Coreference Resolution through a seq2seq Transition-Based System

Bernd Bohnet[1], **Chris Alberti**[2], **Michael Collins**[2]
[1]Google Research, The Netherlands [2]Google Research, USA
{bohnetbd,chrisalberti,mjcollins}@google.com

Abstract

Most recent coreference resolution systems use search algorithms over possible spans to identify mentions and resolve coreference. We instead present a coreference resolution system that uses a text-to-text (seq2seq) paradigm to predict mentions and links jointly. We implement the coreference system as a transition system and use multilingual T5 as an underlying language model. We obtain state-of-the-art accuracy on the CoNLL-2012 datasets with 83.3 F1-score for English (a 2.3 higher F1-score than previous work [Dobrovolskii, 2021]) using only CoNLL data for training, 68.5 F1-score for Arabic (+4.1 higher than previous work), and 74.3 F1-score for Chinese (+5.3). In addition we use the SemEval-2010 data sets for experiments in the zero-shot setting, a few-shot setting, and supervised setting using all available training data. We obtain substantially higher zero-shot F1-scores for 3 out of 4 languages than previous approaches and significantly exceed previous supervised state-of-the-art results for all five tested languages. We provide the code and models as open source.[1]

1 Introduction

There has been a great deal of recent research in pretrained language models that employ encoder-decoder or decoder-only architectures (e.g., see GPT-3, GLAM, Lamda [Brown et al., 2020; Du et al., 2021; Thoppilan et al., 2022]), and that can generate text using autoregressive or text-to-text (seq2seq) models (e.g., see T5, MT5 [Raffel et al., 2019; Xue et al., 2021]). These models have led to remarkable results on a number of problems.

Coreference resolution is the task of finding referring expressions in text that point to the same entity in the real world. Coreference resolution is a core task in NLP, relevant to a wide range of applications (e.g., see Jurafsky and Martin [2021] Chapter 21 for discussion), but somewhat surprisingly, there has been relatively limited work on coreference resolution using encoder-decoder or decoder-only architectures.

The state-of-the-art models on coreference problems are based on encoder-only models, such as BERT (Devlin et al., 2019) or SpanBERT (Joshi et al., 2020). All recent state-of-the-art coreference models (see Table 2), however, have the disadvantage of a) requiring engineering of a specialized search or structured prediction step for coreference resolution, on top of the encoder's output representations; b) often requiring a pipelined approach with intermediate stages of prediction (e.g., mention detection followed by coreference prediction); and c) an inability to leverage more recent work in pretrained seq2seq models.

This paper describes a text-to-text (seq2seq) approach to coreference resolution that can directly leverage modern encoder-decoder or decoder-only models. The method takes as input a sentence at a time, together with prior context, encoded as a string, and makes predictions corresponding to coreference links. The method has the following advantages over previous approaches:

- **Simplicity:** We use greedy seq2seq prediction without a separate mention detection step and do not employ a higher order decoder to identify links.
- **Accuracy:** The accuracy of the method exceeds the previous state of the art.
- **Text-to-text (seq2seq) based:** The method can make direct use of modern generation models that employ the generation of text strings as the key primitive.

A key question that we address in our work is how to frame coreference resolution as a seq2seq problem. We describe three transition systems, where the seq2seq model takes a single sentence

[1]https://github.com/google-research/google-research/tree/master/coref_mt5.

Transactions of the Association for Computational Linguistics, vol. 11, pp. 212–226, 2023. https://doi.org/10.1162/tacl_a_00543
Action Editor: Vincent Ng. Submission batch: 4/2022; Revision batch: 10/2022; Published 3/2023.

Input: *Speaker-A* I still have n't gone to that fresh French restaurant by your house
Prediction: SHIFT: next sentence

Input: *Speaker-A* I_2 still have n't gone to that fresh French restaurant by your house *Speaker-A* I_{17} 'm like dying to go there
Prediction:
A $I_{17} \rightarrow I_2$
B SHIFT: next sentence

Input: *Speaker-A* [1 I] still have n't gone to that fresh French restaurant by your house *Speaker-A* [1 I] 'm like dying to go there *Speaker-B* You mean the one right next to the apartment
Prediction:
A You → [1
B the apartment → your house
C the one right next to the apartment → that fresh French restaurant by your house
D SHIFT: next sentence

Input: *Speaker-A* [1 I] still have n't gone to [3 that fresh French restaurant by [2 your house]] *Speaker-A* [1 I] 'm like dying to go there *Speaker-B* [1 You] mean [3 the one right next to [2 the apartment]] *Speaker-B* yeah yeah yeah
Prediction: SHIFT: next sentence

Figure 1: Example of one of our transition-based coreference systems, the *Link-Append* system. The system processes a single sentence at a time, using an input encoding of the prior sentences annotated with coreference clusters, followed by the new sentence. As output, the system makes predictions that link mentions in the new sentence to either previously created coreference clusters (e.g., ''You → [1'') or when a new cluster is created, to previous mentions (e.g., ''the apartment → your house''). The system predicts ''SHIFT'' when processing of the sentence is complete. Note in the figure we use the word indices 2 and 17 to distinguish the two incidences of ''I'' in the text.

as input, and outputs an action corresponding to a set of coreference links involving that sentence as its output. Figure 1 gives an overview of the highest performing system, ''Link-Append'' which encodes prior coreference decisions in the input to the seq2seq model, and predicts new conference links (either to existing clusters, or creating a new cluster) as its output. We provide the code and models as open source.[2] Section 4 describes ablations considering other systems, such as a ''Link-only'' system (which does not encode previous coreference decisions in the input), and mention-based (Mention-Link-Append), which has a separate mention detection system, in some sense mirroring prior work (see Section 5).

We describe results on the CoNLL-2012 data set in Section 4. In addition, Section 5 describes multilingual results, in two settings: first, the setting where we fine-tune on each language of interest; second, zero-shot results, where an MT5 model fine-tuned on English alone is applied to languages other than English. Zero-shot experiments show that for most languages, accuracies are higher than recent translation-based approaches and early supervised systems.

2 Related Work

Most similar to our approach is the work of Webster and Curran (2014), who use a shift-reduce transition-based system for coreference resolution. The transition system uses two data structures, a queue initialized with all mentions and a list. The SHIFT transition moves from the queue a mention to top of the list. The REDUCE transition merges the top mentions with selected clusters. Webster and Curran (2014) consider the approach to better reflect human cognitive processing, to be simple and to have small memory requirements. Xia et al. (2020) use this transition-based system together with a neural approach for mention identification and transition prediction; this neural model (Xia et al., 2020) gives higher accuracy scores (see Table 2) than Webster and Curran (2014).

Lee et al. (2017) focus on predicting mentions and spans using an end-to-end neural model based on LSTMs (Hochreiter and Schmidhuber, 1997), while Lee et al. (2018) extend this to a differentiable higher-order model considering directed paths in the antecedent tree.

Another important method to gain higher accuracy is to use stronger pretrained language models, which we follow in this paper as well. A number of recent coreference resolution systems kept the essential architecture fixed while they replace the pretrained models with increasingly stronger models. Lee et al. (2018) used *Elmo* (Peters et al., 2018) including feature tuning and show an impressive improvement of 5.1 F1 on the English CoNLL 2012 test set over the baseline score of Lee et al. (2017). The extension from an end-to-end to the differentiable higher-order inference provides an additional 0.7 F1-score on

[2] https://github.com/google-research/google-research/tree/master/coref_mt5.

the test set, which leads to a final F1-score of 73.0 for this approach. Joshi et al. (2019) use the same inference model and explore how to best use and gain another significant improvement of 3.9 points absolute and reach a score of 76.9 F1-score on the test set (see Table 2). Finally, Joshi et al. (2020) use SpanBERT, which leads to a even higher accuracy score of 79.6. SpanBERT performs well for coreference resolution due to its span-based pretraining objective.

Dobrovolskii (2021) considers coreference links between words instead of spans, which reduces the complexity to $O(n^2)$ of the coreference models and uses RoBERTa as language model, which provides better results than SpanBERT for many tasks.

Similarly, Kirstain et al. (2021) reduce the high memory footprint of mention detection by using the start- and end-points of mention spans to identify mentions with a bilinear scoring function. The top λn scored mentions are used to restrict the search space for coreferences prediction using again a bilinear function for scoring. The algorithm has a quadratic complexity since each possible coreference pair has to be scored.

Wu et al. (2020) cast coreference resolution as question answering and report gains originating from pretraining on Quoref and SQuAD 2.0 of 1 F1-score on the development set. The approach first predicts mentions with a recall-oriented objective, then creates queries for these potential mentions for the cluster prediction. This procedure requires the application of the model for each mention candidate multiple times per document, which leads to high execution time.

Our work makes direct use of T5-based models (Raffel et al., 2019). T5 adopts the idea of treating tasks in Natural Language Processing uniformly as ‘‘text-to-text’’ problems, which means to only have text as input and generate text as output. This idea simplifies and unifies the approach for a large number of tasks by applying the same model, objective, training procedure, and decoding process.

3 Three seq2seq Transition Systems

3.1 The Link-Append System

The Link-Append system processes the document a single sentence at a time. At each point the input to the seq2seq model is a text string that encodes the first i sentences together with coreference clusters that have been built up over the first $(i-1)$ sentences. As an example, the input for $i = 3$ for the example in Figure 1 is the following:

Input: *Speaker-A* [1 I] still have n’t gone to that fresh French restaurant by your house # *Speaker-A* [1 I] ’m like dying to go there | # *Speaker-B* You mean the one right next to the apartment **

Here the # symbol is used to delimit sentences, and the start of the focus sentence is marked using the pipe-symbol | and the end of a sentence with two asterisk symbols **.

We have three sentences ($i = 3$). There is a single coreference cluster in the first $i - 1 = 2$ sentences, marked using the [1 . . .] bracketings.

The output from the seq2seq model is also a text string. The text string encodes a sequence of 0 or more actions, terminated by the SHIFT token. Each action links some mention (a span) in the ith sentence to some mention in the previous context (often in the first $i - 1$ sentences, but sometimes also in the ith sentence). An example prediction given the above input is the following:

Prediction You → [1 ; the apartment → your house; the one right next to the apartment → that fresh French restaurant by your house ; SHIFT

More precisely, the first action would actually be ‘‘You ## mean the one → [1’’ where the substring ‘‘mean the one’’ is the 3-gram in the original text immediately after the mention ‘‘You’’. The 3-gram helps to disambiguate the mention fully, in the case where the same string might appear multiple times in the sentence of interest. For brevity we omit these 3-grams in the following discussion, but they are used throughout the models output to specify mentions.[3]

In this case there are three actions, separated by the ‘‘;’’ symbol, followed by the terminating SHIFT action. The first action is

[3]Note that no explicit constraints are placed on the model’s output, so there is the potential for the model to generate mention references that do not correspond to substrings within the input; however this happens very rarely in practice, see section 6.2 for discussion. There is also the potential for the 3-gram to be insufficient context to disambiguate the exact location of a mention; again, this happens rarely, see section 6.2.

You → [1

This is an **append** action: specifically, it appends the mention ‘‘You’’ in the third sentence to the existing coreference cluster labeled [1 . . .]. The second action is

the apartment → your house

This is a **link** action. It links the mention ‘‘the apartment’’ in the third sentence to ‘‘your house’’ in the previous context. Similarly the third action,

the one right next to the apartment → that fresh French restaurant by your house

is a also a link action, in this case linking the mention ‘‘the one right next to the apartment’’ to a previous mention in the discourse.

The sequence of actions is terminated by the SHIFT symbol. At this point the ith sentence has been processed, and the model moves to the next step where the $(i+1)$th sentence will be processed. Assuming the next sentence is ‘‘*Speaker-B* yeah yeah yeah’’, the input at the $(i+1)$th step will be

Input: *Speaker-A* [1 I] still have n’t gone to [3 that fresh French restaurant by [2 your house]] # *Speaker-A* [1 I] ’m like dying to go there # *Speaker-B* [1 You] mean [3 the one right next to [2 the apartment]] | # *Speaker-B* yeah yeah yeah

Note that the three actions in the previous prediction have been reflected in the new input, which now includes three coreference clusters, labeled [1 . . .], [2 . . .] and [3 . . .].

In summary, the method processes a sentence at a time, and uses **append** and **link** actions to build up links between mentions in the current sentence under focus and previous mentions in the discourse.

A critical question is how to map training data examples (which contain coreference clusters for entire documents) to sequences of actions for each sentence. Clearly there is some redundancy in the system, in that in many cases either link or append actions could be used to build up the same set of coreference clusters. We use the following method for creation of training examples:

- Process mentions in the order in which they appear in the sentence. Specifically, mentions are processed in order of their end-point (earlier end-points are earlier in the ordering). Ties are broken by their start-point (later start-points are earlier in the ordering). It can be seen that the order in the previous example, *You*, *the apartment*, *the one right next to the apartment*, follows this procedure.
- For each mention, if there is another mention in the same coreference cluster earlier in the document, either:
  1. Create an *append* action if there are at least two members of the cluster in the previous $i-1$ sentences.
  2. Otherwise create a *link* action to the most recent member of the coreference cluster (this may be either in the first $i-1$ sentences, or in the ith sentence).

The basic idea then will be to always use append actions where possible, but to use link actions where a suitable append action is not available.

3.2 The Link-only System

The Link-only system is a simple variant of the Link-Append system. There are two changes: First, the only actions in the Link-only system are link and SHIFT, as described in the previous section. Second, when encoding the input in the Link-only system, the first i sentences are taken again with the # separator, but no information about coreference clusters over the first $i-1$ sentences is included.

The Link-only system can therefore be viewed as a simplification of the Link-Append system. We will compare the two systems in experiments, in general seeing that the Link-Append system provides significant improvements in performance.

3.3 The Mention-Link-Append System

The Mention-Link-Append system is a modification of the Link-Append system, which includes an additional class of actions, the *mention* actions. A mention action selects a single sub-string from the sentence under focus, and creates a singleton coreference cluster. The algorithm that creates training examples is modified to have an additional step for the creation of *mention* actions, as follows:

- Process mentions in the order in which they appear in the sentence.

- For each mention, if it is the first mention in a coreference structure, introduce a *mention* action for that mention.
- For each mention, if there is another mention in the same coreference cluster earlier in the document, either:
    1. Create an *append* action if there is at least two members of the cluster in the previous $i-1$ sentences.
    2. Otherwise create a *link* action to the most recent member of the coreference cluster (this may be either in the first $i-1$ sentences, or in the ith sentence).

Note that the Mention-Link-Append system can create singleton coreference structures, unlike the LINK-APPEND or Link-only systems. This is its primary motivation.

3.4 A Formal Description

We now give a formal definition of the three systems. This section can be safely skipped on a first reading of the paper.

3.4.1 Initial Definitions and Problem Statement

We introduce some key initial definitions—of *documents*, *potential mentions*, and *clusterings*—before giving a *problem statement*:

Definition 1 (Documents). *A document is a pair $(w_1 \dots w_n, s_1 \dots s_m)$, where w_i is the ith word in the document, and $s_1, s_2, \dots, s_m$ is a sequence of integers specifying a segmentation of $w_1 \dots w_n$ into m sentences. Each s_i is the endpoint for sentence i in the document. Hence $1 \leq s_1 < s_2 \dots s_{m-1} < s_m$, and $s_m = n$. The ith sentence spans words $(s_{i-1}+1) \dots s_i$ inclusive (where for convenience we define $s_0 = 0$).*

Definition 2 (Potential Mentions). *Assume an input document $(w_1 \dots w_n, s_1 \dots s_m)$. For each $i \in 1 \dots m$ we define $\mathcal{M}_i$ to be the set of potential mentions in the ith sentence; specifically,*

$$\mathcal{M}_i = \{(a, b) : s_{i-1} < a \leq b \leq s_i\}$$

Hence each member of $\mathcal{M}_i$ is a pair (a, b) specifying a subspan of the ith sentence. We define

$$\mathcal{M} = \cup_{i=1}^{m} \mathcal{M}_i, \quad \mathcal{M}_{\leq i} = \cup_{j=1}^{i} \mathcal{M}_j$$

hence $\mathcal{M}$ is the set of all potential mentions in the document, and $\mathcal{M}_{\leq i}$ is the set of potential mentions in sentences $1 \dots i$.

Definition 3 (Clusterings). *A clustering K is a sequence of sets $K_1, K_2, \dots K_{|K|}$, where each $K_i \subseteq \mathcal{M}$, and for any i, j such that $i \neq j$, we have $K_i \cap K_j = \emptyset$. We in addition assume that for all i, $|K_i| \geq 2$ (although see Section 3.5 for discussion of the case where $|K_i| \geq 1$). We define $\mathcal{K}$ to be the set of all possible clusterings.*

Definition 4 (Problem Statement). *The coreference problem is to take a document x as input, and to predict a clustering K as the output. We assume a training set of N examples, $\{(x^{(i)}, K^{(i)})\}_{i=1}^{N}$, consisting of documents paired with clusterings.*

3.5 The Three Transition Systems

The transition systems considered in this paper take a document x as input, and produce a coreference clustering K as the output. We assume a definition of transition systems that is closely related to work on deterministic dependency parsing (Nivre, 2003, 2008), and which is very similar to the conventional definition of deterministic finite-state machines. Specifically, a transition system consists of: 1) A set of states $\mathcal{C}$. 2) An initial state $c_0 \in \mathcal{C}$. 3) A set of actions $\mathcal{A}$. 4) A transition function $\delta : C \times \mathcal{A} \rightarrow C$. This will usually be a partial function: That is, for a particular state c, there will be some actions a such that $\delta(c, a)$ is undefined. For convenience, for any state c we define $\mathcal{A}(c) \subseteq \mathcal{A}$ to be the set of actions such that for all $a \in \mathcal{A}(c)$, $\delta(c, a)$ is defined. 5) A set of final states $\mathcal{F} \subseteq \mathcal{C}$.

A path is then a sequence $c_0, a_0, c_1, a_1, \dots c_N$ where for $i = 1 \dots N$, $c_{i+1} = \delta(c_i, a_i)$, and where $c_N \in \mathcal{F}$.

All transition systems in this paper use the following definition of states:

Definition 5 (States). *A state is a pair (i, K) such that $1 \leq i \leq (m+1)$ and $K \in \mathcal{K}$ is a clustering such that for $k \in 1 \dots |K|$, for $j \in (i+1) \dots m$, $K_k \cap M_j = \emptyset$ (i.e,. K is a clustering over the mentions in the first i sentences). In addition we define the following:*

- *$\mathcal{C}$ is the set of all possible states.*
- *$c_0 = (1, \epsilon)$ is the initial state, where ϵ is the empty sequence.*

- $\mathcal{F} = \{(i, K) : (i, k) \in \mathcal{C}, i = (m + 1)\}$ *is the set of final states.*

Intuitively, the state (i, K) keeps track of which sentence is being worked on, through the index i, and also keeps track of a clustering of the partial mentions up to and including sentence i.

We now describe the actions used by the various transition systems. The actions will either augment the clustering K, or increment the index i. The actions fall into four classes—*link actions*, *append actions*, *mention actions*, and the *shift action*—defined as follows:

Link Actions. Given a state (i, K), we define the set of possible link actions as

$$L(i, K) = \{m \rightarrow m' : m \in \mathcal{M}_i, m' \in \mathcal{M}_{\leq i}\}$$

A link action $(m \rightarrow m')$ augments K by adding a link between mentions m and m'. We define $K \oplus (m \rightarrow m')$ to be the result of adding link $m \rightarrow m'$ to clustering K.[4] We can then define the transition function associated with a link action:

$$\delta((i, K), m \rightarrow m') = (i, K \oplus (m \rightarrow m'))$$

Append Actions. Given a state (i, K), we define the set of possible append actions as

$$\text{App}(i, K) = \{m \rightarrow k : m \in \mathcal{M}_i, k \in \{1 \ldots |K|\}\}$$

An append action $(m \rightarrow k)$ augments K by adding mention m to the cluster K_k withing the sequence K. We define $K \oplus (m \rightarrow k)$ to be the result of this action (thereby overloading the $\oplus$ operator); the transition function associated with an append action is then

$$\delta((i, K), m \rightarrow k) = (i, K \oplus (m \rightarrow k))$$

[4]Specifically, the addition of the link $m \rightarrow m'$ can either: 1) create a new cluster within K, if neither m or m' are in an existing cluster within K; 2) add m to an existing cluster within K, if m' is already in some cluster in K, and m is not in an existing clustering; 3) add m' to an existing cluster within K, if m is already in some cluster in K, and m' is not in an existing clustering; 4) merge two clusters, if m and m' are both in clusters within K, and the two clusters are different; 5) leave K unchanged, if m and m' are both within the same existing cluster within K. In practice cases (2), (3), (4), and (5) are never seen in oracle sequences of actions, but for completeness we include them.

Mention Actions. Given a state (i, K), we define the set of possible mention actions as

$$\text{Mention}(i, K) = \{\text{Add}(m) : m \in \mathcal{M}_i\}$$

A mention action $\text{Add}(m)$ augments K by either creating a new singleton cluster containing m alone, assuming that m does not currently appear in K; otherwise it leaves K unchanged. We define $K \oplus \text{Add}(m)$ to be the result of this action, and $\delta((i, K), \text{Add}(m)) = (i, K \oplus \text{Add}(m))$.

The SHIFT Action. The final action in the system is the SHIFT action. This can be applied in any state, and simply advances the index i, leaving the clustering K unchanged:

$$\delta((i, K), \text{SHIFT}) = ((i + 1), K)$$

We are now in a position to define the transition systems:

Definition 6 (The Three Transition Systems). *The link-append transition system is defined as follows:*

- $\mathcal{C}$, c_0, *and* $\mathcal{F}$ *are as defined in definition 5.*
- *For any state* (i, K)*, the set of possible actions is* $\mathcal{A}(i, K) = L(i, K) \cup \textit{App}(i, K) \cup \{\textit{SHIFT}\}$*. The full set of actions is* $\mathcal{A} = \cup_{(i,K) \in \mathcal{C}} \mathcal{A}(i, K)$
- *The transition function* δ *is as defined above.*

The Link-only system is identical to the above, but with $\mathcal{A}(i, K) = L(i, K) \cup \{\textit{SHIFT}\}$*. The Mention-Link-Append system is identical to the above, but with* $A(i, K) = L(i, K) \cup \textit{App}(i, K) \cup \textit{Mention}(i, k) \cup \{\textit{SHIFT}\}$.

All that remains in defining the seq2seq method for each transition system is to: a) define an encoding of the state (i, K) as a string input to the seq2seq model; b) define an encoding of each type of action, and of a sequence of actions corresponding to single sentence; c) defining a mapping from a training example consisting of an (x, K) pair to a sequence of input-output texts corresponding to training examples.

4 Experimental Setup

We train a mT5 model to predict from an input a target text. We use the provided training, development, and test splits as described in section 4.1.

Language	Training docs	Training tokens	Development docs	Development tokens	Test docs	Test tokens
OntoNotes / CoNLL-2012 datasets						
English	1940	1.3M	343	160k	348	170k
Chinese	1729	750k	254	110k	218	90k
Arabic	359	300k	44	30k	44	30k
SemEval 2010 data						
Catalan	829	253k	142	42k	167	49k
Dutch	145	46k	23	9k	72	48k
German	900	331k	199	73k	136	50k
Italian	80	81k	18	16k	46	41k
Spanish	875	284k	140	44k	168	51k

Table 1: Sizes of the SemEval Shared Task data sets and OntoNotes (CoNLL-2012).

For the preparation of the input text, we follow previous work and include the speaker in the input text before each sentence (Wu et al., 2020) as well as the text genre at the document start if this information is available in the corpus. We apply as described in Section 3 the corresponding transitions as an oracle to obtain the input and target texts. We shorten the text at the front if the input text is larger as the sentence piece token input size of the language model and add further context beyond the sentence i when the input space is not filled up (note that as described in Section 3, we use the pipe symbol | to mark the start of the focus sentence and the end with two asterisk symbols **).

4.1 Data

We use the English coreference resolution dataset from the CoNLL-2012 Shared Task (Pradhan et al., 2012) and SemEval-2010 Shared Task set (Recasens et al., 2010) for multilingual coreference resolution experiments. The SemEval-2010 datasets include six languages and is therefore a good test bed for multilingual coreference resolution. We excluded English as the data overlaps with our training data.

The statistics on the dataset sizes are summarized in Table 1. The table shows that the English CoNLL-2012 Shared Task is substantially larger than any of the other data sets.

4.2 Experiments

Setup for English. For our experiments, we use mT5 and initialize our model with either the `xl` or `xxl` checkpoints.[5] For fine-tuning, we use the hyperparameters suggested by Raffel et al. (2019): a batch-size of 128 sequences and a constant learning rate of 0.001. We use micro-batches of 8 to reduce the memory requirements. We save checkpoints every 2k steps. From these models, we select the model with the best development results. We train for 100k steps. We use inputs with 2048 sentence piece tokens and 384 output tokens for training. All our models have been tested with 3k sentence piece tokens input length if not stated otherwise. The training of the `xxl`-model takes about 2 days on 128 TPUs-v4. On the development set, inference takes about 30 minutes on 8 TPUs.

Setup for Other Languages. We used the English model in this work to continue training with the above settings on other languages than English (Arabic, Chinese, and the SemEval-2010 datasets). For few-shot learning, we use the first 10 documents for each language and we train only for 200 steps since the evaluation then shows 100% fit to the training set.

The experimental evaluation changed in recent publication for the SemEval-2010 by reporting F1-scores as an average of MUC, B^3, and $CEAF_{\Phi_4}$ following the CoNLL-2012 evaluation schema (Roesiger and Kuhn, 2016; Schröder et al., 2021; Xue et al., 2021). We follow this schema in this paper as well. Another important difference between the SemEval-2010 and the CoNLL-2012 datasets is the annotation of singletons (mentions without antecedents) in the SemEval datasets. Most recent systems predict only coreference chains. This has lead also to different evaluation methods for the SemEval-2010 datasets. The first method keeps the singletons for the evaluation purposes (e.g., Xia and Durme, 2021) and the second excludes the singletons from evaluation set (e.g., Roesiger and Kuhn, 2016; Schröder et al., 2021; Bitew et al., 2021). The exclusion of singletons seems better suited to compare recent systems but makes direct comparison with previous work difficult. We report numbers for both setups.

In Section 5, we present our work on multilingual coreference resolution and Section 6 discusses the results for all languages.

[5] https://github.com/google-research/multilingual-t5.

	LM	Decoder	MUC			B^3			$CEAF_{\Phi_4}$			Avg.
			P	R	F1	P	R	F1	P	R	F1	F1
English												
Lee et al. (2017)	–	neural e2e	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
Lee et al. (2018)	Elmo	c2f	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Joshi et al. (2019)	BERT	c2f	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
Yu et al. (2020)	BERT	Ranking	82.7	83.3	83.0	73.8	75.6	74.7	72.2	71.0	71.6	76.4
Joshi et al. (2020)	SpanBERT	c2f	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Xia et al. (2020)	SpanBERT	transitions	85.7	84.8	85.3	78.1	77.5	77.8	76.3	74.1	75.2	79.4
Wu et al. (2020)	SpanBERT	QA	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	*83.1*[*]
Xu and Choi (2020)	SpanBERT	hoi	85.9	85.5	85.7	79.0	78.9	79.0	76.7	75.2	75.9	80.2
Kirstain et al. (2021)	LongFormer	bilinear	86.5	85.1	85.8	80.3	77.9	79.1	76.8	75.4	76.1	80.3
Dobrovolskii (2021)	RoBERTa	c2f	84.9	87.9	86.3	77.4	82.6	79.9	76.1	77.1	76.6	81.0
Link-Append	mT5	transition	87.4	88.3	87.8	81.8	83.4	82.6	79.1	79.9	79.5	**83.3**
Arabic												
Aloraini et al. (2020)	AraBERT	c2f	63.2	70.9	66.8	57.1	66.3	61.3	61.6	65.5	63.5	63.9
Min (2021)	GigaBERT	c2f	73.6	61.8	67.2	70.7	55.9	62.5	66.1	62.0	64.0	64.6
Link-Append	mT5	transition	71.0	70.9	70.9	66.5	66.7	66.6	68.3	68.6	68.4	**68.7**
Chinese												
Xia and Durme (2021)	XLM-R	transition	–	–	–	–	–	–	–	–	–	69.0
Link-Append	mT5	transition	81.5	76.8	79.1	76.1	69.9	72.9	74.1	67.9	70.9	**74.3**

Table 2: English, Arabic, and Chinese test set results and comparison with previous work on the CoNLL-2012 Shared Task test data set. The average F1 score of MUC, B^3, and $CEAF_{\Phi_4}$ is the main evaluation criterion. [*]Wu et al. (2020) use additional training data.

5 Multilingual Coreference Resolution Results

5.1 Zero-Shot and Few-Shot

Since mT5 is pretrained on 100+ languages (Xue et al., 2021), we evaluate Zero-Shot transfer ability from English to other languages. We apply our system trained on the English CoNLL-2012 Shared Task dataset to the non-English SemEval-2010 test sets. Table 3 shows evaluation scores for our transition-based systems and reference systems. In our results overview (Table 3), we report in the column *sing.* whether singletons are included (Y) or excluded (N), in the P-column for prediction and in the E-column for evaluation. We use for training the same setting as the reference systems (Kobdani and Schütze, 2010; Roesiger and Kuhn, 2016; Schröder et al., 2021). In the Zero-shot experiments, the transition-based systems are trained only on English CoNLL-2012 datasets and applied without modification to the multilingual SemEval-2010 test sets.

Bitew et al. (2021) use machine translation for the coreferences prediction of the SemEval-2010 datasets. The authors found they obtained the best accuracy when they first translated the test sets to English, then predicted the English coreferences with the system of Joshi et al. (2020) and finally projected back the predictions. They apply this method to four out of the six languages for the SemEval-2010 datasets. We include in Table 3 their results as a comparison to our Zero-Shot results. The two methods are directly comparable as they do not use the target language annotations for training. Our Zero-Shot F1-scores are substantially higher compared with the machine translation approach for Dutch, Italian, and Spanish and a bit lower for Catalan, cf. Table 3.

Xia and Durme (2021) explored for a large number of settings few-shot learning using the continued training approach. We use the same approach with a single setting that uses the first 10 documents for each language. For details about the experimental setup see Section 4.2. Table 3 presents the results for the Link-Append system. This shows that already with a few additional training documents a high accuracy can be reached. This could be useful either to adapt to a specific coreference annotations schema or to specific language (see examples in Figure 2 and 3).

5.2 Supervised

We also carried out experiments in a fully supervised setup in which we use all available

Systems	Sing. P	Sing. E	# training docs./method	Avg. F1
Catalan				
Attardi et al. (2010)	Y	Y	all	48.2
Mention-Link-Append	Y	Y	all	**83.5**
Xia and Durme (2021)	N	Y	all	51.0
Mention-Link-Append	N	Y	all	**59.2**
Bitew et al. (2021)	N	N	∅/Translation	**48.0**
Link-Append	N	N	∅/Zero-shot	47.7
Link-Append	N	N	10/Few-shot	68.9
Dutch				
Kobdani and Schütze (2010)	Y	Y	all	19.1
Mention-Link-Append	Y	Y	all	**66.6**
Xia and Durme (2021)	N	Y	all	55.4
Mention-Link-Append	N	Y	all	**59.9**
Bitew et al. (2021)	N	N	∅/Translation	37.5
Link-Append	N	N	∅/Zero-shot	**57.6**
Link-Append	N	N	10/Few-shot	65.7
German				
Kobdani and Schütze (2010)	Y	Y	all	59.8
Mention-Link-Append	Y	Y	all	**86.4**
Roesiger and Kuhn (2016)	N	N	all	48.6
Schröder et al. (2021)	N	N	all	74.5
Link-Append	N	N	all	**77.8**
Link-Append	N	N	∅/Zero-shot	55.0
Link-Append	N	N	10/Few-shot	69.8
Italian				
Kobdani and Schütze (2010)	Y	Y	all	60.7
Mention-Link-Append	Y	Y	all	**65.9**
Mention-Link-Append	N	N	all	59.4
Bitew et al. (2021)	N	N	∅/Translation	36.2
Link-Append	N	N	∅/Zero-shot	**39.4**
Link-Append	N	N	10/Few-shot	61.2
Spanish				
Attardi et al. (2010)	Y	Y	all	49.0
Mention-Link-Append	Y	Y	all	**83.9**
Xia and Durme (2021)	N	Y	all	51.3
Mention-Link-Append	N	Y	all	**59.3**
Link-Append	N	N	all	83.1
Bitew et al. (2021)	N	N	∅/Translation	46.1
Link-Append	N	N	∅/Zero-shot	**49.4**
Link-Append	N	N	10/Few-shot	72.5

Table 3: Test set results for SemEval-2010 datasets. The *Sing.* column shows whether the singletons are included (Y) or removed (N) in the **P**rediction and the **E**valuation set. The last column shows average F1 score of MUC, B^3, and $CEAF_{\Phi_4}$.

training data of the SemEval-2010 Shared Task. We adopted the method of continued training of Xia and Durme (2021). In our experiments, we start from our finetuned English model and continue training on the SemEval-2010 datasets and the Arabic OntoNotes dataset for the later we use data and splits of the CoNLL-2012 Shared Task.

To verify the finding of Xia and Durme (2021), we compared the results when we continue the training from a finetuned model and from the initial mT5 model. We conducted this exploratory experiment using 1k training steps for the German dataset. The results are in favor of the experiment with continued training using an already fine-tuned model with a score of 84.5 F1 vs 81.0 F1 for fresh mT5 model. This model also achieves 77.3 F1 when evaluated without singletons (cf. Table 3), surpassing previous SotA of 74.5 F1 (Schröder et al., 2021). We did not explore training longer due to the computational cost of training from a fresh mT5 model to reach a potentially better performance. We adopted the approach for all datasets of the SemEval-2010 Shared Task as this model provides competitive coreference models with low training cost.

Table 3 includes the accuracy scores for the cluster/mentions-based transition systems which reaches SotA for all languages when the prediction and evaluation includes the singletons (P=Y, E=Y). In order to compare the results with Xia and Durme (2021), we removed from the results for the cluster/mentions-based transition system the singletons in the prediction but still include them in the evaluation (P=N, E=Y).

Table 2 compares the results for Arabic and Chinese of our model with the recent work. The Link-Append system is 4.1 points better than Min (2021) and 5.3 points better than Xia and Durme (2021), which present previous SotA for Arabic and Chinese, respectively.

6 Discussion

In this section, we analyze performance factors with an ablation study, analyze errors, and reflect on design choices that are important for the model's performance. Table 2 shows the results for our systems on the English, Arabic, and Chinese CoNLL-2012 Shared Task and compares with previous work.

6.1 Ablation Study

Our best-performing transition system is Link-Append, which predicts links and clusters iteratively for sentences of a document without predicting mentions before-hand. Table 4 shows an ablation study. The results at the top of the

System	Ablation	F1
Link-Append	100k steps/3k pieces	83.2
Link-Append	2k sentence pieces	83.1
Link-Append	50k steps	82.9
Link-Append	no context beyond i	82.8
Link-Append	xxl-T5.1.1	82.7
Link-Append	xl-mT5	78.0
Mention-Link-Append	3k pieces	82.6
Mention-Link-Append	2k pieces	82.2
Link-only	link transitions only	81.4

Table 4: **Development set** results for an ablation study using English CoNLL-2012 data sets and reporting Avg. F1-scores. The models have been trained with 100k training steps and tested with 2k sentence pieces filling up remaining space in the input beyond the focus sentence i with further sentences of the document as context. In inference mode, the model uses input length of 3k sentences pieces if not stated otherwise.

table show the development set results for the Link-Append system when, with each SHIFT, the already identified coreference clusters are annotated in the input. This information is then available in the next step and the clusters can be extended by the APPEND transition.

The models are trained with an input size of 2048 tokens using mT5. We use a larger input size of 3000 (3k) tokens for decoding to accommodate for long documents and very long distances between mentions of coreferences. When we use 2k sentence pieces, the accuracy is 83.1 instead of 83.2 averaged F1-score on the development set using the model trained for 100k steps.

At the bottom of Table 4, the performance of a system is shown that does not annotate the identified clusters in the input. In this system the Append transition cannot be applied and hence only the Link and Shift transition are used. The accuracy of this system is substantially lower, by 1.8 F1-score.

We observe drops in accuracy when we do not use context beyond the sentence i or when we train for only 50k steps. We observe 0.5 lower F1-score, when we use xxl-T5.1.1[6] instead of the xxl-mT5 model. An analysis shows that the English OntoNotes corpus contains some

[6]The xxl-T5.1.1 model refers to a model provided by Xue et al. (2021) trained for 1.1 million steps on English data.

Subset	#Docs	JS-L	CM	LA
1 – 128	57	84.6	84.5	85.8
129 – 256	73	83.7	83.6	85.2
257 – 512	78	82.9	83.4	86.0
513 – 768	71	80.1	79.3	83.2
769 – 1152	52	79.1	78.6	83.3
1153+	12	71.3	69.6	74.9
all	343	80.1	79.5	83.2

Table 5: Average F1-score on the development set for buckets of document length incremented by 128 tokens. The column JS-L shows average F1-scores for the SpanBert-Large model (Joshi et al., 2020), CM for the Constant Memory model (Xia et al., 2020), and LA for the Link-Append system. The entries for JS-L and CM are taken from the paper of Xia et al. (2020).

non-English text, speaker names, and special symbols. For instance, there are Arabic names that are mapped to OOV, but also the curly brackets {}. There are also other cases where T5 translated non-English words to English (e.g., German 'nicht' to 'not').

With the Mention-Link-Append system, we introduced a system that is capable of introducing mentions, which is useful for data sets that include single mentions, such as the SemEval-2010 data set. This transition system has an 82.6 F1-score on the development set with an input context of 3k sentences pieces, which is 0.6 F1-score lower than the Link-Append transition system. We added examples in the Appendix to illustrate mistakes in a Zero-shot setting (Figure 2) and supervised English example (Appendix).

6.2 Error Analysis

We observe two problems originating from the sequence-to-sequence models: first, hallucinations (words not found in the input) and second, ambiguous matches of mentions to the input. In order to evaluate the frequency of hallucinations, we counted cases where the predicted mentions and their context could not be matched to a word sequence in the input. We found only 11 cases (0.07%) in all 14.5k Link and Append predictions for the development set. The second problem are mentions with their n-gram context which are found more than once in the input. This constitutes 84 cases (0.6%) of all 14.5k Link and Append predictions.

Table 5 shows average F1-scores for buckets of documents within a length range incremented by 128 tokens, analogous to the analysis of Xia et al. (2020). All systems' F1-scores drop after the segment length 257–512 substantially by about 3-4 points. The Link-Append (LA) system seems to have two more stable F1-score regions 1–512 and 513–1152 tokens divided by the mentioned larger drop while we see for the other system slightly lower accuracy in each segment.

6.3 Design Choice

With this paper, we follow the paradigm of a text-to-text approach. Our goal was to use only the text output from a seq2seq model, and potentially the score associated with the output. Crucial for the high accuracy of the Link-Append systems are the design choices that seem to fit a text-to-text approach well. (1) Initial experiments, not presented in the paper, showed lower performance for a standard two-stage approach using mention prediction followed by mention-linking. The Link-only transition system, which we included as a baseline in the paper, was the first system that we implemented that only predicted conference links, avoiding mention-detection. Hence this crucial first design choice is the prediction of links and not to predict mentions first. (2) The prediction of links in a state-full fashion, where the prior input records previous coreference decisions, finally leads to the superior accuracy for the text-to-text model. (3) The larger model enables us to use the simpler paradigm of a text-to-text model successfully. The smaller models provide substantially lower performance. We speculate in line with the arguments of Kaplan et al. (2020) that distinct capabilities of a model become strong or even emerge with model size. (4) The strong multilingual results originate from the multilingual T5 model, which was initially surprising to us. For English, the mT5 model performed better as well which we attribute to larger vocab of the sentence piece encoding model of mT5.

7 Conclusions

In this paper, we combine a text-to-text (seq2seq) language model with a transition-based systems to perform coreference resolution. We reach 83.3 F1-score on the English CoNLL-2012 data set surpassing previous SotA. In the text-to-text framework, the Link-Append transition system has been superior to hybrid Mention-Link-Append transition system with mixed prediction of mentions, links and clusters. Our trained models are useful for future work as they could be used to initialize models for continuous training or zero-shot transfer to new languages.

Acknowledgments

We would like to thank the action editor and three anonymous reviewers for their thoughtful and insightful comments, which were very helpful in improving the paper.

[1 Hausbesetzer " Prinz **"**] Hamburg (ap) - [1 Ein zwei Jahre alter Schäferhund namens " Prinz **"**] hat im Hamburger Stadtteil Altona [2 eine Wohnung] besetzt . [3 Der 24jährige Besitzer] hatte [1 dem Tier] am Vortag [2 **[3 sein]** zukünftiges Heim] **[4 gezeigt]**. **[4 Das]** gefiel [1 dem Hund] so gut , daß [1 er] unmittelbar hinter der Tür Stellung bezog und niemanden mehr durchließ. Als [5 ein Bekannter [3 des Hundehalters]] versuchte, **die Wohnung** zu räumen, wurde [5 er] gebissen und flüchtete ins Wohnzimmer zur Gattin [3 des Besitzers] . Erst die Feuerwehr konnte beide durch das Fenster befreien. [3 Herrchen] wollte den Hundefänger holen.

Figure 2: German Zero-shot predictions. The **red bold** marked text are wrong predictions.

In the summer of 2005 , a picture that people have long been looking forward to started emerging with frequency in various major [1 Hong Kong] media . With [2 their] unique charm , [2 these well - known cartoon images] once again caused [1 Hong Kong] to be a focus of worldwide attention . [4 The world 's fifth [3 Disney] park] will soon open to the public here . The most important thing about [3 Disney] is that [3 it] is a global brand . Well , for several years , although [4 it] was still under construction and , er , not yet open , it can be said that many people have viewed [1 Hong Kong] with new respect . Then welcome to the official writing ceremony of [4 **Hong Kong** Disneyland] . The construction of [4 **Hong Kong** Disneyland] began two years ago , in [5 2003] . In January of [5 that year] , [1 **the** Hong Kong **government**] turned over to [3 Disney Corporation] [6 200 hectares of land at the foot of [7 Lantau Island] that was obtained following the largest land reclamation project in recent years] . One . Since then , [6 this area] has become a prohibited zone in [1 Hong Kong] . As [8 its] neighbor on [7 Lantau Island] , [8 **Hong Kong** International Airport] had to change [8 its] flight routes to make [6 this area] a no - fly zone . [4 Mickey Mouse 's new home] , settling on Chinese land for the first time , has captured worldwide attention . There 's only one month left before the opening of [4 Hong Kong Disneyland] on September 12 . The subway to [4 Disney] has already been constructed . At subway stations , passengers will frequently press the station for [4 Disney] on ticket machines , trying to purchase tickets to enjoy [4 the park] when [4 it] first opens . Meanwhile , the **[3 Disney]** subway station is scheduled to open on the same day as [4 the park] ...

Figure 3: Mistakes picked from CoNLL-2012 development set, e.g., *Hong Kong* should have been identified recursively within *Hong Kong Disneyland*; in the last sentence, *[3 Disney]* refers to *[3 Disney Corporation]* cluster instead correctly to *[4 The world 's fifth Disney park]* cluster.

... but it is pretty clear that there 's **[18 a lot of blood]** there **[18 great deal of blood]** . Larry Kobilinsky: **[18 It]** makes you think that something criminal occured . Larry Kobilinsky: But again getting back to how you solve [9 the crime] you got to go to [12 the crime scene] . Larry Kobilinsky: and [15 I] think there are multiple scenes here . Larry Kobilinsky: Every place there 's blood it 's a crime scene . Larry Kobilinsky: But the most important place is [1 the cabin] because [1 that] 's presumably where the injury took place . Larry Kobilinsky: Now there was some rumor about some arguments going [21 casino] the night before . Larry Kobilinsky: Uh and it could very well be that [20 he] was arguing with [19 these individuals] . Larry Kobilinsky: so uh that could be the tie the connection . Larry Kobilinsky: and [15 I] think good policework would **[22 connect]** [19 the individuals in question] with [21 the casino] with [20 Mister Smith] . Larry Kobilinsky: and **[22 that]** might help us understand what happened . Dan Abrams: [23 this] is um a little bit more sound . Dan Abrams: [23 This] is again from [3 Cletus Hyman uh who was literally um in a joining or nearby cabin **um as to what** [3 he] **heard around** [3 he] **said four in the morning the day before uh** [20 Mister Smith] **went missing]** . Cletus Hyman: At times it sounded like furniture was being actually picked up and dropped . Cletus Hyman: and then [24 that horrific uh thud] . Dan Abrams: Yeah [7 I] mean that sure sounds to [7 me] [25 chief] like we 're not talking about someone who /- Dan Abrams: [7 I] mean [7 I] guess it 's possible if you 're talking about loud yelling coming /- Dan Abrams: Well look if [3 this guy] 's right that loud yelling is coming from [1 the cabin] right and the wife is not a real suspect here means that there was probably someone else in there . ... Walter Zalisko: But it was five days later that **[27 [13 the ship 's] attorneys]** had begun to question [26 her] not law enforcement but **[27 [13 ship 's] attorneys]** . Dan Abrams: We will uh continue to follow this . Dan Abrams: [25 Chief Zalisko of the Oakhill Police Department] sorry about that [11 Susan Filan] [15 Larry Kobilinsky] thanks a lot .

Figure 4: Mistakes picked from CoNLL-2012 development, e.g., the coreferences *[18 a lot of blood]* as well as [27 [13 the ship's] attorneys] are not in the gold annotation.

References

Abdulrahman Aloraini, Juntao Yu, and Massimo Poesio. 2020. Neural coreference resolution for Arabic. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 99–110, Barcelona, Spain (online). Association for Computational Linguistics.

Giuseppe Attardi, Maria Simi, and Stefano Dei Rossi. 2010. TANL-1: Coreference resolution by parse analysis and similarity clustering. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 108–111, Uppsala, Sweden. Association for Computational Linguistics.

Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2021. Lazy low-resource coreference resolution: A study on leveraging black-box translation tools. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 57–62, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

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

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.605

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2021. Glam: Efficient scaling of language models with mixture-of-experts. *CoRR*, abs/2112.06905.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735, PubMed: 9377276

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77. https://doi.org/10.1162/tacl_a_00300

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1588

Daniel Jurafsky and James H. Martin. 2021. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, third edition.

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

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. 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)*, pages 14–19, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-short.3

Hamidreza Kobdani and Hinrich Schütze. 2010. SUCRE: A modular system for coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 92–95, Uppsala, Sweden. Association for Computational Linguistics.

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

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

Bonan Min. 2021. Exploring pre-trained transformers and bilingual transfer learning for Arabic coreference resolution. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 94–99, Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.crac-1.10

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553. https://doi.org/10.1162/coli.07-056-R1-07-027

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

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Marta Recasens, Lluís Màrquez, Emili Sapena, M. Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. SemEval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8, Uppsala, Sweden. Association for Computational Linguistics. https://doi.org/10.3115/1621969.1621982

Ina Roesiger and Jonas Kuhn. 2016. 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)*, pages 155–160, Portorož, Slovenia. European Language Resources Association (ELRA).

Fynn Schröder, Hans Ole Hatzel, and Chris Biemann. 2021. Neural end-to-end coreference resolution for German in different domains. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS*

2021), pages 170–181, Düsseldorf, Germany. KONVENS 2021 Organizers.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239.

Kellie Webster and James R. Curran. 2014. Limited memory incremental coreference resolution. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2129–2139, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.622

Patrick Xia and Benjamin Durme. 2021. Moving on from OntoNotes: Coreference resolution model transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5241–5256, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8617–8624. Association for Computational Linguistics, Online.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Juntao Yu, Alexandra Uma, and Massimo Poesio. 2020. A cluster ranking model for full anaphora resolution. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 11–20, Marseille, France. European Language Resources Association.