PAIRSPANBERT: An Enhanced Language Model for Bridging Resolution

Hideo Kobayashi[1], **Yufang Hou**[2] and **Vincent Ng**[1]
[1] Human Language Technology Research Institute, University of Texas at Dallas, USA
[2] IBM Research Europe, Ireland
{hideo,vince}@hlt.utdallas.edu
yhou@ie.ibm.com

Abstract

We present PAIRSPANBERT, a SPANBERT-based pre-trained model specialized for bridging resolution. PAIRSPANBERT is pre-trained with a novel 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 the entities within a text through non-identical semantic or encyclopedic relations (Clark, 1975; Prince, 1981). As demonstrated in Example 1, local coherence is established via the implicit link between the *bridging anaphor* (**prices**) and its *antecedent* (***meat, milk and grain***).

(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 the bridging anaphors in a text and linking them to their antecedents, is crucial for machine comprehension of 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. 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) show 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 natural language processing tasks, it is important to understand the reason. Recall that SPANBERT is an extension of BERT that is motivated by entity-based information extraction tasks such as entity coreference resolution 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 objectives. 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 with an objective that is in some sense related to the downstream task.

Motivated by this observation, we design a novel 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

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6931–6946
July 9-14, 2023 ©2023 Association for Computational Linguistics

SpanBERT in combination with its original objectives, yielding PairSpanBERT, a pre-trained model that is specialized for bridging resolution. Note that an important factor that contributes to the success of pre-training is the sheer amount of data on which the model is pre-trained: 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 lot of linguistic and commonsense knowledge. To enable our model to learn the 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 are evaluated in the rather unrealistic setting where gold mentions are 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 has modeled bridging anaphora recognition as a part of the information status (IS) classification problem where each discourse entity is assigned an IS category, with *bridging* being one of the categories (Rahman and Ng, 2011, 2012; Hou et al., 2013a; Hou, 2020b). 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). A comprehensive overview of these approaches can be found in Kobayashi and Ng (2020).

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 an approach of rephrasing bridging anaphors as questions and training question-answering models to directly extract antecedents from their previous contexts. Finally, there are a few works that propose models for full bridging resolution in the end-to-end setting (Kim et al., 2021; Kobayashi et al., 2021; Li et al., 2022) 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. (2022b) 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 schemes and the training objectives when 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 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 are reported in Kobayashi et al. (2022b), who extend Yu and Poesio's (2020) multi-task learning (MTL) approach to bridging resolution by (1) using SpanBERT to encode the input and (2) incorporating the predictions made by a rule-based resolver into the MTL framework. Since we aim to create PairSpanBERT, which specializes Span-

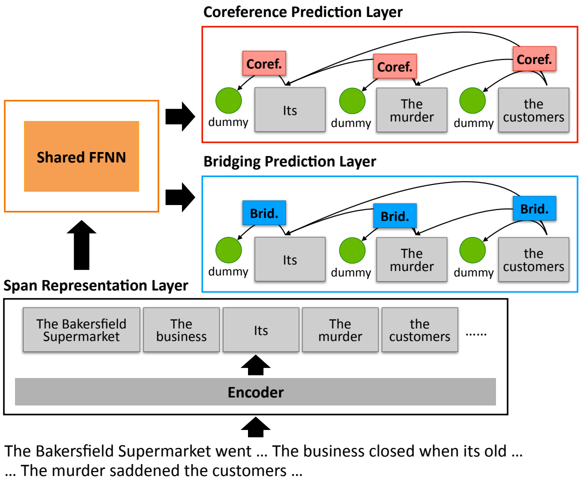
Coreference Prediction Layer
Coref.
Coref.
Coref.
dummy
Its
dummy
The murder
dummy
the customers
Shared FFNN
Bridging Prediction Layer
Brid.
Brid.
Brid.
dummy
Its
dummy
The murder
dummy
the customers
Span Representation Layer
The Bakersfield Supermarket
The business
Its
The murder
the customers
......
Encoder
The Bakersfield Supermarket went ... The business closed when its old ...
... The murder saddened the customers ...

Figure 1: The MTL framework for bridging resolution.

BERT for bridging resolution, and eventually replace SPANBERT with PAIRSPANBERT in the MTL framework, in this section we present Y&P's MTL framework (Section 3.1), Kobayashi et al.'s extensions to the framework (Section 3.2), and the inner workings of SPANBERT (Section 3.3).

3.1 The Multi-Task Learning Framework

Y&P's model takes as input a document D represented as a sequence of word tokens and the associated set of mentions (which can be gold mentions or automatically extracted mentions), and performs joint bridging resolution and coreference resolution, which we define below, in a MTL framework.

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 span i's antecedent, which can be a dummy antecedent ϵ (i.e., i is not anaphoric) or one of the preceding spans. Y&P define the following scoring function:

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

where $s_a(i,j)$ is a pairwise bridging score that 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 the set of candidate antecedents of i.

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

Figure 1 illustrates the structure of the MTL framework, which we describe in detail below.

Span Representation Layer To encode the tokens and the surrounding contexts of a gold mention, Y&P use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) that takes as input the BERT and GloVe embeddings. They define $\mathbf{g}_i$, the representation of span i, as $[\mathbf{x}_{start(i)}; \mathbf{x}_{end(i)}; \mathbf{x}_{head(i)}; \phi_i]$, where $\mathbf{x}_{start(i)}$ and $\mathbf{x}_{end(i)}$ are the hidden vectors of the start and end tokens of i, $\mathbf{x}_{head(i)}$ is an attention-based head vector and ϕ_i is a span width feature embedding.

Bridging Prediction Layer To predict bridging links, Y&P first calculate the pairwise score between spans i and j as follows:

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

where $\text{FFNN}_b(\cdot)$ represents a standard feedforward neural network, and $\circ$ denotes element-wise multiplication. This pairwise score includes $\mathbf{g}_i \circ \mathbf{g}_j$, which encodes the similarity of i and j, and ψ_{ij}, which denotes the distance between them.

Coreference Prediction Layer To predict coreference links, Y&P calculate a pairwise score between two spans that is defined analogously as in Equation 2 using another FFNN, FFNN_c. The model shares the first few hidden layers of FFNN_b and FFNN_c as well as the span representations.

3.2 Extensions to the MTL Framework

Kobayashi et al. (2022b) extend the MTL framework by replacing the LSTM encoder in Y&P with a SPANBERT encoder and proposing a *hybrid* approach to bridging resolution that augments the MTL model with the predictions made by Rösiger et al.'s (2018) rule-based bridging resolver. To implement the hybrid approach, they first define a rule score function $r(i,j)$ whose value is the precision of the rule that posits a bridging link between spans i and j, and then incorporate this rule score function into Equation 1 as follows:

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

where α is a positive constant that controls the impact of the rule information on s'_b. The model then uses $s'_b(i,j)$ to rank the candidate antecedents of span i. Note that (1) if no rule posits i and j as bridging, $r(i,j)$ is 0; (2) rule precision is computed on the training set; and (3) α is tuned on the development set.

The loss function is the weighted sum of the losses of the bridging task (L_b) and the coreference task (L_c). L_b and L_c are defined as the negative marginal log-likelihood of all correct bridging antecedents and coreference antecedents, respectively. The weights associated with the losses are tuned using grid search to maximize the average bridging resolution F-scores on development data.

3.3 SpanBERT

The SpanBERT pre-trained model is an extension of BERT aimed at better learning of the representations of text *spans*.[1] Like BERT, SpanBERT takes as input a sequence of subword tokens $T = [t_1, ..., t_n]$ and produces a sequence of contextualized vector representations $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_n]$. Unlike BERT, which randomly selects individual tokens for masking (where each token selected for masking is replaced with a special $[MASK]$ token), SpanBERT employs a *span* masking scheme where spans of tokens are masked in order to better learn span representations. SpanBERT employs two pre-training objectives:

Masked Language Modeling (MLM) Given a masked span consisting of contiguous tokens $(t_s, ..., t_e)$, the model is asked to predict for each masked token t_i in the span the original token using $\mathbf{t}_i$. The MLM loss, $\mathcal{L}_{MLM}$, is the cross entropy loss.

Span Boundary Objective (SBO) Given a masked span consisting of contiguous tokens $(t_s, ..., t_e)$, the model is asked to predict for each token t_i in the masked span the original token 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 via 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 masking scheme (Section 4.2), and the pre-training task (Section 4.3).

$$\mathcal{L}_{MLM}(\mathit{food}) = -\log P(\mathit{food} \mid \mathbf{t}_5)$$
$$\mathcal{L}_{SBO}(\mathit{food}) = -\log P(\mathit{food} \mid \mathbf{t}_3, \mathbf{t}_7, \mathbf{p}_2)$$

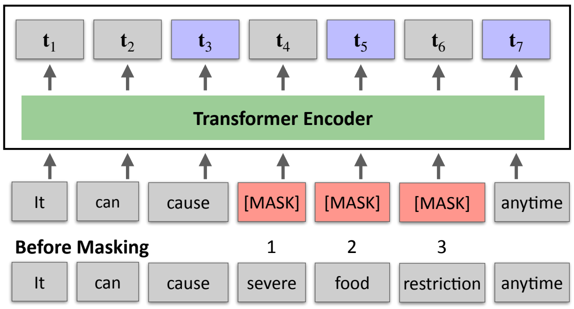

Figure 2: An illustration of the masking scheme and the objectives in SpanBERT. Span masking masks all the subword tokens in the span "severe food restriction". Given the masked token "food", MLM makes predictions based on the contextualized vector $\mathbf{t}_5$, whereas SBO makes predictions based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

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 resembles 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 structures of noun phrases (NPs)) and via distance supervision (with ConceptNet), as described below.

Syntactic Structures of NPs Following Hou (2018b), we extract noun pairs from the automatically parsed Gigaword corpus (Napoles et al., 2012) by using the syntactic structures of NPs. Specifi-

[1] 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.

cally, 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. 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 ConceptNet (Speer et al., 2017). The exploitation of knowledge bases for bridging resolution 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) incorporate knowledge-based features into an SVM model rather than a neural model).

ConceptNet is a knowledge graph that connects phrases with labeled edges. It is built on various sources such as Open Mind Common Sense (Singh, 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 part of a *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 resolution, 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.

The ConceptNet relation types selected for the three datasets (ISNotes, BASHI, ARRAU RST) can be found in Appendix A. The relation types that are used in all three datasets include RELATEDTO, SYNONYM, HASA, ISA, ATLOCATION, CAPABLEOF, and PARTOF. Intuitively, all of these relation types are closely related to bridging.

4.1.2 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 (noisily) labeled 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 ConceptNet) 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 (this is motivated by the observation that 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** is its antecedent. So, when applying the first condition to the pairs collected from the syntactic structures, we consider the condition 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 do not mark which noun is the anaphor and which noun is the antecedent for each ConceptNet relation type.

4.2 Masking

Using the method described in the previous subsection, we will be able to automatically annotate each Gigaword document with bridging links. Next, we describe the two masking schemes we employ in PAIRSPANBERT, based on which we will define the pre-training tasks to predict the masked tokens in the next subsection.

PAIRSPANBERT assumes as input a segment of up to 512 tokens (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 SBO task in Section 3.3 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 masking; and (2) the tokens masked by the two masked schemes do not overlap. Based on experiments on development data, we set p to 20.

4.3 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 3.3): we apply them to predict the tokens masked by both span masking and anchor masking.

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 the masked tokens produced by both masking schemes, ANO is applicable only to the masked tokens produced by anchor masking. 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 $(t_{s2}, ..., t_{e2})$.[2] The probability that ana is associated with anc is defined using their boundary tokens (i.e., start and end tokens) as follows.

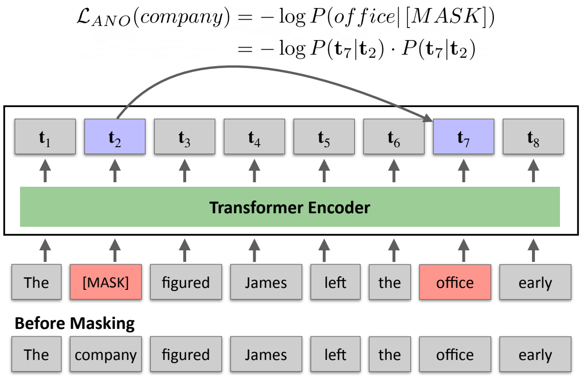

Figure 3: An illustration of anchor masking and ANO. Given the masked anchor "company", $\mathcal{L}_{ANO}$ calculates the probability that "office" is associated with "company" using the contextualized vectors of the start and end subword tokens of (masked) "company" and "office", $\mathbf{t}_2$ and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

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

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 of $\mathbf{t}_i$ 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.

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

5 Evaluation

5.1 Experimental Setup

Corpora. For evaluation, we employ three commonly used corpora for bridging resolution, namely

[2] An 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.

ISNotes, BASHI, and ARRAU RST. Table 1 shows statistics on 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 settings. We report results for bridging resolution in the *end-to-end* setting, where only raw 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.[3] In the gold mention setting, we employ the *harsh* evaluation method (see Appendix B).

Evaluation metrics. Bridging anaphor recognition and resolution results are reported in 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 Rösiger et al. (2018), denoted as Rules(R) in Table 2. For ISNotes and BASHI, we use Kobayashi et al.'s (2022b) 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.[4]

As our second baseline, we design a heuristic system based on the noun pairs extracted from the syntactic structures and ConceptNet[5], denoted as Rules(H). 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, we check whether the cosine similarity of their representations (obtained using Hou's (2018a) word embedding algorithm) exceeds a certain threshold.[6] If so, we posit a bridging link between them. If the anaphor is being linked to more than one antecedent, we pick the antecedent that has the highest cosine similarity with it. Note that we use the noun pairs collected from both the syntactic structures and ConceptNet.

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 3.2 (denoted as SBERT and SBERT(R) respectively in Table 2). 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 are –0.8, –0.6, –0.4, –0.2, 0.0, 0.2, 0.4, 0.6, and 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. Recall that these links are created using the noun pairs extracted from two sources: syntactic structures and ConceptNet. Rather than always use both sources to create bridging links, we use dev data to determine whether we should use one (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 parameters are the same as those in SpanBERT. Pre-training is performed on a machine with four A100 GPUs and lasts for a day.

We fine-tune both SpanBERT and PairSpanBERT for up to 400 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. The learning rates for SpanBERT and PairSpanBERT are searched out of {1e-5, 2e-5, 3e-5}, while the task learning rates are searched out of {1e-4, 2e-4, 3e-4, 4e-4}. We split each document into segments of length 384. Each model consid-

[3] 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 (Manning et al., 2014)

[4] See Appendix C for the re-implementation details.

[5] See Appendix D for statistics on the noun pairs extracted from the syntactic structures and ConceptNet.

[6] We set the threshold to 0.2 in all three datasets after tuning on each development set in the range of {0.0, 0.1, 0.2, 0.4}.

ers up to the K closest preceding candidate antecedents. We search K out of $\{50, 80, 100, 120, 150\}$. We search the weight parameter for the rule score out of $\{50, 100, 150, 200\}$. Following Yu and Poesio (2020), we downsample negative examples. The downsampling rate is searched out of $\{0.2, 0.4, 0.6, 0.8\}$. The remaining parameter values are the same as those reported in Kobayashi et al. (2022b). Fine-tuning is performed on a QUADRO RTX 6000 GPU machine and lasts for six hours.

5.2 Results and Discussion

End-to-end setting. The top half of each subtable in Table 2 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 considerably worse than SBERT(R) on all three datasets. These results suggest that using automatically extracted noun pairs as additional features for 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.

The best resolution F-score is achieved by PSBERT(R), which is created by replacing SPANBERT with PAIRSPANBERT in SBERT(R), on ISNotes and BASHI and by PSBERT, which is created by replacing SPANBERT with PAIRSPANBERT in SBERT, on ARRAU RST. PAIRSPANBERT considerably improves the best baseline in resolution F-score by 2.3 points on ISNotes, 1.3 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 fail 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 for the gold mention setting are shown in the bottom half of each subtable in Table 2.[7] Our observations on the end-to-end results are more or less applicable to the gold mention results, except that PSBERT(R) man-

(a) ISNotes

	Model	Recognition			Resolution		
		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
2	Rules(H)	9.2	21.1	12.8	3.4	7.8	4.7
3	SBERT	34.4	30.9	32.6	22.3	20.1	21.1
4	SBERT(R)	39.7	31.6	35.1	27.0	21.5	23.9
5	SBERT(R,H)	34.6	37.1	35.8	22.8	24.4	23.6
6	PSBERT	36.3	36.8	36.6	22.3	22.6	22.5
7	PSBERT(R)	40.2	39.5	**39.9**	26.4	25.9	**26.2**
	Gold Mention Setting						
8	Rules(R)	52.7	19.2	28.1	34.0	12.4	18.1
9	Rules(H)	9.5	22.9	13.4	3.6	8.6	5.0
10	SBERT	37.1	33.1	35.0	24.5	21.9	23.1
11	SBERT(R)	43.8	34.6	38.6	30.4	24.1	26.8
12	SBERT(R,H)	37.6	39.8	38.7	25.6	27.2	26.4
13	PSBERT	38.7	38.8	38.7	24.9	24.9	24.9
14	PSBERT(R)	41.8	41.5	**41.6**	28.0	27.8	**27.9**

(b) BASHI

	Model	Recognition			Resolution		
		P	R	F	P	R	F
	End-to-End Setting						
1	Rules(R)	33.1	22.5	26.8	15.2	10.3	12.3
2	Rules(H)	3.5	15.1	5.7	1.0	4.3	1.6
3	SBERT	34.7	29.4	31.8	15.3	12.9	14.0
4	SBERT(R)	36.0	27.5	31.2	19.7	15.0	17.0
5	SBERT(R,H)	34.3	29.6	31.8	17.8	15.4	16.5
6	PSBERT	41.5	29.1	**34.2**	17.7	12.7	14.8
7	PSBERT(R)	43.0	25.6	32.1	25.4	14.3	**18.3**
	Gold Mention Setting						
8	Rules(R)	35.8	23.6	28.5	17.8	11.7	14.1
9	Rules(H)	3.6	15.5	5.8	1.1	4.9	1.9
10	SBERT	35.0	29.7	32.1	16.1	13.7	14.8
11	SBERT(R)	37.6	28.8	32.6	21.6	16.6	18.7
12	SBERT(R,H)	34.9	30.3	32.4	19.2	16.7	17.9
13	PSBERT	43.7	30.3	**35.8**	19.3	13.4	15.8
14	PSBERT(R)	44.5	27.0	33.6	27.3	15.3	**19.6**

(c) ARRAU AST

	Model	Recognition			Resolution		
		P	R	F	P	R	F
	End-to-End Setting						
1	Rules(R)	12.4	15.5	13.7	6.8	8.5	7.6
2	Rules(H)	6.6	14.5	9.0	1.6	3.6	2.2
3	SBERT	29.7	24.9	27.1	19.0	15.9	17.3
4	SBERT(R)	25.9	22.7	24.2	15.1	13.4	14.2
5	SBERT(R,H)	21.6	24.4	22.9	11.5	13.0	12.2
6	PSBERT	31.1	26.5	**28.6**	21.2	16.9	**18.8**
7	PSBERT(R)	28.1	23.2	25.4	16.7	14.1	15.3
	Gold Mention Setting						
8	Rules(R)	18.0	31.5	22.9	12.1	21.1	15.3
9	Rules(H)	7.3	15.6	10.0	1.8	3.9	2.5
10	SBERT	31.3	26.3	28.6	20.6	17.3	18.8
11	SBERT(R)	29.9	27.8	28.8	20.3	18.8	19.5
12	SBERT(R,H)	25.2	29.5	27.2	16.0	18.8	17.3
13	PSBERT	32.7	30.0	**31.3**	22.6	18.1	20.1
14	PSBERT(R)	32.9	27.6	30.0	22.9	18.9	**20.7**

Table 2: Results of different resolvers (averaged over two runs). The highest recognition and resolution F-scores for each dataset and each setting are boldfaced.

[7] See Appendix E for a discussion of the Rules(R) results.

ages to achieve the best resolution F-score on all three datasets. These are the best resolution results obtained to date on these datasets for this setting.

We conclude this subsection with two points that we believe deserve mention. First, all the PAIRSPANBERT results reported in Table 2 are obtained using the version of the model that is trained on noun pairs from both the syntactic structures and ConceptNet, as using the pairs from both sources always yields better resolution F-scores on the dev set than using the pairs from either source. Second, in order to confirm that PAIRSPANBERT's superiority over SPANBERT is indeed attributable to the addition of ANO rather than the additional pre-training steps it receives, we further pre-train SPANBERT using MLM and SBO for as many epochs as we pre-train PAIRSPANBERT and show that SPANBERT's performance changes after further pre-training are negligible (see Appendix F).

5.3 Analysis of Results

Error analysis of the best end-to-end models. We conduct an error analysis of our top-performing end-to-end models, PSBERT(R) for ISNotes and BASHI and PSBERT for ARRAU RST, to gain additional insights into them. Overall, it appears that these models struggle to recognize the majority of the bridging anaphors, with the recall scores ranging between 25.6% and 39.5% on the three datasets. In addition, only a small percentage of the recall errors in bridging anaphora recognition are 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. These models consistently make more recall errors at identifying definite bridging anaphors (i.e., NPs modified by the definite article "the") than 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. Mention prediction errors (i.e., 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 are caused by misclassifying *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 are due to *new* mentions being misclassified as bridging anaphors. These findings corroborate the results reported in previous research on bridging recognition (Hou et al., 2018), which suggest that models 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 resolver, PSBERT(R), with the previous state-of-the-art resolver, 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. 31.6%). Overall, PSBERT(R) is better than SBERT(R) at predicting bridging pairs in which the bridging anaphors are not modified by any determiners (i.e., *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 (43.0% 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 achieves a higher precision for bridging anaphor recognition (31.1% vs. 29.7%). We observe that PSBERT is better than SBERT at recognizing bridging anaphors that are *bare NPs*, especially proper names such as "*Seoul*".

6 Conclusion

We designed a novel pre-training task for bridging resolution using automatically annotated documents that contain noun pairs that are likely to be linked via implicit relations, and demonstrated that our newly pre-trained model, PAIRSPANBERT[8], effectively captures bridging relations. On three commonly-used datasets for bridging resolution, our new resolver based on PAIRSPANBERT outperformed the previous state-of-the-art models and other strong baselines for full bridging resolution.

In future work, we plan to apply PAIRSPANBERT to other language processing tasks, particularly relation extraction tasks, since the noun pairs extracted from the syntactic structures and ConceptNet are likely to have non-identical relations.

[8] The model checkpoint can be downloaded from `https://huggingface.co/utd/pairspanbert`.

Acknowledgments

We thank the three anonymous reviewers for their insightful comments on an earlier draft of the paper. This work was supported in part by NSF Grants IIS-1528037 and CCF-1848608. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the NSF.

Limitations

There are at least two limitations. First, PAIRSPANBERT is specialized for the bridging resolution task, which could limit its applicability to other downstream tasks. Second, there are other pre-training objectives and knowledge sources that may be useful for bridging resolution (e.g., Wikidata), but we have designed only one pre-training objective and employed only two knowledge sources.

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, Sofia, Bulgaria. Association for Computational Linguistics.

Herbert H. Clark. 1975. Bridging. In *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*, TINLAP '75, page 169–174, USA. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2205.10455*.

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

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

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.

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, 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. 2020. Bridging resolution: A survey of the state of the art. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3708–3721, Barcelona, Spain (Online). International Committee on 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.

Shengjie Li, Hideo Kobayashi, and Vincent Ng. 2022. Neural anaphora resolution in dialogue revisited. In *Proceedings of the CODI-CRAC 2022 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 32–47, Gyeongju, Republic of Korea. 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.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

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.

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. 2011. Learning the information status of noun phrases in spoken dialogues. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1069–1080, Edinburgh, Scotland, UK. 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 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. 2002. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. AAAI Press, Palo Alto, California USA.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 4444–4451, San Francisco, California. AAAI Press.

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.

A ConceptNet Relation Types

Table 3 shows the list of ConceptNet relation types selected for each of the three evaluation datasets based on their respective *training* data. Recall that we conduct five-fold cross-validation experiments on ISNotes and BASHI owing to the lack of an official train-test split. As a result, for ISNotes and BASHI, we end up with five sets of ConceptNet relation types, one from each of the five train-test splits. Rather than showing all five sets, we show in the table both the *union* and the *intersection* of the five sets of relation types for ISNotes and BASHI.

B Harsh Evaluation Method

When evaluating the resolvers in the gold mention setting, we use the "harsh" evaluation method that is also employed in some previous work (e.g., Hou et al. (2018), Kobayashi et al. (2022b)). More specifically, in ISNotes and BASHI, some bridging anaphors have clausal antecedents that correspond

Dataset		Relation Types
ISNotes	Union	RelatedTo, Synonym, UsedFor, HasA, IsA, AtLocation, CapableOf, PartOf, InstanceOf, HasContext, FormOf, DerivedFrom
	Intersection	RelatedTo, Synonym, UsedFor, HasA, IsA, AtLocation, CapableOf, PartOf
BASHI	Union	RelatedTo, Synonym, UsedFor, HasA, IsA, AtLocation, CapableOf, PartOf, InstanceOf, HasContext, HasFirstSubevent, HasPrerequisite, DistinctFrom
	Intersection	RelatedTo, Synonym, HasA, IsA, AtLocation, CapableOf, PartOf, InstanceOf
ARRAU RST		RelatedTo, Synonym, UsedFor, HasA, IsA, AtLocation, capital, CapableOf, PartOf, InstanceOf

Table 3: ConceptNet relation types selected for each evaluation dataset.

to *events*. While clausal antecedents are annotated, they are not annotated as gold mentions, and previous studies differ in terms of how they should be handled. Some previous work (e.g., Hou et al. (2014), Hou et al. (2018)) chose not to include these clausal antecedents in the list of candidate antecedents while others (e.g., Rösiger et al. (2018), Yu and Poesio (2020)) did. Obviously, the setting in which gold clausal antecedents are not included in training/evaluation is harsher because it implies that anaphors with clausal antecedents will always be resolved incorrectly. We believe that including gold clausal antecedents during evaluation does not represent a realistic setting, and therefore only report results using the "harsh" setting when evaluating on gold mentions in this paper.

C Re-Implementation of Rules(R) for ARRAU AST

Recall that our first baseline, Rules(R), is Rösiger et al.'s (2018) rule-based resolver. As mentioned in Section 5.1, for ARRAU RST, no publicly-available implementation of Rules(R) that can be applied to automatically extracted mentions is available. Consequently, we re-implement Rösiger et al.'s (2018) resolver, which was designed to operate on gold mentions, and extend it so that it can operate on automatically extracted mentions. The extension, which is motivated by Kobayashi et al. (2022b), is fairly straightforward. While Rösiger et al. use gold annotations (i.e., gold POS tags, gold parse trees, and gold entity types) when computing the information needed by the rules, we use Stanford CoreNLP (Manning et al., 2014) to provide automatic constituency and dependency parse trees and spaCy (Honnibal and Montani, 2017) to provide automatic part-of-speech tags and entity types. We apply the resulting rules to the mentions extracted by Hou's (2021) neural mention extractor.

Model	Bridging	
	Recognition	Resolution
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 our re-implementation on ARRAU AST.

The results in Table 4 show that our re-implementation of Rules(R) is comparable to Rösiger et al.'s (2018) implementation in recognition and resolution F-scores when applied to gold mentions. Note that since Rösiger et al. do not report end-to-end results, we are unable to compare the two resolvers in the end-to-end setting.

When applying our re-implmentation to automatically extracted mentions, we find that resolution F-score drops by 7.7%. This performance drop stems primarily from mention extraction errors and imperfect feature computations. Below we provide examples of recall errors and precision errors resulting from the application of our rules to automatically extracted mentions.

A category of recall errors arises from imperfect computation of semantic category information. As mentioned above, when applied to automatically extracted mentions, the rules rely on the semantic category information automatically obtained using spaCy. However, when applied to gold mentions, the rules rely on the gold semantic categories defined in ARRAU RST, which are different from those provided by spaCy. For example, "abstract" and "concrete" are two semantic categories defined in ARRAU RST that indicate whether an entity refers to an abstract object or a concrete object, but neither of these category labels exist in spaCy. Consequently, when applied to gold mentions, the "Subset/Element-of" rule, which resolves an anaphor modified by an adjective, a noun, or a relative clause to the closest candidate antecedent in the preceding three sentences if the two mentions have the same semantic category and the same head, correctly identifies the bridging

Noun Pairs	Bridging Links
Syntactic Structures	
9,776,957	1,712,180,318
ConceptNet	
1,804,399–1,872,782	65,091,952–65,766,480

Table 5: Statistics on (1) the number of noun pairs extracted from the syntactic structures and ConceptNet and (2) the number of bridging links obtained by applying the resulting noun pairs to the Gigaword documents.

link between "rents" and "Manhattan retail rents", as both mentions possess the gold semantic category "abstract". On the other hand, no category labels are provided by spaCy for these two mentions, so the rule does not posit these two mentions as having a bridging relation when it is applied to automatically extracted mentions. The rules in the end-to-end setting underperform their counterparts in the gold mention setting by 9.6% in recognition recall and by 7.1% in resolution recall.

A category of precision errors arises from erroneously identified mentions. For example, an end-to-end rule (wrongly) posits "federal district court in Dallas" and "the Fifth U.S. Circuit Court" as having a bridging relation, but "the Fifth U.S. Circuit Court" is not a gold mention. The rules in the end-to-end setting underperform their counterparts in the gold mention setting by 5.3% in recognition precision and by 4.1% in resolution precision.

D Statistics on Noun Pairs

Recall from Section 4.1.1 that we collect noun pairs from both the syntactic structures and ConceptNet, which are subsequently applied to the Gigaword documents to automatically annotate them with bridging relations (Section 4.1.2). Table 5 shows the statistics on (1) the number of noun pairs that can be extracted from each of the two knowledge sources and (2) the number of bridging links that we obtain when applying the resulting noun pairs to the Gigaword documents. Since the ConceptNet relations we use to extract noun pairs from different datasets are not the same, the number of bridging links we can establish will depend on which set of relations we use. Hence, only the ranges are shown for ConceptNet in the table.

E Results of Rules(R) for the Gold Mention Setting

It is worth mentioning that the results of Rules(R) for the gold mention setting in Table 2 are lower than the corresponding results in Rösiger et al.'s (2018) paper. We attribute the performance differences to two reasons. First, we evaluate Rules(R) using the harsh evaluation method. Second, Rösiger et al. post-process their resolver's output with *gold* coreference information.

Model	ISNotes		BASHI		ARRAU	
	Rec.	Res.	Rec.	Res.	Rec.	Res.
End-to-End Setting						
SBERT(R)	35.1	23.9	31.2	17.0	24.8	14.8
CSBERT(R)	34.4	23.6	30.8	16.7	24.0	14.9
Gold Mention Setting						
SBERT(R)	38.6	26.8	32.6	18.7	29.4	20.1
CSBERT(R)	37.4	26.9	31.9	18.5	30.0	20.3

Table 6: Comparison of SpanBERT (SBERT) and SpanBERT with additional pre-training (CSBERT) in the end-to-end and gold mention settings. Each result is the average of two runs.

F Continued Pre-training of SpanBERT

One may argue that the comparison between PairSpanBERT and SpanBERT in our experiments is not entirely fair. Specifically, PairSpanBERT may have an unfair advantage over SpanBERT because it is pre-trained for more epochs than SpanBERT. To investigate whether the performance improvement of PairSpanBERT stems from the additional pre-training steps, we conduct an experiment to determine if SBERT(R) can be improved with additional pre-training. Specifically, we additionally pre-train SBERT(R) using MLM and SBO on the same dataset as PairSpanBERT for as many epochs as we pre-train PairSpanBERT.

Table 6 shows the SBERT(R) results on anaphor recognition and resolution (expressed in terms of F-score) before and after the additional pre-training steps. In the end-to-end setting, additionally pre-training SBERT(R) causes resolution F-score to change by –0.3–0.1 points. In the gold mention setting, the corresponding changes in resolution F-score are –0.2–0.2 points. Given that these changes are negligible, we conclude that PairSpanBERT's superior performance can be attributed to the addition of ANO rather than the additional pre-training steps.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations Section

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement Section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

4.1.1

☑ B1. Did you cite the creators of artifacts you used?
4.1.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
5

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not clearly explained in the paper, but our paper's use is consistent with their intended use.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The datasets used in the paper do not include offensive content.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
5

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Yes. Section 5

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.