PAIRSPANBERT: An Enhanced Language Model for Bridging Resolution

Anonymous ACL submission

Abstract

We present PAIRSPANBERT, a SPANBERT-based pre-trained model specialized for bridging resolution. To this end, we design a novel pre-training objective that aims to learn the contexts in which two mentions are implicitly linked to each other from a large amount of data automatically generated either heuristically or via distance supervision with a knowledge graph. Despite the noise inherent in the automatically generated data, we achieve the best results reported to date on three evaluation datasets for bridging resolution when replacing SPANBERT with PAIRSPANBERT in a state-of-the-art resolver that jointly performs entity coreference resolution and bridging resolution.

1 Introduction

Bridging is essential for establishing coherence among entities within a text through non-identical semantic or encyclopedic relations (Clark, 1975; Prince, 1981). As demonstrated in Example 1, the implicit link established through the *bridging anaphor* (**prices**) and its *antecedent* (***meat, milk and grain***) exemplifies the local entity coherence.

(1) In June, farmers held onto ***meat, milk and grain***, waiting for July's usual state directed price rises. The Communists froze **prices** instead.

The task of *bridging resolution*, which involves identifying all bridging anaphors in a text and link them to their antecedents, is crucial to comprehend the relations between discourse entities for various downstream applications, such as question answering (Anantha et al., 2021) and dialogue systems (Tseng et al., 2021).

The most successful natural language learning paradigm to date is arguably the "pre-train and fine-tune" paradigm, where a model is first pre-trained on very large amounts of data in a task-agnostic, self-supervised manner and then fine-tuned using a potentially small amount of task-specific training data in the usual supervised manner. This paradigm is ideally applicable to bridging resolution, where the amount of annotated training data is relatively small, especially in comparison to the related task of entity coreference resolution.[1] In fact, by using SPANBERT (Joshi et al., 2020) to encode the input and fine-tuning it using bridging-annotated data, Kobayashi et al. (2022b) have managed to achieve the best results reported to date on two commonly-used evaluation datasets for bridging resolution, namely ISNotes (Markert et al., 2012) and BASHI (Rösiger, 2018).

A natural question is: how can we build upon the successes of this pre-train and fine-tune framework for bridging resolution? Apart from achieving state-of-the-art results, Kobayashi et al. (2022b) showed that bridging resolution performance deteriorates when SPANBERT is replaced with BERT (Devlin et al., 2019) as the encoder. While it is perhaps not surprising that SPANBERT achieves better resolution results than BERT given its superior performance on a wide variety of NLP tasks, it is important to step back and understand why. Recall that SPANBERT is an extension of BERT that is motivated by entity-based information extraction (IE) tasks such as entity coreference and relation extraction. These tasks typically involve the extraction of entity mentions, which are text *spans*. In order to learn *span* (as opposed to word) representations, SPANBERT is pre-trained with *span-level* masking and objective. The key point here is that a pre-trained model tends to work better for a downstream task (which in our case is bridging resolution) if it is pre-trained on an objective that is in some sense "related" to the downstream task.

Motivated by this observation, we design a novel

[1] While one of the largest annotated entity coreference datasets, OntoNotes (Pradhan et al., 2012), is composed of 2802 English documents for model training, two of the most widely used English corpora for bridging resolution research, ISNotes and BASHI, only comprise 50 WSJ documents each.

pre-training objective for bridging resolution that allows a model to learn the *contexts* in which two mentions are implicitly linked to each other. We subsequently use our objective to further pre-train SpanBERT in combination with its original objectives, yielding PairSpanBERT, a pre-trained model that is intended to specialize in bridging resolution. Note that an important factor that contributes to the success of pre-training is the sheer amount of data the model is pre-trained on: since pre-training tasks are designed to be self-supervised learning tasks, a very large amount of annotated training data can be automatically generated, thus allowing the model to potentially acquire a large amount of linguistic and commonsense knowledge. To enable our model to learn contexts that are indicative of bridging, we employ a large amount of data that can be automatically generated either heuristically (Hou, 2018a) or via distance supervision using a knowledge graph.

While the vast majority of existing bridging resolvers were evaluated in the rather unrealistic setting where gold mentions were assumed as input, we follow Kobayashi et al.'s (2022b) recommendation and evaluate our bridging resolver in both the (realistic) end-to-end setting, where we assume raw text as input, and the gold mention setting, where gold mentions are given. When replacing SpanBERT with PairSpanBERT in Kobayashi et al's bridging resolver, we achieve the best results reported to date on three datasets for bridging resolution, ISNotes, BASHI, and ARRAU RST (Poesio and Artstein, 2008), in both evaluation settings despite the large amount of noise inherent in our automatically generated data. To our knowledge, this is the first work that reports end-to-end bridging resolution results on the ARRAU RST dataset.

2 Related Work

Bridging resolution. The two sub-tasks of bridging resolution, namely *bridging anaphora recognition* and *bridging anaphora resolution*, have been tackled separately. One line of research (Rahman and Ng, 2012; Hou et al., 2013a; Hou, 2020b) has modeled bridging anaphora recognition as a part of the information status (IS) classification problem, assigning each discourse entity to an IS category, with *bridging* being one of the categories. In contrast, bridging anaphora resolution focuses on identifying the antecedents for gold bridging anaphors (Poesio et al., 2004; Hou et al., 2013b; Pandit et al., 2020). There have been several studies addressing full bridging resolution, which involves recognizing bridging anaphors and determining their antecedents. These works include rule-based approaches (Hou et al., 2014; Rösiger et al., 2018), learning-based approaches (Hou et al., 2018; Yu and Poesio, 2020), and hybrid approaches (Kobayashi and Ng, 2021; Kobayashi et al., 2022a).

Recent studies have begun tackling bridging resolution and its sub-tasks in the end-to-end setting. For example, Hou (2021) uses a combination of neural mention extraction and IS classification models for bridging anaphora recognition. Furthermore, Hou (2020a) proposes a QA approach of rephrasing bridging anaphors as questions and training question answering models to directly extract antecedents from the previous context. Finally, there are a few works (Kim et al., 2021; Kobayashi et al., 2021; Li et al., 2022) proposing models for full bridging resolution in the end-to-end setting in the 2021 and 2022 CODI-CRAC shared tasks on Anaphora, bridging, and Discourse Deixis in Dialogue (Khosla et al., 2021; Yu et al., 2022). Recently, Kobayashi et al. (2022c) conduct a systematic evaluation of bridging resolvers using different standard encoders, including BERT (Devlin et al., 2019) and SpanBERT (Joshi et al., 2020), in the end-to-end setting.

Enhanced pre-trained language models. BERT (Devlin et al., 2019), which is based on the Transformer architecture (Vaswani et al., 2017), has recently attracted significant attention. Researchers have proposed methods to enhance it for a wide range of downstream tasks. One line of research focuses on improving the masking scheme and training objectives for pre-training models for tasks such as question answering and sentence selection (Ram et al., 2021; Ye et al., 2020; Di Liello et al., 2022). Another line of work focuses on incorporating external knowledge into the pre-trained models to solve knowledge-driven problems such as relation extraction (Liu et al., 2020; Qin et al., 2021).

3 The Current State of the Art

State-of-the-art results on ISNotes and BASHI were reported in Kobayashi et al. (2022b). As mentioned before, they first use SpanBERT to encode the input sentences and then feed the resulting span representations to a multi-task learning model originally proposed by Yu and Poesio (2020) that jointly learns entity coreference and bridging res-

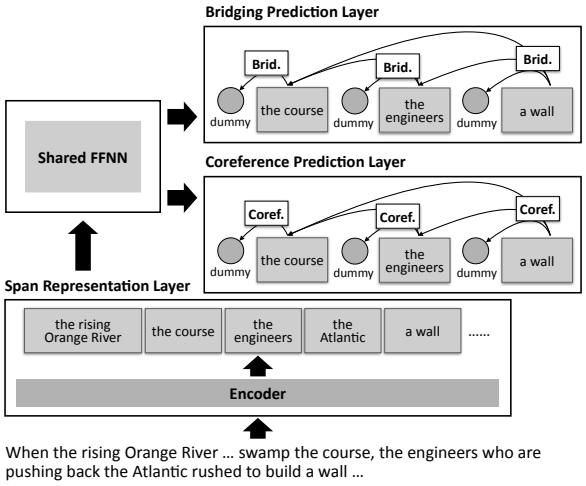

Figure 1: The MTL framework for bridging resolution.

olution. Since we aim to create PAIRSPANBERT, which specializes SPANBERT for bridging resolution, and eventually replace SPANBERT with PAIRSPANBERT in the aforementioned multi-task learning framework, in this section we will give an overview of the modified Yu and Poesio multi-learning framework described in Kobayashi et al. as well as the inner workings of SPANBERT.

3.1 The Multi-Task Learning Framework

The model takes as input a document D and either a set of gold mentions associated with D (in the gold mention setting) or a set of mentions extracted from D by a mention extractor (in the end-to-end setting). It simultaneously learns the two tasks.

The *bridging resolution* task involves assigning span i an antecedent $y_b \in \{1, ..., i-1, \epsilon\}$, where the value of y_b is the id of the span i's antecedent, which can be a dummy antecedent ϵ (i.e., i is not anaphoric) or one of the preceding spans. Yu and Poesio define the scoring function as below.

$$s_b(i,j) = \begin{cases} 0 & j = \epsilon \\ s_a(i,j) & j \neq \epsilon \end{cases} \quad (1)$$

where $s_a(i,j)$ is a pairwise bridging score which indicates how likely span i refers to a preceding span j. The model predicts the antecedent of i to be $y_b^* = \arg\max_{j \in \mathcal{Y}_b(i)} s_b(i,j)$, where $\mathcal{Y}_b(i)$ is candidate antecedents of i.

The *entity coreference resolution* task involves identifying entity mentions that refer to the same real-world entity. The task aims to find an antecedent y_c for each span i using the scoring function s_c, which can be defined in a similar way to how s_b is in the bridging resolution task.

The model structure, which is shown in Figure 1, is described below.

Span Representation Layer Kobayashi et al. adopt the independent version of the entity-coreference resolver in Joshi et al. (2019) to the bridging resolution. An input document is divided into non-overlapping segments, each of which has a fixed size. These word sequences serve as input training sequences and are passed to SPANBERT to encode tokens and their contexts. Finally, span i's representation $\mathbf{g}_i$ is set as $[\mathbf{h}_{start(i)}; \mathbf{h}_{end(i)}; \mathbf{h}_{head(i)}; \phi_i]$, where $\mathbf{h}_{start(i)}$ and $\mathbf{h}_{end(i)}$ are the hidden vectors of the start and end tokens in the span, $\mathbf{h}_{head(i)}$ is an attention vector computed over the tokens of the span, and ϕ_i is a feature embedding that encodes a span width.

Bridging Prediction Layer Yu and Poesio compute the following pairwise score to predict a bridging link between span i and span j:

$$s_a(i,j) = \text{FFNN}_b([\mathbf{g}_i; \mathbf{g}_j; \mathbf{g}_i \circ \mathbf{g}_j; \psi_{ij}]) \quad (2)$$

where $\text{FFNN}_b(\cdot)$ is a feedforward neural network used in the bridging prediction layer, and ψ_{ij} encodes the segment distance between span i and span j. $\circ$ denotes element-wise multiplication, and $\mathbf{g}_i \circ \mathbf{g}_j$ encodes the similarity between two spans.

Coreference Prediction Layer The coreference prediction layer is defined analogously as the bridging prediction layer with another FFNN, FFNN_c. The first few layers of FFNN_b and FFNN_c are shared, as well as the span representations.

Kobayashi et al. propose a "hybrid" approach that incorporates Rösiger et al.'s (2018) rules into the MTL model by defining a rule score function $r(i,j)$, the value of which is the precision of each rule that posits a bridging link between two spans i, j. Then, the rule score is added into equation (1) as below:

$$s_{b'}(i,j) = \begin{cases} 0 & j = \epsilon \\ s_b(i,j) + \alpha r(i,j) & j \neq \epsilon \end{cases} \quad (3)$$

where α is a positive constant that determines the impact of the rule information on s'_b. The model uses $s'_b(i,j)$ to rank the bridging candidate antecedents of i.[2].

[2]Note that (1) the value of $r(i,j)$ is 0 if no rule predict a bridging link for i, j; (2) precisions of rules are computed on training set; and (3) α is tuned on the development set.

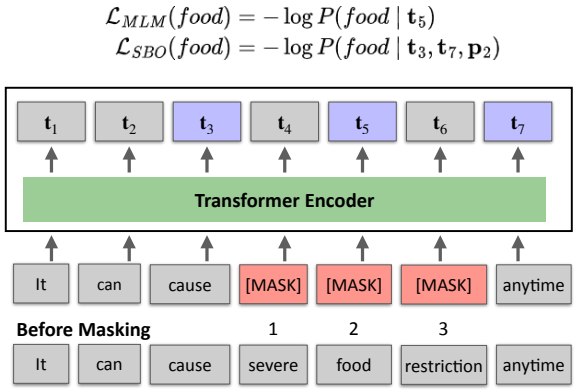

Figure 2: An illustration of the masking scheme and objectives in SPANBERT. Span masking masks all subword tokens in the span, "sever food restriction". Given a masked token "food", MLM makes predictions based on the contextualized vector $\mathbf{t}_5$, whereas SBO makes predictions based on the external boundaries tokens of the span $\mathbf{t}_3, \mathbf{t}_7$ as well as the position embedding $\mathbf{p}_2$ that indicates that "food" is the second token from $\mathbf{t}_3$.

The loss for each task is defined as the negative marginal log-likelihood of all correct bridging antecedents (or coreference antecedents). The bridging task loss and coreference task losses are combined using the weighted sum. The weight for each loss is tuned using grid search so that the average bridging resolution F-scores is maximized on the development set.

3.2 SpanBERT

The SPANBERT pre-trained model is an extension of BERT aimed at better learning of the representation of text *spans*.[3] Like BERT, SPANBERT takes as input a sequence of subword tokens $T = [t_1, ..., t_n]$ and produces a sequence of contextualized vector representations $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_n]$. In order to better learn span representations, SPANBERT employs two pre-training objectives:

Masked Language Modeling (MLM). In the original definition of the MLM task, given a sequence of tokens $T = [t_1, ..., t_n]$, a randomly chosen subset of tokens is replaced with a special $[MASK]$ token, and the goal is to predict for each masked token in a sequence the original token using $\mathbf{T}$. The MLM loss, $\mathcal{L}_{MLM}$, is the cross entropy loss. To better learn span representations, SPANBERT extends MLM so that it allows not only token masking but also *span* masking, which masks spans of tokens, and the goal is to predict each masked *span* using the surrounding context.

Span Boundary Objective (SBO) . Given a masked span consisting of contiguous tokens $(t_s, ..., t_e)$, SBO aims to predict the original token for each token in the masked span using the contextualized vectors of two tokens, namely the token to the left of the span boundary and the one to the right of its span boundary (i.e., $\mathbf{t}_{s-1}$ and $\mathbf{t}_{e+1}$), as well as the position embedding of the target token $\mathbf{p}_i$. The SBO loss, $\mathcal{L}_{SBO}$, is the cross-entropy loss.

Figure 2 illustrates how MLM and SBO work with an example.

4 PAIRSPANBERT

Next, we present PAIRSPANBERT, an extension of SPANBERT specialized for bridging resolution. To create PAIRSPANBERT, we use SPANBERT as a starting point and add a pre-training step to it that would enable the model to learn the contexts in which two mentions are implicitly linked to each other from data that is automatically generated either heuristically or via distant supervision with the help of a knowledge graph. To do so, we will describe how we obtain automatically generated data (Section 4.1), the pre-training task (Section 4.2), and the pre-training objective (Section 4.3).

4.1 Labeled Data Creation

We aim to collect automatically labeled data that would enable the model to learn the contexts in which two mentions are implicitly linked. As noted in the introduction, a pre-training task tends to be more effective for improving a target task (which in our case is bridging resolution) if the pre-training task is "closer" to the target task. Hence, we seek to collect automatically labeled data in which the two implicitly linked mentions are *likely* to have a bridging relation. We begin by (1) collecting noun pairs that are likely involved in a bridging relation in a *context-independent* manner, and then (2) using these pairs to automatically label sentences.

4.1.1 Collecting Noun Pairs

We obtain noun pairs that are likely to be involved in a bridging relation heuristically (via the syntactic structure of NPs) and via distance supervision (with the help of ConceptNet), as described below.

Syntactic Structure of NPs Following Hou (2018b), we extract noun pairs from the automatically parsed Gigaword corpus (Napoles et al., 2012)

[3] Although SPANBERT is often viewed as an extension of BERT, not everything in BERT appears in SPANBERT. For example, while BERT is pre-trained on the so-called next sentence prediction (NSP) task, SPANBERT is not.

by using the syntactic structures of NPs. Specifically, we first extract two NPs, X and Y, that are involved in the prepositional structure **X** *preposition* **Y** (e.g., "the door of the red house") or the possessive structure **Y** $'s$ **X** (e.g., "Japan's prime minister"), since Hou (2018b) has shown that these structures encode a variety of bridging relations between anaphors and their antecedents. Then, we create a noun pair from each extracted (**X**, **Y**) pair using the head noun of **X** and the head noun of **Y**. Note that the bridging relations captured in the resulting noun pairs, if any, are asymmetric. Typically, **X** corresponds to an anaphor while **Y** corresponds to its antecedent. For example, in "the door of the red house", the extracted **X** and **Y** would be "the door" and "the house", respectively.

ConceptNet Next, we show how to extract noun pairs that are likely involved in a bridging relation from ConcepNet. The exploitation of knowledge bases for bridging resolution so far has largely focused on deriving features from WordNet (e.g., computing the lexical distance between two mentions) (Poesio et al., 2004) and using these features to improve weak baselines (e.g., Pandit et al. (2020) incorporated knowledge-based features into an SVM model rather than a neural model). To our knowledge, we are the first to investigate if ConceptNet can be used to improve a state-of-the-art bridging resolver.

ConceptNet is a knowledge graph that connects phrases with labeled edges. It is built on various sources such as Open Mind Common Sense (Singh et al., 2002), Open Multilingual WordNet (Bond and Foster, 2013), and "Games with a purpose" (Von Ahn et al., 2006). There are 34 relations (i.e., edge labels) in ConceptNet 5.5. For example, *gearshift-car* has a PartOf relation label, meaning *gearshift* is a part of *car*. We obtain NP pairs in which two NPs are related through these ConceptNet relations, and for each NP pair (X,Y), we create a noun pair using the head noun of X and the head noun of Y.

Since not all ConceptNet relations are useful for bridging, we empirically identify the useful relations w.r.t. each evaluation dataset (e.g., ISNotes) as follows. First, for each ConceptNet relation type r, we apply the noun pairs extracted from r (see the previous paragraph) to the training portion of the dataset, positing a bridging link between two nouns in a training document if (1) their heads are related according to r and (2) they appear within two sentences of each other. Then, we compute a bridging resolution F-score w.r.t. r using the resulting bridging links. Finally, we sort the relation types in decreasing order of F-score and retain the top k relation types that collectively maximize the bridging resolution F-score on the training set. Only the noun pairs that are related through the selected relation types will be used to create automatically labeled data.

As an example, the set of ConceptNet relations that is determined to be useful for ARRAU RST is shown in Table 2.

4.2 Automatically Generating Labeled Data

The success of pre-training stems in part from learning from very large amounts of labeled data. Automatic generation of labeled data will enable us to easily generate a large amount of labeled (though noisy) data and allow the model to learn a variety of contexts in which two mentions are likely to have a bridging relation. In this subsection, we describe how we create automatically labeled instances, each of which is composed of one of the noun pairs collected in the previous subsection (through syntactic structures or ConcepNet) and the surrounding context.

For each document in parsed Gigaword, we automatically posit a bridging link between two nouns if two conditions are satisfied. First, they appear in one of the noun pairs collected in the previous subsection. Second, they are no more than two sentences apart from each other (because bridging links typically appear in a two-sentence window). There is a small caveat, however. Recall that the two nouns in a noun pair (**X**, **Y**) extracted from the syntactic structures play an asymmetric role, where **X** is an anaphor and **Y** its antecedent. So, when applying the first condition using the pairs collected from syntactic structures, the condition is satisfied only if **X** appears after **Y** in the associated document. For the noun pairs collected from ConceptNet, we do not have such a restriction since we did not mark which noun is the anaphor and which noun is the antecedent for each ConceptNet relation type.

4.3 Masking

Using the method described in the previous subsection, we will be to automatically annotate each Gigaword document with bridging links. Next, we will describe the two masking schemes we employ in PAIRSPANBERT, based on which we will define

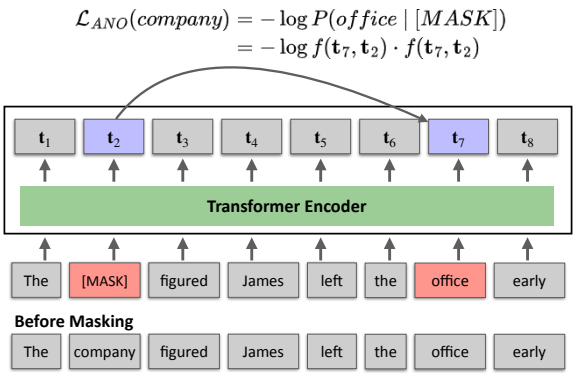

Figure 3: An illustration of anchor masking and ANO. $f(\cdot)$ corresponds to equation (5). Given a masked anchor "company", $\mathcal{L}_{ANO}$ calculates the probability that "office" is associated with "company", using the contextualized vectors of the start and end subword tokens of (masked) "company" and "office", $\mathbf{t}_2, \mathbf{t}_7$. In this example, neither words are divided into subwords, so the start and end tokens are the same.

pre-training tasks to predict the masked tokens in the next subsection.

PAIRSPANBERT assumes as input a 384-token segment (which in our case is taken from an automatically annotated Gigaword document). We define two masking schemes to mask the tokens in a given segment. First, we employ span masking, as described in the MLM task in Section 3.2 where randomly selected spans of tokens are replaced with the $[MASK]$ tokens. This masking strategy does not rely on the automatically identified bridging relations. Second, we define an *anchor masking* strategy, where we randomly choose the antecedents (i.e., anchors) in our automatically identified bridging relations and replace each (subword) token in each selected antecedent with the $[MASK]$ token.

We consider both masking schemes important for PAIRSPANBERT. As bridging resolution involves identifying relations between spans, span masking will ensure that the model learns good span representations. In contrast, anchor masking is designed to eventually enable the model to learn the contexts in which two nouns are likely involved in a bridging relation.

Following previous work Joshi et al. (2020), we mask at most 15% of the tokens in each input segment. In addition, we ensure that (1) among the masked tokens, p% will be masked using anchor masking, and the remaining ones will be masked using span ranking; and (2) the tokens masked by the two masked schemes do not overlap. Based on preliminary experiments on development data, we set p to 20.

4.4 Pre-Training Tasks

PAIRSPANBERT employs three pre-training tasks, MLM, SBO, and Associative Noun Objective (ANO). The MLM and SBO tasks are the same as those used in SPANBERT (see Section 3.2): we apply them to predict the tokens masked by both span ranking and anchor ranking.

ANO is a novel pre-training task we define specifically to enable the model to learn knowledge of bridging. Unlike MLM and SBO, which we apply to masked tokens produced by both masking schemes, ANO is applicable only to the masked tokens produced by anchor ranking. Specifically, given a sequence of input tokens $T = [t_1, ..., t_n]$ and a masked anchor anc consisting of subword tokens $(t_{s1}, ..., t_{e1})$, the goal of ANO is to predict an anaphor ana consisting of subword tokens $(t_{s2}, ..., t_{e2})$.[4] The probability that ana is associated with anc is defined using their boundary tokens (i.e., start and end tokens) as follows.

$$P(ana|anc) = P(t_{s2}|t_{s1}) \cdot P(t_{e2}|t_{e1}) \quad (4)$$

We calculate the probability of token t_i given token t_j in the sequence T using the contextualized vectors $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_n]$ produced by SPANBERT.

$$P(t_i|t_j) = \frac{\exp(s(\mathbf{t}_i, \mathbf{t}_j))}{\sum_{\mathbf{t}_k \in \mathbf{T}} \exp(s(\mathbf{t}_k, \mathbf{t}_j))} \quad (5)$$

where $s(\mathbf{t}_i, \mathbf{t}_j)$, the similarity between $\mathbf{t}_i$ and $\mathbf{t}_j$, is computed as $(\mathbf{w} \circ \mathbf{t}_i) \cdot \mathbf{t}_j$, $\mathbf{w}$ is a trainable vector of parameters, $\cdot$ is the dot product, and $\circ$ is element-wise multiplication. Figure 3 illustrates ANO and anchor masking with an example.

Given a set of masked anchors $anc \in A$ and anaphors associated with each anchor $ana \in C$, we define the loss $\mathcal{L}_{ANO}$ as follows.

$$\mathcal{L}_{ANO} = -\log \prod_{anc \in A} \sum_{ana \in C} P(ana|anc) \quad (6)$$

Finally, we compute the loss for PAIRSPANBERT $\mathcal{L}$ as the sum of the losses of its three pre-training objectives.

$$\mathcal{L} = \mathcal{L}_{MLM} + \mathcal{L}_{SBO} + \mathcal{L}_{ANO} \quad (7)$$

[4] Note that a given anchor may be associated with more than one anaphor.

Corpora	Docs	Tokens	Mentions	Anaphors
ISNotes	50	40,292	11,272	663
BASHI	50	57,709	18,561	459
ARRAU RST	413	228,901	72,013	3,777

Table 1: Statistics on different corpora.

Relation Type
RelatedTo, Synonym, UsedFor, HasA, IsA, AtLocation, capital, CapableOf, PartOf, InstanceOf

Table 2: ConceptNet relations used for ARRAU RST.

5 Evaluation

5.1 Experimental Setup

Corpora. For evaluation, we employ three commonly used corpora for bridging resolution, namely ISNotes, BASHI, and ARRAU RST. Table 1 shows statistics for these corpora. Because ISNotes and BASHI lack a standard train-test split, we perform five-fold cross validation on these corpora, using 70% of the documents for model training, 10% for development, and 20% for model evaluation. For ARRAU RST, we use the official train-test split.

Evaluation setting. We report results for bridging resolution in two settings, the end-to-end setting, where only raw, unannotated documents are given, and the gold mention setting, where gold mentions are given. In the end-to-end setting, we apply a mention detector to extract mentions.[5]

Evaluation metrics. Bridging anaphor recognition and resolution are reported in terms of precision, recall, and F-score. Recognition (Resolution) precision is the proportion of predicted anaphors that are correctly recognized (resolved). Recognition (Resolution) recall is the proportion of gold anaphors that are correctly recognized (resolved).

Baseline systems. We employ five baselines.

The first baseline is a state-of-the-art rule-based approach by Roesiger et al. (2018), denoted as Rules(R) in Table 3. For ISNotes and BASHI, We use Kobayashi et al.'s (2022b) publicly-available re-implementation of Rules(R). For ARRAU RST, no publicly-available implementation of Rules(R) that can be applied to automatically extracted mentions is available, so we re-implement Rules(R) for ARRAU RST for both the end-to-end and gold mention settings.

As a second baseline, we design a rule-based system based on the noun pairs extracted from the syntactic structures and ConceptNet. Specifically, we apply these noun pairs to the test set of each evaluation corpus as follows. If the two nouns in a pair appear within two sentences of each other in a test document, check whether the cosine similarity of their representations (obtained using Hou's

The remaining baselines are all SpanBERT-based. The third and fourth baselines are the state-of-the-art SpanBERT-based resolver and its hybrid version introduced in Section 3 (denoted as SBERT and SBERT(R) respectively in the table). The final baseline incorporates the similarity value computed by Rules(H) for each mention pair into SBERT(R), denoted as SBERT(R,H), as a set of 9 binary features. Specifically, each binary feature is associated with a threshold, and a binary feature fires if the similarity value is greater than the threshold associated with it. The 9 thresholds we use are: [-0.8, -0.6, -0.4, -0.2, 0.0, 0.2, 0.4, 0.6, 0.8].

Implementation details. To pre-train PairSpanBERT, we initialize it with the SpanBERT-large checkpoint and continue pre-training on the Gigaword documents automatically labeled with bridging links (see Section 4.1). Recall that these bridging links were created using noun pairs extracted from two sources: syntactic structures and ConceptNet. Rather than always use both sources to create bridging links, we use development data to determine whether we should use one of them (and if so, which one) or both of them. We optimize PairSpanBERT using Adam (Kingma and Ba, 2014) for 4k steps with a batch size of 2048 through gradient accumulation, a maximum learning rate of 1e-4, and a linear warmup of 400 steps followed by a linear decay of the learning rate. The remaining hyperparameters are the same as those in SpanBERT.

We fine-tune both SpanBERT and PairSpanBERT for up to 300 epochs with Adam (Kingma and Ba, 2014) in each dataset, with early stopping based on the development set. The version of SpanBERT we use is SpanBERT-large, so the PairSpanBERT we end up with is PairSpanBERT-large. The learning rates for SpanBERT and PairSpanBERT are searched in the range of {1e-5, 2e-5, 3e-5} while the task learning rates are searched in the range of {1e-4, 2e-4, 3e-4}. We split each document into segments of length 384. Each model considers up to the K closest preceding antecedent candidates. We search K out of {50, 80, 100, 120, 150}. We search the weight

[5] Following Kobayashi et al. (2022b), for ISNotes and ARRAU RST, we extract mentions using Hou's (2021) neural mention extractor; for BASHI, we extract mentions from syntactic parse trees produced by Stanford CoreNLP.

	Model	ISNotes						BASHI						ARRAU RST					
		Recognition			Resolution			Recognition			Resolution			Recognition			Resolution		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
	End-to-End Setting																		
1	Rules(R)	49.4	17.4	25.7	31.8	11.2	16.5	33.1	22.5	26.8	15.2	10.3	12.3	12.4	15.5	13.7	6.8	8.5	7.6
2	Rules(H)	9.2	21.1	12.8	3.4	7.8	4.7	3.5	15.1	5.7	1.0	4.3	1.6	6.6	14.5	9.0	1.6	3.6	2.2
3	SBERT	34.4	30.9	32.6	22.3	20.1	21.1	34.7	29.4	31.8	15.3	12.9	14.0	30.4	25.5	27.7	19.6	16.5	17.9
4	SBERT(R)	39.7	31.6	35.1	27.0	21.5	23.9	36.0	27.5	31.2	19.7	15.0	17.0	26.5	23.3	24.8	15.9	14.0	14.8
5	SBERT(R,H)	34.6	37.1	35.8	22.8	24.4	23.6	34.3	29.6	31.8	17.8	15.4	16.5	22.1	25.0	23.5	12.0	13.6	12.8
6	PSBERT	36.3	36.8	36.6	22.3	22.6	22.5	42.7	30.7	**35.7**	17.7	12.7	14.8	35.9	31.0	**33.2**	20.9	18.0	**19.4**
7	PSBERT(R)	40.2	39.5	**39.9**	26.4	25.9	**26.2**	43.8	27.0	33.4	24.6	15.1	**18.7**	28.5	23.8	26.0	17.5	14.6	15.9
	Gold Mention Setting																		
8	Rules(R)	52.7	19.2	28.1	34.0	12.4	18.1	35.8	23.6	28.5	17.8	11.7	14.1	18.0	31.5	22.9	12.1	21.1	15.3
9	Rules(H)	9.5	22.9	13.4	3.6	8.6	5.0	3.6	15.5	5.8	1.1	4.9	1.9	7.3	15.6	10.0	1.8	3.9	2.5
10	SBERT	37.1	33.1	35.0	24.5	21.9	23.1	35.0	29.7	32.1	16.1	13.7	14.8	31.9	26.9	29.2	21.2	17.9	19.4
11	SBERT(R)	43.8	34.6	38.6	30.4	24.1	26.8	37.6	28.8	32.6	21.6	16.6	18.7	30.4	28.4	29.4	20.8	19.4	20.1
12	SBERT(R,H)	37.6	39.8	38.7	25.6	27.2	26.4	34.9	30.3	32.4	19.2	16.7	17.9	25.8	30.1	27.8	16.6	19.4	17.9
13	PSBERT	38.7	38.8	38.7	24.9	24.9	24.9	43.7	30.3	**35.8**	19.3	13.4	15.8	35.3	32.5	**33.8**	21.6	19.9	20.7
14	PSBERT(R)	41.8	41.5	**41.6**	28.0	27.8	**27.9**	44.8	27.4	34.0	26.2	16.0	**19.9**	33.9	28.9	31.2	23.8	20.2	**21.9**

Table 3: Results of different resolvers in the end-to-end and gold settings. Each result is the average of five runs. The highest recognition and resolution F-scores for each dataset and each setting are boldfaced.

parameter for the rule score out of {50, 100, 150, 200}. Following previous work (Yu and Poesio, 2020), we downsample negative examples. The downsampling rate is searched out of {0.2, 0.4, 0.6, 0.8}. The rest of the parameters are set to be those reported in Kobayashi et al. (2022b).

5.2 Results and Discussion

End-to-end setting. The top half of Table 3 shows the end-to-end results. Consider first the baseline results. Two points deserve mention. First, in terms of F-score, SBERT(R,H) is significantly worse than SBERT(R) on all three datasets.[6] These results seem to suggest that the use of the automatically extracted noun pairs as additional features into SBERT(R) fails to improve its performance, probably because the noun pairs are too noisy to offer benefits when incorporated as features. Second, SBERT outperforms SBERT(R) on ARRAU RST. An inspection of the results reveals the reason: the rules designed by Rösiger et al. (2018) for ARRAU RST have low precision, thus adversely affecting the performance of SBERT(R) on ARRAU RST.

Second, the best resolution F-score is achieved by PSBERT(R) on ISNotes and BASHI and by PSBERT on ARRAU RST (again due to the low precision rules). PAIRSPANBERT significantly improves the best baseline in terms of resolution F-score by 2.3 points on ISNotes, 1.7 points on BASHI, and 1.5 points on ARRAU RST. PAIRSPANBERT's recognition F-scores are also generally higher than those of the SPANBERT-based resolvers. Although the noun pairs failed to improve SBERT when used as features, our results show that using these noun pairs to create automatically labeled data for pre-training is a better method to exploit such noisy information. Overall, we manage to achieve the best results to date on the three datasets using either PSBERT or PSBERT(R).

Gold mention setting. Results using the gold mention setting are shown in the bottom half of Table 3. The observations we made on the end-to-end results are more or less applicable to the gold mention results, except that PSBERT(R) manages to achieve the best resolution F-score on all three datasets. These are also the best resolution results obtained to date on these datasets for this setting.

5.3 Error Analysis

For a detailed analysis of the errors made by PAIRSPANBERT on the three datasets, we refer the reader to Appendix E.

6 Conclusion

We designed a novel pre-training task for bridging resolution using automatically annotated documents that contain NP pairs that are likely to be linked via implicit relations, and demonstrated that our newly pre-trained model, PAIRSPANBERT, effectively captures bridging relations. On three commonly-used datasets for bridging resolution, our new resolver, based on PAIRSPANBERT, outperforms the previous state-of-the-art model and other strong baselines on full bridging resolution in the end-to-end setting. All code and data will be made available upon publication of this paper.

[6] All significance test results are two-tailed paired t-tests, with $p < 0.05$ unless otherwise stated.

Limitations

Our pre-trained models targets a bridging resolution task, which could limit its application in other NLP tasks. There might be other pre-training objectives and knowledge sources such as wikidata that might be useful for bridging resolution, but we designed only one additional pre-training objective and used two knowledge sources. The pre-training was conducted using four A100 GPUs for one day, and the fine-tuning was done using a QUADRO RTX 6000 GPU for six hours.

Ethics Statement

About the pre-training dataset we created within this work, we do not expect to see any risk being posed by the user of this dataset nor any financial harm associated with its use. We will open-source the pre-trained model (PAIRSPANBERT) produced from this work immediately after publication. We plan to make it available on Hugging Face Model Hub.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362.

Herbert H Clark. 1975. Bridging. In *Theoretical issues in natural language processing*.

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.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022. Pre-training transformer models with sentence-level objectives for answer sentence selection.

Yufang Hou. 2018a. A deterministic algorithm for bridging anaphora resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1938–1948, Brussels, Belgium. Association for Computational Linguistics.

Yufang Hou. 2018b. Enhanced word representations for bridging anaphora resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 1–7, New Orleans, Louisiana. Association for Computational Linguistics.

Yufang Hou. 2020a. Bridging anaphora resolution as question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1428–1438, Online. Association for Computational Linguistics.

Yufang Hou. 2020b. Fine-grained information status classification using discourse context-aware BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6101–6112, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yufang Hou. 2021. End-to-end neural information status classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1377–1388, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yufang Hou, Katja Markert, and Michael Strube. 2013a. Cascading collective classification for bridging anaphora recognition using a rich linguistic feature set. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 814–820, Seattle, Washington, USA. Association for Computational Linguistics.

Yufang Hou, Katja Markert, and Michael Strube. 2013b. Global inference for bridging anaphora resolution. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 907–917, Atlanta, Georgia. Association for Computational Linguistics.

Yufang Hou, Katja Markert, and Michael Strube. 2014. A rule-based system for unrestricted bridging resolution: Recognizing bridging anaphora and finding links to antecedents. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2082–2093, Doha, Qatar. Association for Computational Linguistics.

Yufang Hou, Katja Markert, and Michael Strube. 2018. Unrestricted bridging resolution. *Computational Linguistics*, 44(2):237–284.

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

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

Sopan Khosla, Juntao Yu, Ramesh Manuvinakurike, Vincent Ng, Massimo Poesio, Michael Strube, and Carolyn Rosé. 2021. The CODI-CRAC 2021 shared task on anaphora, bridging, and discourse deixis in dialogue. In *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongjin Kim, Damrin Kim, and Harksoo Kim. 2021. The pipeline model for resolution of anaphoric reference and resolution of entity reference. In *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 43–47, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hideo Kobayashi, Yufang Hou, and Vincent Ng. 2022a. Constrained multi-task learning for bridging resolution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 759–770, Dublin, Ireland. Association for Computational Linguistics.

Hideo Kobayashi, Yufang Hou, and Vincent Ng. 2022b. End-to-end neural bridging resolution. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 766–778.

Hideo Kobayashi, Yufang Hou, and Vincent Ng. 2022c. End-to-end neural bridging resolution. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 766–778, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hideo Kobayashi, Shengjie Li, and Vincent Ng. 2021. Neural anaphora resolution in dialogue. In *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 16–31, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hideo Kobayashi and Vincent Ng. 2021. Bridging resolution: Making sense of the state of the art. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1652–1659, Online. Association for Computational Linguistics.

Bai Li, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2901–2908.

Katja Markert, Yufang Hou, and Michael Strube. 2012. Collective classification for fine-grained information status. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 795–804, Jeju Island, Korea. Association for Computational Linguistics.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Onkar Pandit, Pascal Denis, and Liva Ralaivola. 2020. Integrating knowledge graph embeddings to improve mention representation for bridging anaphora resolution. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 55–67, Barcelona, Spain (online). Association for Computational Linguistics.

Massimo Poesio and Ron Artstein. 2008. Anaphoric annotation in the ARRAU corpus. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 1170–1174.

Massimo Poesio, Rahul Mehta, Axel Maroudas, and Janet Hitzeman. 2004. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 143–150, Barcelona, Spain.

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.

Ellen F. Prince. 1981. Toward a taxonomy of given-new information. In P. Cole, editor, *Syntax and semantics: Vol. 14. Radical Pragmatics*, pages 223–255. Academic Press, New York.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3350–3363, Online. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. 2012. Learning the fine-grained information status of discourse entities. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 798–807, Avignon, France. Association for Computational Linguistics.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3066–3079, Online. Association for Computational Linguistics.

Ina Roesiger, Arndt Riester, and Jonas Kuhn. 2018. Bridging resolution: Task definition, corpus resources and rule-based experiments. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3516–3528, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ina Rösiger. 2018. BASHI: A corpus of Wall Street Journal articles annotated with bridging links. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ina Rösiger, Arndt Riester, and Jonas Kuhn. 2018. Bridging resolution: Task definition, corpus resources and rule-based experiments. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3516–3528, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Push Singh et al. 2002. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*.

Bo-Hsiang Tseng, Shruti Bhargava, Jiarui Lu, Joel Ruben Antony Moniz, Dhivya Piraviperumal, Lin Li, and Hong Yu. 2021. CREAD: Combined resolution of ellipses and anaphora in dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3390–3406, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Luis Von Ahn, Mihir Kedia, and Manuel Blum. 2006. Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Juntao Yu, Sopan Khosla, Ramesh Manuvinakurike, Lori Levin, Vincent Ng, Massimo Poesio, Michael Strube, and Carolyn Rosé. 2022. The CODI-CRAC 2022 shared task on anaphora, bridging, and discourse deixis in dialogue. In *Proceedings of the CODI-CRAC 2022 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 1–14, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Juntao Yu and Massimo Poesio. 2020. Multitask learning-based neural bridging reference resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3534–3546, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Model	Bridging	
	Recognition	Resolution
	ISNotes	
Rösiger et al. (2018)	25.6	17.5
Kobayashi et al. (2022b)	28.1	18.1
	BASHI	
Rösiger et al. (2018)	27.2	14
Kobayashi et al. (2022b)	28.5	14.1
	ARRAU RST	
Rösiger et al. (2018)	23.7	15.2
Our re-implementation	22.9	15.3

Table 4: Comparison of Rösiger et al's (2018) resolver and re-implementation. The table shows results of re-implementation provided in Kobayashi et al. (2022b) on ISNotes and BASHI, and our re-implementation on ARRAU RST.

A Re-implementation of Rules

Recall that we re-implement the rules designed by Rösiger et al. (2018) for ARRAU RST. These rules were designed to operate on gold mentions.

Table 4 shows the performances of our re-implementation of a rule-based system in the gold mention setting on ARRAU RST as well as the performances of re-implementation of rule-based system in Kobayashi et al. (2022b) on ISNotes and BASHI.

B Performance of End-to-end Rules in ARRAU RST

The end-to-end rules underperforms its gold version by 7.7% in resolution F-score in ARRAU RST. This performance mainly drop steps from mention extraction results and the quality of automatically calculated features used for rules in the end-to-end setting. While there are eleven rules for ARRAu RST, eight of them are designed specifically for ARRAu RST and not part of the rules for ISNotes and BASHI. The performances of these rules largely depend on gold annotation, particularly semantic category information.

We give examples for one of those eight rules, called the "Subset/Element-of" rule. In this rule, an anaphor must be modified by either an adjective, a noun, or a relative clause. Then, this rule searches for the closest candidate antecedent that has the same category and the same head as the anaphor in the last three sentences. In the following, we give an example of recall errors and precision errors made by the end-to-end rules but not by the gold mention setting version.

In the end-to-end setting, we automatically obtained the semantic category information using spaCy. However, gold categories in ARRAU RST have a different set of category labels. For example, ARRAu RST annotates "abstract" and "concrete" as part of categories to indicate if an entity refers to an abstract or a concrete object. These labels do not exist in spaCy. For example, in the gold mention setting, the "Subset/Element-of" rule correctly predicts "rents"-"Manhattan retail rents", and both mentions are annotated as "abstract" in gold annotations. On the other hand, there is no category label for these two mentions from the automatically obtained category information, so the rule does not predict this bridging pair. Compared to the gold mention setting, the end-to-end rule underperforms by 9.6% in recognition recall and by 7.1% in resolution recall.

One reason of the performance drop in precision is the wrong predicted mentions. For example, the end-to-end rule predicts "federal district court in Dallas"-"the Fifth U.S. Circuit Court" but "the Fifth U.S. Circuit Court" is not a gold mention. Compared to the gold mention setting, the end-to-end rule underperforms by 5.3% in recognition and by 4.1% in resolution precision.

	Gigaword	ConceptNet
Noun Pairs	9,776,957	1,804,399-1,872,782
Mapped Pairs	1,712,180,318	65,091,952-65,766,480

Table 5: Statistics for noun pairs extracted from each knowledge source. In ConceptNet, we have a different set of relations in each dataset, so we show the range of statistics from different datasets.

C Baseline Performances

Results of Rules(R) (Rösiger et al., 2018) for the gold setting in Table 3 are lower than the results shown in the original paper because 1) the original paper post-processes the system output with gold coreference information and 2) we evaluate the systems using the "harsh" setting described below. Note that Rösiger et al. (2018) do not have end-to-end results.

In the gold mention setting, we use the "harsh" evaluation setting used in some previous work (e.g., (Hou et al., 2018; Kobayashi et al., 2022b)). In ISNotes and BASHI, some bridging antecedents are *events* while events are *not* annotated as gold mentions. Previous studies handled these event antecedents differently. When reporting results on resolving gold mentions, some work (e.g., Hou et al. (2014), Hou et al. (2018)) chose not to include these event antecedents in the set of candidate antecedents while others (e.g., Rösiger et al. (2018), Yu and Poesio (2020)) did. However, the setting in which gold event antecedents are not included in training/evaluation is harsher because it implies that anaphors with event antecedents will always be resolved incorrectly. We believe that including gold event antecedents during evaluation does not represent a realistic setting, and will only report results using the "harsh" setting in this paper.

D Noun Pair Statistics

Table 5 shows statistics for 1) the number of noun pairs extracted from each knowledge source, and 2) number of possible mapping of these noun pairs to all documents in Gigaword when following the mapping method described in Section 4.2.

E Error Analysis

In this section, we analyze the errors made by PAIRSPANBERT.

Error analysis of the best end-to-end model. We perform an error analysis on our top-performing end-to-end model, *PSBERT(R)* for ISNotes and

BASHI and *PSBERT* for ARRAU RST, to gain a deeper understanding of its performance. Overall it seems that the system still struggles to recognize the majority of the bridging anaphors, with the recall scores ranging between 23.8% to 39.5% on the three testing datasets. Our analysis reveals that only a small percentage of recall errors in bridging anaphora recognition were due to mention prediction errors: 3%, 1.3%, and 2% of the gold bridging anaphors are misclassified as non-mentions in ISNotes, BASHI, and ARRAU RST, respectively. We find that the system constantly makes more recall errors at predicting definite bridging anaphors (i.e., NPs modified by the definite article "the") compared to other bridging anaphors across all datasets. For instance, on ISNotes, the recall scores of identifying definite bridging anaphors and other bridging anaphors are 31% and 45%, respectively.

Next we analyze the precision errors on ISNotes and ARRAU RST, as BASHI does not have mention annotations. Overall, we found that mention prediction errors (misclassifying non-mentions as bridging anaphors) account for 8.7% and 10.9% of the precision errors on ISNotes and ARRAU RST, respectively. On ISnotes, the majority of the precision errors were caused by classifying *new* and *old* mentions as bridging anaphors, accounting for 43% and 25% of the precision errors, respectively. On ARRAU RST, 71% of the precision errors were due to *new* mentions being classified as bridging anaphors. This observation on both datasets is in line with the previous research on bridging recognition (Hou et al., 2018), which suggests that systems often struggle to distinguish bridging anaphors from generic *new* mentions with simple syntactic structures.

Comparison of *PSBERT(R)* and *SBERT(R)* on ISNotes and BASHI. We further compare our best end-to-end system, *PSBERT(R)*, with the previous state-of-the-art model, *SBERT(R)*. On ISNotes, *PSBERT(R)* predicts 35% more bridging pairs than *SBERT(R)*, resulting in a higher recall for recognizing bridging anaphors (39.5% vs. 35.1%). Overall, *PSBERT(R)* outperformed *SBERT(R)* at predicting bridging pairs in which bridging anaphors are not modified by any determiners (*bare NPs*), such as "guests" or "walls". On BASHI, however, the trend is the opposite. *PSBERT(R)* predicts 18% less bridging pairs than *SBERT(R)* but achieves a higher precision score for bridging anaphora recognition (43.8% vs. 36.0%).

Comparison of *PSBERT* and *SBERT* on ARRAU RST. On ARRAU RST, we compare *PSBERT* with *SBERT* in the end-to-end setting. Both models predict a similar number of bridging pairs but *PSBERT* achives a higher precision score at recognizing bridging anaphors (35.9% vs. 30.4%). We observe that *PSBERT* outperforms *SBERT* at recognizing bridging anaphors that are *bare NPs*, especially proper names such as "*Seoul*" or "*Missouri*".