ontonotes”. In: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*. 2011, pp. 1–27.

[68] Sameer Pradhan et al. “CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes”. In: *Joint Conference on EMNLP and CoNLL-Shared Task*. 2012, pp. 1–40.

[69] Sameer Pradhan et al. “Scoring coreference partitions of predicted mentions: A reference implementation”. In: *Proceedings of the conference. Association for Computational Linguistics. Meeting*. Vol. 2014. NIH Public Access. 2014, p. 30.

[70] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018). URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[71] Pranav Rajpurkar et al. “Squad: 100,000+ questions for machine comprehension of text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 2383–2392.

[72] Marta Recasens and Eduard Hovy. “BLANC: Implementing the rand index for coreference evaluation”. In: *Natural Language Engineering* 17.4 (2011), pp. 485–510.

[73] Marta Recasens et al. “Semeval-2010 task 1: Coreference resolution in multiple languages”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. 2010, pp. 1–8.

[74] Matthew Richardson and Pedro Domingos. “Markov logic networks”. In: *Machine learning* 62.1 (2006), pp. 107–136.

[75] Frank Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[76] Ina Rösiger and Jonas Kuhn. “IMS HotCoref DE: A data-driven co-reference resolver for German”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 155–160.

[77] Ina Rösiger and Arndt Riester. “Using prosodic annotations to improve coreference resolution of spoken text”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2015, pp. 83–88.

[78] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation.* Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[79] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *NATURE* 323.6088 (1986), pp. 533–536.

[80] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2019).

[81] Fynn Schröder, Hans Ole Hatzel, and Chris Biemann. "Neural end-to-end coreference resolution for German in different domains". In: *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021).* 2021, pp. 170–181.

[82] Ankit Srivastava et al. "Different German and English coreference resolution models for multi-domain content curation scenarios". In: *International Conference of the German Society for Computational Linguistics and Language Technology.* Springer. 2017, pp. 48–61.

[83] Rhea Sukthanker et al. "Anaphora and coreference resolution: A review". In: *Information Fusion* 59 (2020), pp. 139–162.

[84] Heike Telljohann et al. "Stylebook for the Tübingen treebank of written German (TüBa-D/Z)". In: (2017).

[85] Don Tuggener. "Incremental coreference resolution for German". PhD thesis. University of Zurich, 2016.

[86] Gorka Urbizu, Ander Soraluze, and Olatz Arregi. "Deep cross-lingual coreference resolution for less-resourced languages: The case of basque". In: *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference.* 2019, pp. 35–41.

[87] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems.* 2017, pp. 5998–6008.

[88] Marc Vilain et al. "A model-theoretic coreference scoring scheme". In: *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995.* 1995.

[89] E Voita et al. "Context-aware neural machine translation learns anaphora resolution". In: *ACL 2018-56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers).* 2018, pp. 1264–1274.

[90] Elena Voita, Rico Sennrich, and Ivan Titov. "When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2019, pp. 1198–1212.

[91] Alex Waibel et al. "Phoneme recognition using time-delay neural networks". In: *IEEE transactions on acoustics, speech, and signal processing* 37.3 (1989), pp. 328–339.

[92] Alex Wang et al. "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.* 2018, pp. 353–355.

[93] Ralph Weischedel et al. "OntoNotes Release 5.0". In: (2013).

[94] Lesly Miculicich Werlen and Andrei Popescu-Belis. “Using coreference links to improve spanish-to-english machine translation”. In: *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017).* 2017, pp. 30–40.

[95] Lesly Miculicich Werlen et al. “Document-level neural machine translation with hierarchical attention networks”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* 2018, pp. 2947–2954.

[96] Thomas Wolf et al. “Transformers: State-of-the-art natural language processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* 2020, pp. 38–45.

[97] Wei Wu et al. “CorefQA: Coreference resolution as query-based span prediction”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* 2020, pp. 6953–6963.

[98] Yingce Xia et al. “Deliberation networks: Sequence generation beyond one-pass decoding”. In: *Advances in neural information processing systems* 30 (2017).

[99] Liyan Xu and Jinho D Choi. “Revealing the myth of higher-order inference in coreference resolution”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* 2020, pp. 8527–8533.

A. Appendix

A.1. PyTorch Implementation Performance

Table A.1 shows the detailed coreference score on the English portion of the OntoNotes 5.0 test set [93] achieved by the PyTorch implementation discussed in Section 4.2. Besides the results obtained by following the original training procedures of the models described in Section 4.1 I also report the results for training the *e2e-model* as well as the *bert-model* with BERT$_{\text{BASE}}$ and SpanBERT$_{\text{BASE}}$ for longer.

	MUC			B^3			*CAEF*			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
e2e-model	78.4	73.3	75.8	68.8	61.4	64.9	62.6	59.6	61.0	67.2
+ 500k Steps	78.7	73.2	75.9	69.1	61.5	65.1	63.0	59.4	61.2	67.4
c2f-model	82.4	76.7	79.4	72.8	65.4	68.9	68.6	61.4	64.8	71.0
BERT$_{\text{BASE}}$	82.5	79.8	81.1	72.5	69.8	71.1	70.0	65.9	67.9	73.4
+ 30 Epochs	82.6	80.2	81.4	73.6	70.7	72.1	71.2	66.9	69.0	74.2
BERT$_{\text{LARGE}}$	84.2	82.8	83.5	76.1	74.4	75.2	73.7	70.5	72.0	76.9
SpanBERT$_{\text{BASE}}$	83.6	83.1	83.4	75.1	75.5	75.3	74.0	71.2	72.6	77.1
+ 30 Epochs	84.3	83.4	83.9	76.2	75.8	76.0	74.9	71.3	73.0	77.6
SpanBERT$_{\text{LARGE}}$	85.5	85.0	85.3	78.4	78.0	78.2	76.3	74.9	75.6	79.7

Table A.1.: Coreference scores of the PyTorch implementation on the OntoNotes 5.0 test set.

A.2. Comparison of Coreference Corpora

Table A.2 shows the sizes of the different coreference annotated corpora used in this thesis. The values for the OntoNotes 5.0 corpora [93] refer to the English portion of that dataset. For splitting the TüBa-D/Z v10 corpus [84] into training, development, and test set, I follow the boundaries described in [76]. SemEval-2010 describes the split of the TüBa-D/Z v8 defined in the coreference resolution shared task [73]. The DIRNDL corpora [7] used in this thesis is identical to the version used in [77].

	TRAINING			*DEVELOPMENT*			*TEST*		
	Docs	Sents	Tokns	Docs	Sents	Tokns	Docs	Sents	Tokns
OntoNotes 5.0	2,802	73,629	1,299,312	343	9,308	163,104	348	9,386	169,579
TüBa-D/Z v10	2,190	65,416	1,258,512	727	15,593	276,635	727	14,586	252,652
SemEval-2010	900	19,233	331,614	199	4,129	73,145	136	2,736	50,287
DIRNDL	294	1,492	22,408	55	238	3,570	132	753	12,656

Table A.2.: Sizes of English and German coreference corpora

A.3. Hyperparameters for English Coreference Resolution

In order to obtain results on the OntoNotes 5.0 comparable to the results in [38] and [39], I tuned the hyperparameters in a similar way. For the segment sizes of 128 and 512, I tried the BERT and task learning rates suggested in [39]. That includes the values {1e-5, 2e-5} as BERT learning rates and {1e-4, 2e-4, 3e-4} as task-specific learning rates. Table A.3 shows the best parameters for the *bert-model* with different pre-trained language models. For the ELECTRA$_{\text{LARGE}}$, I took over the learning rates from its base-sized counterpart.

	Segments		Learning Rates	
	NUM	SIZE	BERT	TASK
RoBERTa$_{\text{BASE}}$	3	512	1e-5	2e-4
DistilBERT (cased)	11	128	2e-5	3e-4
DistilBERT (uncased)	11	128	2e-5	3e-4
DistilRoBERTa	11	128	1e-5	2e-4
TinyBERT$_4$	11	128	2e-5	3e-4
TinyBERT$_6$	11	128	1e-5	3e-4
ELECTRA$_{\text{SMALL}}$	3	512	1e-5	3e-4
ELECTRA$_{\text{BASE}}$	3	512	1e-5	2e-4
ELECTRA$_{\text{LARGE}}$	11	512	1e-5	2e-4

Table A.3.: Number of segment and their sizes as well as BERT and tasks learning rates of the *bert-model* for different language models.

In general, changing the learning rates within the specified scope only has a minor impact on the performance on the OntoNotes 5.0 development set, whereas the segment size has a much greater influence.

A.4. Hyperparameters for Cross-Lingual Coreference Resolution

Learning rates were optimized for fine-tuning cross-lingual coreference resolution systems on German after previously training them on English. For various numbers of epochs, the best learning rate for the underlying language model and the coreference layers on top were determined on all three German corpora. Table A.4 shows the parameters which were found. Interestingly the language model learning rate is pretty similar for each dataset and language

	TüBa-D/Z v10		SemEval-2010		DIRNDL	
	LR BERT	LR TASK	LR BERT	LR TASK	LR BERT	LR TASK
M-BERT (1 Epoch)	1e-5	2e-4	2e-5	2e-5	1e-5	2e-5
M-BERT (5 Epochs)	1e-5	2e-4	2e-5	2e-5	1e-5	2e-6
M-BERT (10 Epochs)	1e-5	2e-4	1e-5	1e-4	1e-5	1e-4
M-BERT (20 Epochs)	-	-	-	-	1e-5	2e-6
XLM-R (1 Epoch)	1e-5	2e-7	1e-5	2e-5	4e-6	2e-6
XLM-R (5 Epochs)	9e-6	2e-5	2e-5	2e-5	9e-6	2e-5
XLM-R (10 Epochs)	1e-5	2e-6	1e-5	2e-6	1e-6	2e-5
XLM-R (20 Epochs)	-	-	-	-	1e-6	2e-5

Table A.4.: BERT and tasks learning rates for fine-tuning English model on German datasets.

model, whereas the task learning rate varies over a broader range and does not seem to be very sensitive. Overall the learning rates on the larger TüBa-D/Z v10 corpus are more consistent than on smaller datasets and are almost identical to the learning rates used in [38].

A.5. Adversarial Cross-Lingual Learning

To evaluate if the adversarial task had the intended effect on the language model, Keung et al. measured the cosine similarity of the mean pooled embeddings on parallel data before and after training the language model simultaneously on the downstream and adversarial task [42]. Table A.5 shows the cosine similarity between the English and German embeddings on parallel and non-parallel data. While the parallel data consists of 10,000 documents randomly sampled from the WMT17 dataset [8], the non-parallel data are randomly joint documents from the English OntoNotes 5.0 [93] and the German TüBa-D/Z v10 [84] datasets.

	Parallel Data	Non-Parallel Data
Multilingual BERT	0.75408	0.66266
+ Fine-tuned on Coreference-Task	0.86427	0.77595
+ Adversarial CL Learning	0.91110	0.82981

Table A.5.: Comparison of the cosine similarities of English and German embeddings, using M-BERT with and without adversarial cross-lingual learning. The similarities are evaluated on parallel and non-parallel data.

The adversarial cross-lingual learning does, indeed, lead to a higher cosine similarity on the parallel data as intended and similar to what was reported in [42]. However, the similarity for non-parallel data also increases, despite the English and German sentences being no translations of each other and completely independent. Also noteworthy is that most of the increase in similarity is already achieved by fine-tuning the general M-BERT$_{\text{BASE}}$ on the coreference resolution task without the adversarial task. This suggests that the adversarial cross-lingual learning forces the model to squeeze the embeddings into a smaller vector space rather than learning a meaningful mapping between English and German.

	Steps / Sec	ContraPro Score	BLEU Score
base-model	0.4613	47.82	31.00
+ Input Cluster Labels	0.3593	49.78	31.09
+ Text-level Annotation	0.4551	64.42	30.58
+ Larger Base Model #1	0.2821	66.17	31.61
+ Larger Base Model #2	0.3160	65.36	31.08
+ Larger Base Model #3	0.1707	66.81	31.82
CADec	0.0645	60.87	31.70
+ Output Cluster Labels	0.0643	61.76	31.76
+ Input Cluster Labels	0.0660	62.61	31.84
+ Text-level Annotation	0.0660	62.58	31.87

Table A.6.: Speed, ContraPro and BLEU Score of NMT Systems

A.6. Performance of NMT Systems

Table A.6 shows the exact values used for Figure 6.4 in Section 6.4. The speed is defined as steps per second during the training. Detailed information about how the batch sizes are calculated for the *base-model* and the *CADec* can be found in the corresponding repository.[1] The pronoun score is evaluated on the complete 12,000 examples of the ContraPro test set. For the BLEU score, the performance on the WMT17 [8] development data described in Section 6.3.1 is reported. The snapshot for each configuration is picked by the highest ContraPro score.

A.7. Coreferences in ContraPro Test Set

The ContraPro test set was created with the help of a coreference resolution system to find suitable samples from a much larger parallel corpus automatically. Therefore not all coreferent mentions of the pronoun in question may have been detected, and the distance information provided for each sample might be incorrect. Missed intra-sentence coreferences are of particular interest since they could be exploited by a context-agnostic model even though the incorrect distance information suggests otherwise. Table A.7 shows how many coreferences for the pronoun *it* in the source sentence were found by my coreference system despite ContraPro stating greater sentence distances. A manual random evaluation indicates that in many of these cases, the coreferent mention can actually provide additional information helpful for translating the pronoun.

	DISTANCE			
	1	2	3	>3
total samples	7,075	1,510	573	442
found coreferences	940	218	91	62
filtered coreferences	130	31	8	12

Table A.7.: Number of intra-sentence coreferences not considered in the ContraPro test set, but detected by my coreference resolver. For the filtered coreferences the mentions *it*, *its* and *itself* were excluded, since they most probably do not provide additional information for the NMT system.

[1] https://github.com/lena-voita/good-translation-wrong-in-context#batch-size