The KITMUS Test: Evaluating Knowledge Integration from Multiple Sources

Akshatha Arodi[1,2]*, **Martin Pömsl**[1,2]*, **Kaheer Suleman**[3],
Adam Trischler[3], **Alexandra Olteanu**[3], **Jackie Chi Kit Cheung**[1,2]
[1] McGill University [2] Mila Quebec AI Institute [3] Microsoft Research
{akshatha.arodi@mail, martin.pomsl@mail, jcheung@cs}.mcgill.ca
{kaheer.s}@gmail.com {adam.trischler, alexandra.olteanu}@microsoft.com

Abstract

Many state-of-the-art natural language understanding (NLU) models are based on pretrained neural language models. These models often make inferences using information from multiple sources. An important class of such inferences are those that require both background knowledge, presumably contained in a model's pretrained parameters, and instance-specific information that is supplied at inference time. However, the integration and reasoning abilities of NLU models in the presence of multiple knowledge sources have been largely understudied. In this work, we propose a test suite of coreference resolution subtasks that require reasoning over multiple facts. These subtasks differ in terms of which knowledge sources contain the relevant facts. We also introduce subtasks where knowledge is present only at inference time using fictional knowledge. We evaluate state-of-the-art coreference resolution models on our dataset. Our results indicate that several models struggle to reason on-the-fly over knowledge observed both at pretrain time and at inference time. However, with task-specific training, a subset of models demonstrates the ability to integrate certain knowledge types from multiple sources. Still, even the best performing models seem to have difficulties with reliably integrating knowledge presented only at inference time.

1 Introduction

Progress on natural language understanding (NLU) benchmarks has recently been driven by pretrained large language models (LLMs), which can be adapted to specific tasks via finetuning (Peters et al., 2018; Devlin et al., 2019; Le Scao and Rush, 2021). These models draw on a variety of knowledge sources, such as knowledge given in inputs at inference time and knowledge contained in their parameters, usually acquired via pretraining.

1. **Servin** is a judge. **Kea** is a baker. Servin and Kea met at a park. After a long day at work deciding cases in a law court, **he** was happy to relax. [Answer: **Servin**]

2. Schwing is a gladiower. The work of a gladiower is inwaging ledmonly. The work of a popesmer is chodoling larely. Bate is a popesmer. At the coffee shop, **Schwing** and **Bate** connected. The coffee was excellent. **She** shared experiences from a career of chodoling larely. [Answer: **Bate**]

Figure 1: Examples from KITMUS showing coreference cases that require real (1.) and fictional (2.) knowledge. To resolve the pronoun (in red), a model needs to draw on entity-specific knowledge about an entity's occupation as well as on background knowledge about what kind of work the occupation entails.

Recent work suggests that models can use pretrain-time knowledge in tasks like translation and question answering to obtain performance gains (Brown et al., 2020; Roberts et al., 2020). However, natural language understanding often requires knowledge that is only supplied at inference time because of, e.g., time sensitivity or instance specificity. Consider the passage "John saw the newly elected president on TV." Pretrained parameters can conceivably contain information about what presidents do and what a TV is, but they cannot contain reliable knowledge about who John is (since "John" is an instance-specific identifier) or who the president is (because the president might have changed since pretraining). It follows that successful models for knowledge-intensive NLU tasks might require the ability to use both pretrain-time and inference-time knowledge.

To effectively use these two knowledge sources, models must (1) retrieve relevant information from each knowledge source, (2) adjudicate between potentially conflicting information, and (3) integrate multiple units of information from both knowledge sources and reason over them on the fly. For example, pretrained parameters might contain the knowledge that Donald Trump is the president of

* Equal contribution

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 15088–15108
July 9-14, 2023

the United States, but inference-time inputs might state that Joe Biden is the president. Based on the contextual information available in a task, models must infer the correct president.

Studying whether current models can use multiple knowledge sources effectively can help identify and debug errors that models make when relying on outdated sources. To this end, we introduce a coreference resolution task designed to probe models' ability to draw on knowledge available in different sources. Recent work by Longpre et al. (2021) examined the effects of knowledge conflicts across different knowledge sources. In our work, we aim to more broadly examine the behaviour of NLU models in the presence of multiple knowledge sources. While Longpre et al. (2021) study how models handle conflicting facts, our goal is to evaluate whether models can combine complementary knowledge drawn from multiple sources rather than choose between sources.

In our task, the resolution of a given pronoun requires two types of knowledge (see Figure 1): 1) entity-specific knowledge, e.g., "Servin is a judge" and 2) background knowledge, e.g., "Judges decide cases in law courts." Generally, background knowledge is learned during the pretraining of LLMs i.e., at pretrain-time, while entity-specific knowledge is typically observed at inference time. We vary the availability of the required information such that it may either be found in a single source or in multiple sources. We evaluate a model's ability to integrate and reason over the two knowledge types (entity-specific and background knowledge), given in two knowledge sources (pretrain-time and inference-time).

We propose KITMUS,[1] a diagnostic test suite. The KITMUS tests evaluate Knowledge InTegration from MUltiple Sources. KITMUS's distinguishing feature is that it contains texts in which we methodically vary the mapping of knowledge types to knowledge sources, which allows us to pinpoint the specific strengths and limitations of models. We also analyze the behaviour of models when knowledge is available only at inference-time by introducing variants where a model needs to reason over fictional knowledge, which is presumably not contained in the parameters. Unlike previous reasoning datasets, where inference-time knowledge is retrieved (Onoe et al., 2021), we provide the knowledge necessary to solve the task in each instance of KITMUS. This allows for a more controlled setting where we can focus on knowledge integration, rather than on retrieval, which we hold out as a separate problem. In a study with human participants, we empirically validated that both entity-specific and background knowledge are required to perform well on KITMUS, and that the automatically generated labels are consistent with human annotations.

We evaluate state-of-the-art coreference resolution models on the KITMUS. In our experiments, many established models appear unable to integrate knowledge from two different knowledge sources and reason over them. With task-specific training, two models—BERT4Coref (Joshi et al., 2019) and C2F (Lee et al., 2018)—demonstrate the ability to reason over both knowledge observed at pretrain time and at inference time. However, we find that the ability to integrate knowledge from different sources seems to depend on the knowledge type in that source. While knowledge integration through concatenation at inference time seems to be effective for entity-specific knowledge, experiments with fictional knowledge indicate that even the best performing models cannot reliably integrate all types of background knowledge when provided only at inference time.

2 Related Work

Coreference resolution as a reasoning task: There has been extensive work to study NLU models' ability to exploit linguistic knowledge that involves shallow cues such as gender, position, and number cues (Durrett and Klein, 2013), as well as other properties like semantic roles (Baker et al., 1998; Chambers and Jurafsky, 2009). The Winograd Schema Challenge (WSC) (Levesque et al., 2012) inspired a number of specialized datasets such as GAP (Webster et al., 2018) and Winogrande (Sakaguchi et al., 2020) where coreference resolution is used as a test bed for reasoning over knowledge and cases cannot be solved with shallow features (Emami et al., 2019; Rahman and Ng, 2012). Following this line of work, we use templates that omit shallow cues, such that a model must integrate knowledge about the world to determine the coreference. While WSC and KnowRef focus on abstract external knowledge that is valid independent of the specific entities involved (Emami et al., 2019), KITMUS is more diverse and allows both entity-specific and background knowledge.

[1] https://github.com/mpoemsl/kitmus

World knowledge for reasoning tasks: Prior work has shown that integrating world knowledge can lead to improvement in coreference solvers. Bean and Riloff (2004) learn caseframe co-occurrence statistics, which they use to predict coreference. Rahman and Ng (2012); Zhang et al. (2019); Aralikatte et al. (2019); Emami et al. (2019) showed improved results using data augmentation.

Longpre et al. (2021) recognized the distinction between pretrain-time and inference-time knowledge, but they call them parametric and contextual knowledge. In the context of our work, the term "contextual" has many different interpretations and could consequently lead to misunderstandings. Therefore, we instead focus on the time at which the knowledge is typically observed in order to distinguish the two knowledge sources.

Chan et al. (2022) show that transformers exhibit different inductive biases in how they represent and generalize from the information in pretrain-time and inference-time knowledge sources using synthetic sequences of characters. Mallen et al. (2022) probe LMs on factual knowledge memorization using open-domain question answering and show improved results with retrieved knowledge augmentation. Complementing prior tasks that require background knowledge found in off-the-shelf knowledge bases, KITMUS instances require both entity-specific and background knowledge—we map a mentioned entity to its occupation and occupations to situations. Onoe et al. (2021) pose fact-checking tasks that require combining entity knowledge with commonsense knowledge. In contrast to our dataset, they do not provide the required knowledge, and expect models to either use only pretrain-time knowledge in a closed-book setting or to retrieve the knowledge from an external knowledge base at inference time. In our work, the knowledge associated with each instance is generated in a controlled way and provided as part of the inputs.

Reasoning over knowledge with Transformers: Zhou et al. (2021) present a dataset that evaluates the ability of pretrained Transformer language models to make inferences over axioms stated in natural language. Similarly, Clark et al. (2020) study the limits of reasoning in Transformer models with an approach where classical logic facts and rules are stated using natural language instead of a formal representation. Though our task is presented as a natural language text that requires reasoning, and is evaluated on Transformer models (among others), our work differs from Zhou et al. (2021) and Clark et al. (2020)'s in that the prediction target is the resolution of pronoun coreferences within a text. This requires identifying those mentions of an entity in a text that corefer with a pronoun using both pretrain-time and inference-time knowledge. In contrast, Zhou et al. (2021) and Clark et al. (2020) predict whether a conclusion is consistent with a preceding premise.

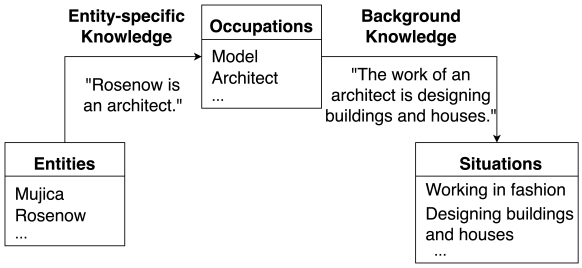


Figure 2: Schema of knowledge types in KITMUS.

3 The KITMUS Test Suite

We evaluate the knowledge integration capability of coreference resolution models from different knowledge sources: 1) pretrain-time: knowledge accumulated in the parameters during (pre-)training and 2) inference-time: knowledge observed in an input text.

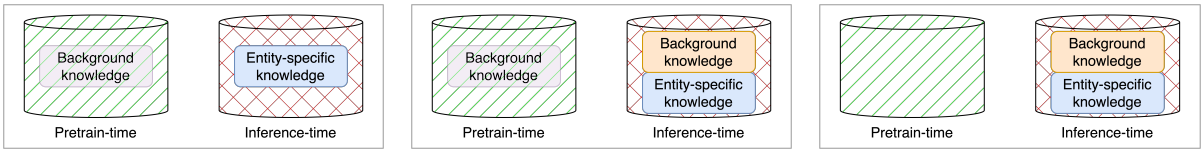
To design KITMUS, we formulate a coreference resolution task which requires access to two facts. We systematically vary the presence of these facts across the knowledge sources to evaluate the models. As an instantiation of the idea of presenting two facts, we experiment with the following two knowledge types:

– **Entity-specific**: occupation of an entity e.g., "Rosenow is an architect."
– **Background**: situation typical for an occupation e.g., "architects design building and houses."

For example, consider the following task to predict whether Mujica or Rosenow is the correct antecedent of the pronoun "he."

> Mujica is a model. Rosenow is an architect. At the bus station, **Mujica** and **Rosenow** connected. Public transports are eco-friendly. **He** shared experiences from a career of designing buildings and houses. [Answer: **Rosenow**]

Here, the occupations are *model* and *architect*, and the situational cue is *designing building and houses*. Both knowledge types are required in order to resolve this coreference. An illustration of this knowledge schema can be found in Figure 2.



(a) BACKGROUND-PRETRAIN (b) BACKGROUND-BOTH (c) BACKGROUND-INFERENCE

Figure 3: Variants of KITMUS based on the source of background knowledge.

We explore three main variants of the dataset as shown in Figure 3. With entity-specific knowledge always provided in the instance, the variants differ based on when and where background knowledge is available:

– BACKGROUND-PRETRAIN: Background knowledge is available only in the model parameters
– BACKGROUND-BOTH: Background knowledge is available in the model parameters and explicitly provided in the instance
– BACKGROUND-INFERENCE: Background knowledge is available only in the instance

Each instance of the KITMUS task consists of two fragments of text that are concatenated: 1) a knowledge text—containing the inference-time knowledge that models are given access to—and 2) a task text—consisting of the coreference task that models solve.

3.1 BACKGROUND-PRETRAIN

In this variant, entity-specific knowledge is provided at inference time and background knowledge about occupations is assumed to be pretrain-time knowledge, since information such as "architects design buildings and houses" is likely to have been observed during pretraining. An example is shown in the previous section. Here, the entity-specific knowledge about Mujica and Rosenow is inference-time; however, the knowledge about the occupations of a model and architect is pretrain-time. With this variant, we evaluate whether models have the ability to integrate and reason over both pretrain-time and inference-time knowledge effectively.

3.2 BACKGROUND-BOTH

In this variant, background knowledge is provided at both inference-time and assumed to be captured by the parameters. Entity-specific and background facts are present in the same knowledge source. They both represent inference-time knowledge being listed in the knowledge text as part of the inference-time inputs. For example:

> **Chichester** is a politician. The work of a politician is seeking an elected seat in government. **Klose** is an astronomer. The work of an astronomer is studying the stars and the universe. **Chichester** and **Klose** met at the train station. After a long day at work seeking an elected seat in government, **she** was happy to relax.
> [Answer: **Chichester**]

3.3 BACKGROUND-INFERENCE

In order to evaluate whether a model can solve this task using exclusively inference-time knowledge (i.e., in the absence of pretrain-time knowledge), we introduce fictional "knowledge." Fictional knowledge such as "the work of a mornisdeiver is gupegaing advaily" is unlikely to have been observed during pretraining, in contrast to real-world knowledge which is likely to have been observed. The entities in all variants are always fictional, ensuring that entity-specific knowledge about them was not observed at pretrain time. Thus, in this variant, both knowledge types are fictional and not contained in the pretrained parameters. For example:

> The work of a johumker is toncing ignaftedly. The work of a fangher is sparluing gobewly. Amezcua is a johumker. Hundley is a fangher. **Hundley** and **Amezcua** met at the yoga class. Yoga is best done in silence. **He** reflected on whether sparluing gobewly for a living was a good career choice.
> [Answer: **Hundley**]

Background knowledge about occupations maps occupations to situations that are typical for the occupation, such as "astronomer" and "studying the stars and the universe." To make background knowledge fictional, either the occupation, the situation, or both have to be fictional. For situations, we furthermore distinguish between levels of fictionality and define two sub-variants: 1) word-level fictional situations that use existing words but describe novel occupations, and 2) character-level fictional situations that use novel words. The methods we use to generate these fictional occupations and situations are detailed in the following section. Example texts resulting from different forms of fictionality can be seen in Table 1.

Var.	Occupation	Situation	Example
BB	Real	Real	The work of a *politician* is *seeking an elected seat in government*. Chichester is a politician[...]
BI	Real	CharFict	The work of a *politician* is *ehemting smorbtly*. Chichester is a politician[...]
BI	Real	WordFict	The work of a *politician* is *controlling the pool of an aircraft by using its directional flight controls*. Chichester is a politician[...]
BI	CharFict	Real	The work of a *mirituer* is *seeking an elected seat in government*. Chichester is a mirituer[...]
BI	CharFict	CharFict	The work of a *mirituer* is *ehemting smorbtly*. Chichester is a mirituer[...]
BI	CharFict	WordFict	The work of a *mirituer* is *controlling the pool of an aircraft by using its directional flight controls*. Chichester is a mirituer. [...]

Table 1: Different combinations of fictional occupations and situations in BACKGROUND-INFERENCE (BI) variant. An instance of BACKGROUND-BOTH (BB) variant is also shown.

4 Dataset Creation

To construct KITMUS, we manipulate which entities are mentioned in each instance, what occupations those entities have, what situations those occupations pertain to, what contexts they are mentioned in, and whether noise is present. Each entry is structured to first (1) introduce the entities, (2) then place them in the same location, and (3) finally, place one of them in a situation related to their occupation. In the BACKGROUND-BOTH and BACKGROUND-INFERENCE variants, this is preceded by a knowledge text mapping entities to their respective occupations using the phrase "is a."

The dataset entries are generated using handcrafted English-language templates and sampling from a variety of resource pools to fill the template slots. The use of templates facilitates control over the source a certain type of knowledge is stored in, which may not be possible to do with a natural dataset.

We aim to minimize the likelihood of models learning to exploit any spurious correlations in the templates or resources and promote data diversity using the following methods:

– We use multiple templates for each sentence. Examples are shown in Table 4 in the Appendix.
– We sample from diverse resource pools to fill template slots as detailed in Section 4.1.
– We include location-dependent noise statements that act as distractors and serve to vary the distance between entities.
– We create canonical train, validation, and test splits for each variant that are generated using disjunct subsets of templates and resources.

With these measures, we ensure that all entity names, occupations, situations, locations, templates, and noise statements that occur in the test instances do not occur in the train instances.

4.1 Resource Pools

We collect 20,000 last names as entities, 60 common occupations as occupations and their associated job descriptions as situations and 112 common meet-up places as locations from a mix of governmental and other publicly available resources (see Appendix A.2.3 for more details). We manually filter for gender and semantic cues. For example, we remove the occupations that provide referential gender cues such as "fire-man" and locations that might provide surface cues related to an entity's occupation.

Pronouns are sampled randomly from both the gendered pronouns `he` and `she` as well as gender-indefinite pronouns such as singular `they` and the neopronouns `ey` and `ze` following the gender-inclusive coreference resolution dataset `GICoref` (Cao and Daumé III, 2020). Ideally, we would want the distribution of pronouns to approximate the frequency in naturally occurring text, but few reliable statistics exist to estimate them. We include 40% `he`, 40% `she`, 10% `they`, and 10% neopronouns.

Noise statements are sampled randomly from a collection of statements based on the selected location in order to maintain a natural flow of the text. Each location is associated with 25 noise sentences. These sentences are generated using GPT-2 (Radford et al., 2019), and then manually verified by the authors not to include cues related to any entity or occupation.

4.2 Fictional Occupations

To create fictional background knowledge that maps occupations to situations, we create fictional occupations and fictional situations. Following the

methodology of Malkin et al. (2021), we generate 60 names of fictional occupation by sampling from a character-level LSTM language model.

4.3 Dataset Formats

Each variant in KITMUS consists of three subtasks—based on the number of entities—with increasing difficulty: two entity, three entity, and four entity subtasks. Each subtask has train, validation and test splits with 2000, 400, and 2000 examples respectively. The size of KITMUS is comparable to that of the GAP dataset (Webster et al., 2018), which similarly tests for a specific phenomenon in ambiguous pronoun coreference resolution.

We provide the test suite in two different formats which are commonly used by state-of-the-art coreference solvers: the CoNLL 2012 format (Pradhan et al., 2012) and the GAP format (Webster et al., 2018). The CoNLL format allows for a comprehensive annotation of mentions of an entity—including in the knowledge text. The GAP format, however, allows for the annotation of only two entities and only one mention per entity.

4.4 Human Validation

To investigate whether human assessors agree on the resolution of our test cases and whether this resolution is in agreement with our automatically generated labels, we conduct a human validation study. We also investigate whether our assumption that both background and entity-specific knowledge are required to resolve the cases by including instances where the knowledge text is not provided to human participants.

We created a multiple-choice questionnaire by randomly selecting an instance from each variant of our dataset (e.g., BACKGROUND-PRETRAIN), from each subtask (e.g., two entities), and from each split (e.g., validation). Additionally, we included one instance from each variant and from each subtask where the participants were only given the task text and not the accompanying knowledge text. A total of 96 sampled instances were presented to six different participants in random order.

A high inter-annotator agreement of 0.938 as measured by Fleiss' Kappa (Fleiss et al., 2003) leads us to believe that human participants agree on the resolution of KITMUS test cases. We use accuracy as a measure of agreement with the automatically generated labels and find that mean accuracy aggregated over all participants and subtasks is higher than 0.9 for all variants when the knowledge text is given. As expected, when neither background nor entity-specific knowledge are given, accuracy is below 0.1 for all variants, since most participants indicate that the question cannot be answered. This suggests that there are no inadvertent cues that can be exploited by humans to solve the task without having access to the entity-specific knowledge and background knowledge contained in the knowledge text.

Additional details on collection and processing of resource pools, fictional occupations, dataset formats and human validation are in the Appendix.

5 Experimental Setup

5.1 Model Selection

In this work, we focus on state-of-the-art and well-known coreference resolution models. We experiment with two families of coreference resolution models: 1) general coreference models and 2) pronoun coreference models.

Models that focus on general coreference resolution are often trained on the large Ontonotes corpus in the CoNLL 2012 format (Pradhan et al., 2012). We include BERT4Coref (Joshi et al., 2019) as an example of a state-of-the-art models on CoNLL 2012, C2F (Lee et al., 2018), which is the direct successor to the first end-to-end neural coreference resolution model (Lee et al., 2017), and Stanford's statistical (Clark and Manning, 2015) and neural (Clark and Manning, 2016) models.

Models that focus on pronoun coreference resolution are trained on the smaller GAP dataset in the GAP format (Webster et al., 2018). We include GREP (Attree, 2019), the winner of the GAP Kaggle competition and PeTra (Toshniwal et al., 2020), an efficient memory-augmented model.

5.2 Training

We conduct task-specific training with all models on the train split of KITMUS using their default hyperparameters. The larger general coreference models BERT4Coref and C2F are conventionally not trained on datasets with just 2000 train instances such as GAP or KITMUS, but rather trained on Ontonotes and then evaluated on smaller datasets (Joshi et al., 2019). Since coreference cases in KITMUS diverge significantly from those in Ontonotes, we test these models both in the Ontonotes-trained setting and KITMUS-trained setting. For these models, we report mean metrics over 6 runs. We

use only the pretrained versions of the Stanford models, since they are conventionally used off-the-shelf. We train the GAP-based models—PeTra and GREP—only on the two entity subtasks following the GAP format constraints outlined earlier. Additional training details are in Appendix A.5.

5.3 Evaluation

We evaluate all models on the KITMUS test split of each subtask. We use two metrics to assess each model performance: antecedent classification F1 and pronoun accuracy. Antecedent classification F1 is typically used for GAP format datasets. It considers the coreference between each candidate antecedent mention and the pronoun as a binary classification decision i.e., for a text with two entities, it considers two binary predictions and calculates the scores accordingly. Pronoun accuracy considers for each pronoun whether the correct candidate antecedent is predicted by the model, so independent from the number of entities in a text, only one decision is made among all possible candidate antecedents. We compare against a random baseline, which is implemented as random choice among the gold candidate mentions.

6 Experimental Results

6.1 BACKGROUND-PRETRAIN

Table 2 shows that none of the evaluated models are able to outperform the random baseline without task-specific training on KITMUS. Some models exhibit below random performance, indicating that they may fail to recognize and choose the correct mentions that could be antecedents. When trained on KITMUS, BERT4Coref (all) and C2F (for the four-entities subtask) perform significantly better than random, as shown in Table 2b. The high performance of BERT4Coref and C2F on the BACKGROUND-PRETRAIN variant suggests that both models have the ability to draw background knowledge from their parameters, entity-specific knowledge from the inference-time inputs, and reason over them on-the-fly with task-specific training.

The performance of all models we experimented with generally decreases as the number of entities increases; which is unsurprising since the more candidate entities there are, the less likely the accidental selection of the correct entity becomes. Moreover, we observe high variance across the six runs of KITMUS-trained C2F (see Table 7 in the Appendix A.6).

In order to explore the effect of noise statements, we conduct additional experiments on the BACKGROUND-PRETRAIN variant without noise. The removal of noise does not result in a significant performance change (see Table 8 in the Appendix).

6.2 BACKGROUND-BOTH and BACKGROUND-INFERENCE

We conduct additional experiments on the BACKGROUND-BOTH and BACKGROUND-INFERENCE variants with BERT4Coref and C2F, since they demonstrate the ability to learn the BACKGROUND-PRETRAIN variant of the task. In Table 3, we report results on the four-entity subtask, which Table 2 suggests to be the most challenging. While BERT4Coref's performance on the BACKGROUND-BOTH is comparable to its BACKGROUND-PRETRAIN variant results, C2F's performance is much worse, suggesting that it cannot effectively absorb the background knowledge provided at inference time and is distracted by it. On the BACKGROUND-INFERENCE variant, BERT4Coref seems to be able to integrate background knowledge about fictional occupations by outperforming the random baseline. However, it shows the ability to integrate word-level fictional, but not character-level fictional knowledge.

7 Discussion

Models trained on "general" coreference datasets fail on KITMUS: The poor performance of Ontonotes-trained models suggests that when trained on general coreference resolution datasets, models learn to exploit surface cues, which does not help when testing on KITMUS where such cues are removed. Another factor might be the structure of the texts in KITMUS, which are designed to place knowledge in specific knowledge sources. This might affect models' abilities to form useful representations resulting in poor performance of Ontonotes-trained models. These failures suggest that training on (what are meant to be) "general" datasets is not enough to induce knowledge integration from multiple sources and task-specific training is required.

Effect of dataset format and size: We observe that the models that accept input in the CoNLL format (Pradhan et al., 2012) perform better than those models that accept the GAP format (Webster et al., 2018). This indicates that mention annotations

Model	2 Entities	3 Entities	4 Entities
BERT4Coref	**0.43**	**0.18**	**0.14**
C2F	0.34	**0.18**	0.13
Stanford Neural	0.20	0.10	0.09
Stanford Stat.	0.05	0.02	0.01
Random	0.50	0.33	0.25

(a) Ontonotes-trained

Model	2 Entities	3 Entities	4 Entities
BERT4Coref	**0.99**	**0.98**	**0.94**
C2F	0.52	0.28	0.48
GREP†	0.49	-	-
PeTra†	0.01	-	-
Random	0.50	0.33	0.25

(b) KITMUS-trained

Table 2: Accuracy on BACKGROUND-PRETRAIN variant of KITMUS. Models marked with † operate on GAP format which only allows for the annotation of two entities. All other models operate on the CoNLL format. F1 scores shown in Table 6 track the accuracy scores. Note that models are not forced to choose between entities.

Var.	Occupation	Situation	BERT4Coref	C2F
BB		Real	0.96	0.09
BI	Real	CharFict	0.25	0.18
BI		WordFict	0.48	0.08
BI		Real	0.43	0.08
BI	CharFict	CharFict	0.26	0.18
BI		WordFict	0.38	0.11

Table 3: KITMUS-trained accuracy on BACKGROUND-BOTH (BB) & BACKGROUND-INFERENCE (BI) variants of KITMUS with 4 entities. Random performance is 0.25.

in the knowledge text—which only the CoNLL format provides—might be significant.

To evaluate whether the failure cases are due to the small train set size of 2000, we repeat experiments with a train set size of 5000. While we do see some improvements, the general trends persist and our observations remain consistent with the previous results (see the limitations section for additional discussion). This suggests that further scaling of the train set size might not be sufficient to improve performance on cases where existing models are currently failing.

Performance of current pretrained LLMs: BERT4Coref seems to consistently outperform C2F. This might be due to the difference in pretrained LLMs: BERT4Coref uses the Transformer architecture (Vaswani et al., 2017), which has been shown to be effective at reasoning tasks presented in natural language form (Clark et al., 2020) and utilizing information presented in inference-time contexts (Petroni et al., 2020), while C2F uses ELMo (Peters et al., 2018). To verify that BERT and ELMo contain background knowledge mapping occupations to situations, we ran a LAMA probe (Petroni et al., 2020). We find that BERT is more likely to contain the background knowledge compared to ELMo (see Section 8 for details). This corroborates the better performance of BERT on knowledge intensive tasks such as KITMUS.

Integration of fictional knowledge: As shown in Table 3, BERT4Coref performs consistently poorly on character-level fictional situations compared to real and word-level fictional situations. An example of character-level fictional occupation knowledge erroneously answered by BERT4Coref is shown below:

> The work of a remaller is socring clatodemnly. **Nims** is a mamser. **Formica** is a remaller. The work of a mamser is slimbing murstly. At the birthday party, **Nims** and **Formica** ran into each other. The party is filled with local and national celebrities and entertainers. **She** shared experiences from a career of socring clatodemnly.
> [Correct answer: **Formica**; BERT4Coref: **Nims**]

One possible reason could be BERT's tokenization strategy, which involves pooling subword representations (Devlin et al., 2019). In character-level fictional words, the subwords are meaningless, rendering their representations unhelpful. This is consistent with previous work showing that representations of LLMs for character-level fictional "Jabberwocky" words are less useful (Kasai and Frank, 2019) and that the presence of out-of-vocabulary (OOV) words decreases performance of neural models for NLU tasks (Schick and Schütze, 2020; Moon and Okazaki, 2020; He et al., 2021).

Despite the character-fictional occupations and situations, we expect the models to resolve the coreferences successfully in this setting. In the given example, the pronoun "she" can be resolved by matching the situation "socring clatodemnly" to the occupation "remaller" (using the word overlap between the situations and the occupation descriptions) and identifying the correct entity associated with the occupation i.e, Formica.

Humans can successfully make these inferences by matching fictional occupations and situations. However, the current models do not perform better than a random baseline in this setting. Our hope is that eventually, models should be able to handle

even knowledge presented in previously unknown terms. Given that languages are forever growing, robustness to neologisms is crucial, considering that OOV words e.g., new occupations like "TikToker" develop constantly.

Effects of knowledge type: Experiments on the BACKGROUND-PRETRAIN variant indicate that BERT4Coref is able to integrate fictional entity-specific knowledge observed at inference time reliably, yet this does not seem to be the case for fictional background knowledge. This suggests that models' ability to integrate and reason over the knowledge on-the-fly depends on the knowledge type—whether the knowledge is background or entity-specific—and not on whether it is fictional or not. One possible explanation could be that LMs observed different frequencies of unseen entities, occupations, and situations during pretraining, which result in a difference in their ability to adapt to novel instances of those categories.

8 Conclusion

We investigated the ability of models to integrate knowledge from multiple knowledge sources to resolve linguistic ambiguities in a coreference resolution task. We formulated a task that requires access to two knowledge types, entity-specific and background knowledge, and controlled for two knowledge sources that knowledge is available in, pretrain-time and inference-time.

Our results show that with task- and dataset-specific training, some models have the ability to reason over both knowledge observed at pretrain time and at inference time. For these models, knowledge can be integrated by concatenating textual knowledge to the model inputs. Furthermore, our findings imply that supplying additional information (e.g., from a retriever) at inference time to models can be successful even if the knowledge required for the task has not been observed before. However, in our task this ability seems to require task-specific training and depend on the type of knowledge being supplied.

Future work could explore finetuning models on KITMUS to encourage knowledge integration across different sources. One might also consider extending the KITMUS test suite to other languages or to create a multilingual test suite. Instructions for using our code and adapting the templates and resources to other languages can be found in Appendix A.1.

Limitations

Data diversity: As a template-generated dataset, KITMUS does not reflect the full diversity of natural data. However, we do not attempt to emulate the diversity of natural datasets. Using templates over natural data for diagnostic purposes has a few advantages. Templates facilitate control over the source of a certain type of knowledge, which may not be possible to do with more natural datasets like Ontonotes. This allows us to isolate the model behavior we want to probe. We also take several steps to add diversity, like using multiple templates, sampling from large resource pools, random shuffling of entities, addition of noise sentences, and canonical data splits with non-overlapping templates and resources. To prevent spurious factors at lexical level, the templates are hand-crafted to remove surface cues and validated in a study with human participants.

Background Knowledge Assumption in LMs: The results of our work is based on the assumption that pretrained LMs have access to background knowledge about real occupations. To verify that the pretrained LMs evaluated in this work contain background knowledge mapping occupations to situations, we ran a LAMA probe (Petroni et al., 2020) on BERT and ELMo. Given the template "The work of a [MASK] is [SITUATION].", we compared the probabilities the LMs assigned to all single-token occupation names used in KITMUS (probing for multi-token words is not supported by LAMA). BERT assigned higher probabilities to the correct occupation than to any other occupation for 90% of occupations. ELMo assigned the highest probability to the correct occupation for only 45% occupations, which might contribute to explaining why the ELMo-based model C2F generally performs worse than BERT4Coref on the BACKGROUND-PRETRAIN variant KITMUS, which requires such knowledge about occupations.

Root Word Overlap: One potential limitation of testing for non-fictional background knowledge like "firefighters put out fires" is that the natural occurrence of the root word "fire" in both occupation and situation might enable models to solve the task without having access to background knowledge. An analysis of trigram overlaps in all occupation-situation pairs shows that 45% of non-fictional occupations have at least one overlapping root word. However, a comparison of performances on those samples with and without root word overlap

showed neither systematic increase nor decrease for any model, indicating that models do not rely on the root word mappings. Results split up by root word overlap can be found in Table 10.

Train Set Size: The size of the train set for KITMUS, 2000, was chosen to mirror that of GAP (Webster et al., 2018). To evaluate whether the failure of models to learn the task is due to the relatively small number of samples observed during training, we re-generated all variants with 5000 train examples and repeated all experiments. We observe an increase in the magnitude of performance both in BERT4Coref and C2F on those variants where performance was higher than random performance with 2000 examples, but not on those that were equal to or below random performance. Consistent with previous results, BERT4Coref performs well on BACKGROUND-PRETRAIN and BACKGROUND-BOTH, but not on all fictional BACKGROUND-INFERENCE variants (Tables 7 and 13). We release the KITMUS generation code to enable experimentation with other train set sizes in future work.

Ethical Considerations

While KITMUS is intended as a diagnostic tool, users should be aware of the possibility of unintended biases when interpreting model performances on this dataset. To document these in more detail, our dataset release will be accompanied by a datasheet (Gebru et al., 2018) which is included in Appendix A.7.

Despite the synthetic nature, depending on its use, KITMUS might also have adverse impacts. The randomized sampling of resources to fill slots is meant to minimize bias in terms of the demographic cues that might be associated with the entities referenced in our tests (e.g., gender and nationality). The names and occupation descriptions in our test suite are drawn from United States governmental resources or English-language websites. This means that our test suite is not representative and likely skewed in terms of names, locations, occupations, and situations more common in the e.g., anglophone world. Additional resources such as noise statements and fictional entities were generated using word-level and character-level language models trained on English-language texts, which are known to reproduce a variety of biases found in natural data (Bordia and Bowman, 2019; Solaiman et al., 2019).

Our human validation study was IRB approved.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable suggestions. This work was supported by Microsoft Research. Jackie Chi Kit Cheung is supported by the Canada CIFAR AI Chair program, and is also a consulting researcher for Microsoft Research. The authors acknowledge the material support of NVIDIA in the form of computational resources. This research was enabled in part by compute resources provided by Mila (`mila.quebec`).

References

Rahul Aralikatte, Heather Lent, Ana Valeria Gonzalez, Daniel Herschcovich, Chen Qiu, Anders Sandholm, Michael Ringaard, and Anders Søgaard. 2019. Rewarding coreference resolvers for being consistent with world knowledge. 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 1229–1235, Hong Kong, China. Association for Computational Linguistics.

Sandeep Attree. 2019. Gendered ambiguous pronouns shared task: Boosting model confidence by evidence pooling. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 134–146, Florence, Italy. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

David Bean and Ellen Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 297–304, Boston, Massachusetts, USA. Association for Computational Linguistics.

Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yang Trista Cao and Hal Daumé III. 2020. Toward gender-inclusive coreference resolution. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4568–4595, Online. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.

Stephanie CY Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K Lampinen, and Felix Hill. 2022. Transformers generalize differently from information stored in context vs in weights. *arXiv preprint arXiv:2210.05675*.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*.

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.

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

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

Joseph L. Fleiss, Bruce Levin, and Myunghee Cho Paik. 2003. *The Measurement of Interrater Agreement*, chapter 18. John Wiley and Sons, Ltd.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for datasets.

Keqing He, Yuanmeng Yan, and Weiran Xu. 2021. From context-aware to knowledge-aware: Boosting oov tokens recognition in slot tagging with background knowledge. *Neurocomputing*, 445:267–275.

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.

Jungo Kasai and Robert Frank. 2019. Jabberwocky parsing: Dependency parsing with lexical noise. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 113–123.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

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

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

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nikolay Malkin, Sameera Lanka, Pranav Goel, Sudha Rao, and Nebojsa Jojic. 2021. GPT perdetry test: Generating new meanings for new words. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5542–5553, Online. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*, page 273.

Sangwhan Moon and Naoaki Okazaki. 2020. Patchbert: Just-in-time, out-of-vocabulary patching. In *Proc. of EMNLP*, pages 7846–7852.

Yasumasa Onoe, Michael J. Q. Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge.

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

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, Jeju Island, Korea. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proc. of AAAI*, volume 34, pages 8732–8740.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proc. of AAAI*, volume 34, pages 8766–8774.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

Shubham Toshniwal, Allyson Ettinger, Kevin Gimpel, and Karen Livescu. 2020. PeTra: A Sparsely Supervised Memory Model for People Tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5415–5428, 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 *In NeurIPS*, volume 30. Curran Associates, Inc.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Hongming Zhang, Yan Song, Yangqiu Song, and Dong Yu. 2019. Knowledge-aware pronoun coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 867–876, Florence, Italy. Association for Computational Linguistics.

Pei Zhou, Rahul Khanna, Seyeon Lee, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. 2021. RICA: Evaluating robust inference capabilities based on commonsense axioms. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7579, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

A.1 Creating a Custom Dataset

Our code can be used to create a custom dataset in different languages by using custom resources in place of the canonical resources listed in A.2.

Detailed instructions for how to do this can be found in the code repository's README[2] file.

A.2 Dataset-specific Resources

This section details the resources that were used to create the KITMUS dataset.

A.2.1 Templates

Table 4 shows the sets of templates used to to introduce and refer to entities.

A.2.2 Fictional Occupations and Situations

We generally follow the methodology of Malkin et al. (2021) in creating fictional occupations and siutations. To bias the model towards strings that can be used as occupation names, we train it on a reversed sequence of characters and prompt with the suffix `er`. We manually filter the words to eliminate unpronounceable or pre-existing English words.

We employ the following two methodologies to generate fictional situations: 1) character-level fictional—like the fictional occupations—is generated with the suffix prompts `ing` and `ly`, and 2) word-level fictional is generated by randomly shuffling existing words with the same POS tags followed by manual filtering based on semantic plausibility. Examples are shown in Table 1.

A.2.3 Resource Pools

Entities are sampled from a pool of the 20,000 most frequent last names in the 2010 U.S. census.[3] We use last names as entity names in order to avoid introducing gender-related cues. We discard those last names that are also first names. The order of entities within a template is also randomized. We assume that there is no confounding pretrain-time knowledge based on the entity names in the models.

Occupations consist of a curated list of 60 common occupations compiled by scraping a career website[4] and the US Labor census data.[5] Following Cao and Daumé III (2020), we remove referential gender cues from the occupations such as "fireman." The jobs pertaining to very specific domains or related to one of the locations where entities meet are removed from the list.

Situations are assembled using the occupation descriptions of the scraped occupations. We manually filter the pairs of situations that are semantically similar, such as an accountant and an analyst.

Locations are derived from a curated list of 112 locations scraped from a website of common meetup places.[6] We manually filter out locations that could provide inadvertent surface cues related to the entities' occupation, nationality, or gender.

A.3 Dataset Format

The CoNLL format contains token and sentence boundaries, Penn Treebank POS tags (Marcinkiewicz, 1994), and gold coreference clusters for all entity mentions. This means that all mentions of an entity—including in the knowledge text—are annotated in a single cluster. Models that operate on the CoNLL format predict these clusters, which involves both detecting mentions and clustering them. In contrast, the GAP format allows for the annotation of only two entities and only one mention per entity (excluding the pronoun), so entity mentions in the knowledge text remain unannotated. Models that operate on the GAP format are presented with exactly two mentions and for each of them make a binary decision whether or not they are coreferring with a pronoun. The GAP format task is more restricted in that models do not have to detect mentions and there are at most two entities per instance.

A.4 Human Validation

The participants were undergraduate and graduate students with fluency in English which were recruited via an institution-wide open call. The participants were compensated with the equivalent

[2] https://github.com/mpoemsl/kitmus/blob/main/README.md

[3] https://www.census.gov/topics/population/genealogy/data/2010_surnames.html

[4] https://ca.indeed.com/career-advice/finding-a-job/common-jobs

[5] https://www.bls.gov/emp/tables/emp-by-detailed-occupation.htm

[6] https://www.happierhuman.com/meet-new-people/

At {location}, {mentions} met.
At {location}, {mentions} ran into each other.
At {location}, {mentions} started a conversation.
At {location}, {mentions} came across each other.
At {location}, {mentions} encountered each other.
At {location}, {mentions} bumped into each other.
At {location}, {mentions} connected.
{mentions} met at {location}.
{mentions} ran into each other at {location}.
{mentions} started a conversation at {location}.
{mentions} came across each other at {location}.
{mentions} encountered each other at {location}.
{mentions} bumped into each other at {location}.
{mentions} connected at {location}.

(a) Meet Sentence Templates

After a long day at work {situation}, {pronoun} was happy to relax.
{pronoun} told anecdotes from a career of {situation}.
{pronoun} reflected on whether {situation} for a living was a good career choice.
When a question related to {situation} arose, {pronoun} offered a professional opinion.
{pronoun} was relieved to unwind after a demanding day at work {situation}.
{pronoun} was glad to unwind after a long day at work {situation}.
{pronoun} shared experiences from a career of {situation}.
{pronoun} pondered whether choosing {situation} as a career was a wise decision.

(b) Pronoun Sentence Templates

Table 4: Templates used to introduce ("Meet Sentence") and refer to ("Pronoun Sentence") entities in KITMUS task.

of 12 USD for their participation.[7] The study was approved by the institution's ethics review board and the participants gave their written consent via a form.

The participants were tasked to resolve the coreferences in a randomly sampled subset of KITMUS texts. The task is presented to the participants as a multiple choice questionnaire. The participants are given gold mentions and have to select the antecedent that is referred to by the pronoun. The answer options include the names of all mentioned entities and a "can't say" option to indicate that the question is not answerable. The questionnaire contained 96 questions to be completed in 60 minutes, which was generous for most participants.

The human validation was conducted using Google forms. The participants are introduced to the task with examples as shown in Figure 4.

This is followed by 96 questions where the participants have to choose one option among all entity names and the option "can't say," which indicates that the task cannot be solved for this instance. The aggregated results of the validation study are shown in Table 5.

A.5 Training Details

We train our models in a compute cluster infrastructure on Nvidia Quadro RTX 8000 GPUs. For BERT4Coref, training on the train split of one KITMUS subtask took about 8 hours per run. For C2F it took about 16 hours, the training of the ensemble model GREP took 18 hours. The training of smaller models and inference on pretrained models took about 4 hours per run.

[7]Matches the minimum wage in the participants' demographic

A.6 Additional Experiments

As a supplement to our main experiments, we report the following experiment results on the BACKGROUND-PRETRAIN variant:

- F1 score in Table 6
- Accuracy with 5000 instead of 2000 train examples in Table 7
- Accuracy without noise in Table 8
- Accuracy on train set in Table 9
- Accuracy with and without root word overlap in Table 10

On the BACKGROUND-BOTH and BACKGROUND-INFERENCE variants, we report:

- F1 score in Table 11
- Accuracy on train set in Table 12
- Accuracy with 5000 instead of 2000 train examples in Table 13

A.7 Datasheet

A.7.1 Motivation

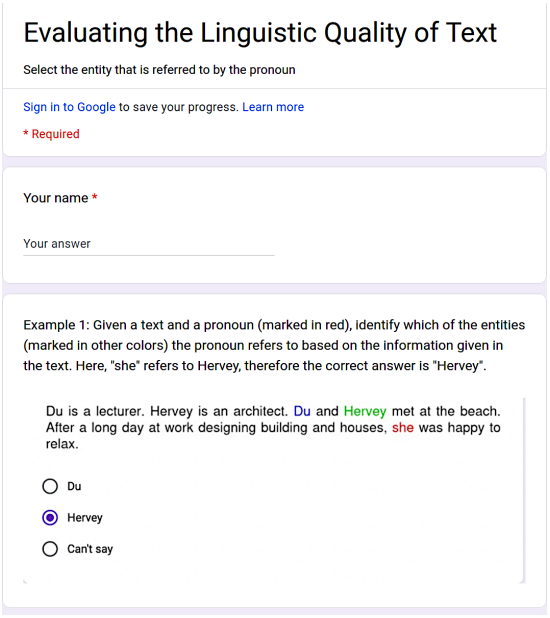
For what purpose was the dataset created?

The KITMUS dataset was created to enable research on reasoning over knowledge for the task of coreference resolution - i.e. given a piece of text, identify mentions and determine whether or not they co-refer. The dataset was created with the intention to focus on those cases of coreference resolution that require knowledge about specific entities and their occupations to accomplish the task.

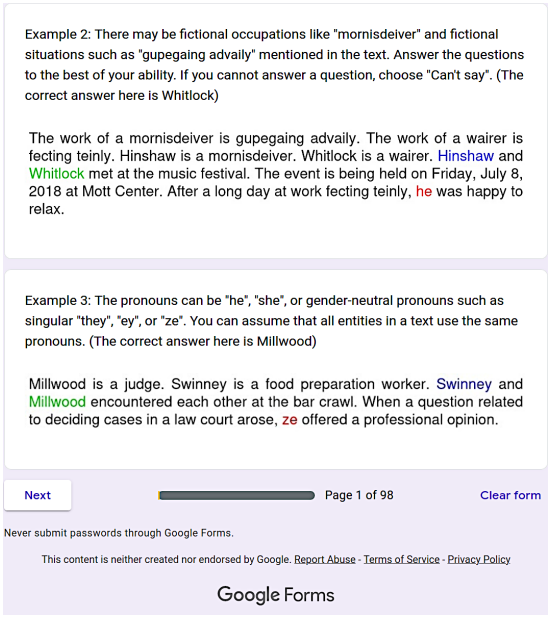
Who created the dataset and on behalf of which entities?

The dataset was created by the authors of this paper.

Who funded the creation of the dataset?



(a) Top Half



(b) Bottom Half

Figure 4: Human validation questionaire introduction (split into two halves because of space constraints).

Variant	Occupation	Situation	With Knowledge	Without Knowledge
BACKGROUND-PRETRAIN	Real	Real	0.93	0.00
BACKGROUND-PRETRAIN without noise			0.91	0.00
BACKGROUND-BOTH	Real	Real	1.00	0.00
BACKGROUND-INFERENCE		CharFict	1.00	0.00
BACKGROUND-INFERENCE		WordFict	0.98	0.00
BACKGROUND-INFERENCE	CharFict	Real	0.98	0.00
BACKGROUND-INFERENCE		CharFict	0.98	0.00
BACKGROUND-INFERENCE		WordFict	0.96	0.06

Table 5: Accuracy on all variants aggregated over subtasks, splits, and participants. Random performance is 0.25. Human participants could select "can't say," which is never in agreement with the automatically generated labels.

Model	2 Entities	3 Entities	4 Entities
BERT4Coref	**0.49**	0.24	0.19
C2F	0.48	**0.33**	**0.25**
Stanford Neural	0.29	0.15	0.13
Stanford Stat.	0.09	0.04	0.02
Random	0.50	0.33	0.25

(a) Ontonotes-trained

Model	2 Entities	3 Entities	4 Entities
BERT4Coref	**0.99**	**0.99**	**0.94**
C2F	0.52	0.35	0.48
GREP†	0.49	-	-
PeTra†	0.67	-	-
Random	0.50	0.33	0.25

(b) KITMUS-trained

Table 6: Antecedent F1 on BACKGROUND-PRETRAIN variant of KITMUS. Models marked with † operate on GAP format which only allows for the annotation of two entities. All other models operate on the CoNLL format. PeTra has higher F1 scores than pronoun accuracy, since it defaults to always predicting `true` for each antecedent, which results in a recall of 1.00 and a thus a high F1 score.

Funding was provided by multiple sources as mentioned in the acknowledgements in section 8.

Any other comments?

None.

A.7.2 Composition

What do instances that comprise the dataset represent?

The dataset consist of text pairs that were generated to capture knowledge about entities, occupations, and situations, as well as coreference cases whose resolution depends on this knowledge. The labels are clusters of tokens in the text.

How many instances are there in total?

There are $4400 \cdot 3 \cdot (2+1+5) = 105600$ instances in total: 4400 instances for each of the three entity numbers for variants BACKGROUND-PRETRAIN (also without noise), BACKGROUND-BOTH, and

Model	Train Data	2 Entities		3 Entities		4 Entities	
		2k	5k	2k	5k	2k	5k
PeTra	KITMUS	0.00	0.01	-	-	-	-
GREP		0.49	0.50	-	-	-	-
BERT4Coref		0.99 ± 0.00	1.00 ± 0.00	0.98 ± 0.01	0.97 ± 0.00	0.94 ± 0.01	0.94 ± 0.02
C2F		0.52 ± 0.02	0.58 ± 0.06	0.28 ± 0.08	0.63 ± 0.03	0.48 ± 0.06	0.24 ± 0.08
Random	-	0.50	0.50	0.33	0.33	0.25	0.25

Table 7: Accuracy on BACKGROUND-PRETRAIN variant of KITMUS with 2000 (2k) and 5000 (5k) train examples. Standard deviation is given after $\pm$.

Model	Train Data	2 Entities		3 Entities		4 Entities	
		Noise	No Noise	Noise	No Noise	Noise	No Noise
BERT4Coref	Ontonotes	0.43	0.43	0.18	0.23	0.14	0.13
C2F		0.34	0.34	0.18	0.18	0.13	0.14
Stfd. Neural		0.20	0.33	0.10	0.15	0.09	0.14
Stfd. Stat.		0.05	0.15	0.02	0.06	0.01	0.06
PeTra	KITMUS	0.00	0.01	-	-	-	-
GREP		0.49	0.49	-	-	-	-
BERT4Coref		0.99	1.00	0.98	0.98	0.94	0.92
C2F		0.52	0.52	0.28	0.34	0.48	0.24
Random	-	0.50	0.50	0.33	0.33	0.25	0.25

Table 8: Accuracy on BACKGROUND-PRETRAIN variant of KITMUS with and without noise.

Model	Train Data	2 Entities		3 Entities		4 Entities	
		Test	Train	Test	Train	Test	Train
PeTra	KITMUS	0.00	0.01	-	-	-	-
GREP		0.49	0.51	-	-	-	-
BERT4Coref		0.99	1.00	0.98	1.00	0.94	1.00
C2F		0.52	0.96	0.28	1.00	0.48	1.00
Random	-	0.50	0.50	0.33	0.33	0.25	0.25

Table 9: Test and train accuracy on BACKGROUND-PRETRAIN variant of KITMUS.

Model	Train Data	2 Entities		3 Entities		4 Entities	
		Overlap	No Overlap	Overlap	No Overlap	Overlap	No Overlap
BERT4Coref	Ontonotes	0.43	0.45	0.18	0.19	0.15	0.14
C2F		0.34	0.36	0.17	0.19	0.13	0.12
Stfd. Neural		0.20	0.19	0.11	0.08	0.08	0.09
Stfd. Stat.		0.05	0.04	0.02	0.01	0.01	0.00
PeTra	KITMUS	0.00	0.01	-	-	-	-
GREP		0.47	0.52	-	-	-	-
BERT4Coref		0.99	0.99	0.99	0.97	0.95	0.92
C2F		0.53	0.50	0.29	0.26	0.49	0.46
Random	-	0.50	0.50	0.33	0.33	0.25	0.25

Table 10: Accuracy on BACKGROUND-PRETRAIN variant of KITMUS with and without root word overlap.

five versions of BACKGROUND-INFERENCE with different degrees of fictionality.

Does the dataset contain all possible instances or is it a sample of instances from a larger set?

The dataset contains all instances that we generated. They are generated by filling slots in a template by sampling from a pool of resources. The pool of resources only contains a subset of resources in the world, and the sampling process selects a random subset of the pool of resources.

What data does each instance consist of?

The instances are pairs of template-generated texts: one knowledge text and one task text. The knowledge text contains knowledge about fictional

Var.	Occupation	Situation	C2F	BERT4Coref
BB		Real	0.11	0.96
BI	Real	CharFict	0.20	0.25
BI		WordFict	0.10	0.49
BI		Real	0.09	0.43
BI	CharFict	CharFict	0.21	0.27
BI		WordFict	0.14	0.39

Table 11: KITMUS-trained F1 Score on BACKGROUND-BOTH (BB) and BACKGROUND-INFERENCE (BI) variants of KITMUS with four entities. Random performance is 0.25.

entities and real or fictional occupations in text form. The task text contains a case of coreference involving the same fictional entities. Labels for the coreferences are given in the form of coreference clusters over tokens.

Is there a label associated with each instance?

Yes. The label is a coreference cluster that represents the true resolution of the coreference presented in the text.

Is any information missing from individual instances?

No.

Are relationships between individual instances made explicit?

Yes. The entities are fictional and created separately for each instance. Instances are completely independent from each other and are not consistent across the dataset, i.e. conflicting knowledge may be given for the same fictional entity across different instances in the dataset.

Are there recommended data splits?

Yes. Each subcategory of the dataset is provided in recommended data splits of 2000 `train` instances, 400 `validation` instances, and 2000 `test` instances. The numbers are chosen for size comparability with other coreference resolution datasets such as GAP (Webster et al., 2018). Resources are disjunct across the splits for each subcategory, which enables the evaluation of the ability of models to generalize beyond observed resources.

Are there any errors, sources of noise, or redundancies in the dataset?

None that we are aware of. Since the dataset is template-generated, only the intentionally provided noise in the appropriate subcategory is present. We control for redundancies in the dataset. A human validation has not brought to light any errors in the dataset, however, due to the synthetic nature of the dataset texts can appear wooden and non-natural to readers.

Is the dataset self-contained, or does it link to or otherwise rely on external resources?

The dataset is created using external resources to fill slots in templates, but the finished dataset is entirely self-contained.

Does the dataset contain data that might be considered confidential?

The dataset contains only information about fictional entities and public knowledge about occupations which is not confidential.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?

Both the templates and the resources used to fill the slots were manually inspected for content that might cause anxiety to viewers.

The dataset does not contain any text that might cause anxiety to viewers.

Does the dataset identify any subpopulations?

The fictional entities have neither an explicit age nor gender. The only distinguishing features of the entities are their names and occupations, which are uniformly sampled, and their pronoun use, which is sampled according to the following distribution: 40% `he`, 40% `she`, 10% `they`, and 10% neopronouns.

Is it possible to identify individuals either directly or indirectly?

No. Since the entities are entirely fictional, any similarities to existing individuals are due to chance.

Does the dataset contain data that might be sensitive in any way?

No.

Any other comments?

None.

A.7.3 Collection Process

How was the data associated with each instance acquired?

The data was generated by filling slots in templates that were hand-engineered. The slot-filling resources were obtained from publicly available raw text sources such as governmental name statistics and professional job websites. Noise sentences were generated with the language model GPT-2 (Radford et al., 2019) and manually edited and verified to conform with the rest of the dataset. Fictional occupation names and descriptions were created by random sampling from a character-level LSTM language model following methodology of Malkin et al. (2021).

Variant	Occupation	Situation	C2F Test	C2F Train	BERT4Coref Test	BERT4Coref Train
BB		Real	0.09	1.00	0.96	1.00
BI	Real	CharFict	0.18	0.97	0.25	0.88
BI		WordFict	0.08	0.95	0.48	0.73
BI		Real	0.08	0.96	0.43	0.97
BI	CharFict	CharFict	0.18	0.83	0.26	0.78
BI		WordFict	0.11	1.00	0.38	0.96

Table 12: Train and test accuracy on BACKGROUND-BOTH (BB) and BACKGROUND-INFERENCE (BI) variants of KITMUS. Random performance is 0.25.

Variant	Occupation	Situation	C2F 2k	C2F 5k	BERT4Coref 2k	BERT4Coref 5k
BB		Real	0.09	0.49	0.96	0.97
BI	Real	CharFict	0.18	0.25	0.25	0.27
BI		WordFict	0.08	0.26	0.48	0.78
BI		Real	0.08	0.21	0.43	0.57
BI	CharFict	CharFict	0.18	0.25	0.26	0.26
BI		WordFict	0.11	0.25	0.38	0.59

Table 13: KITMUS-trained accuracy on BACKGROUND-BOTH (BB) and BACKGROUND-INFERENCE (BI) variants of KITMUS with four entities with 2000 (2k) and 5000 (5k) train examples. Random performance is 0.25.

What mechanisms or procedures were used to collect the data?

The dataset was generated using Python scripts, which will be made publicly available in a GitHub repository.

If the dataset is a sample from a larger set, what was the sampling strategy?

Not applicable. The entire dataset will be released.

Who was involved in the data collection process and how were they compensated?

Not applicable. There was no human involved in the dataset creation prcoess.

Over what timeframe was the data collected?

The dataset was created immediately prior to the submission of this draft for review.

Were any ethical review processes conducted for the data collection process?

Not applicable, data was not collected. The human evaluation study used to evaluate the dataset was approved by an institutional review board.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources?

The dataset was created via templates. The resources were collected directly from publicly available data online.

Were the individuals in question notified about the data collection?

The resources were collected directly online from institutions and authors who made the resources available publicly. The authors and institutions were not explicitly informed about the way their resources are used in this dataset.

Did the individuals in question consent to the collection and use of their data?

Not applicable.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?

Not applicable.

Has an analysis of the potential impact of the dataset and its use on data subjects been conducted?

No.

Any other comments?

None.

A.7.4 Preprocessing

Was any preprocessing/cleaning/labeling of the data done?

The template building blocks were manually tokenized and POS tagged with the Stanford CoreNLP pipeline, which was then manually verified. In terms of resources, the occupations were filtered manually to avoid overlaps in descriptions. Referential gender cues such as "fireman" were removed from the occupations. Occupations pertaining to very specific domains or related to location were removed from the list. GPT-2 generated noise sentences were manually checked for coherence and also tokenized and POS tagged with the Stanford CoreNLP pipeline. Fictional occupation names and

descriptions were likewise manually checked for coherence and suitability.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?

No.

Is the software that was used to preprocess/clean/label the data available?

The Stanford CoreNLP pipeline is available here: https://stanfordnlp.github.io/CoreNLP/.

Any other comments?

None.

A.7.5 Uses

Has the dataset been used for any tasks already?

None.

Is there a repository that links to any or all papers or systems that use the dataset?

Not applicable.

What (other) tasks could the dataset be used for?

The dataset could potentially be used for research on mention detection, cross-document coreference resolution, or entity linking, since the annotations are compatible with these tasks as well.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?

Due to its template-generated nature, the data does not consist of naturally occurring texts and should not be used for purposes which require naturally occurring texts.

Are there tasks for which the dataset should not be used?

The entities in the texts are entirely fictional and have an arbitrary distribution of attributes. Consequently, the information in this dataset should not be used to make decisions about real people.

Any other comments?

None.

A.7.6 Distribution

Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?

Yes, the dataset will be available publicly on the internet.

How will the dataset be distributed?

The dataset will be released in the GitHub repository for this paper.

When will the dataset be distributed?

Upon publication of the corresponding paper.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?

The dataset and the code used to generate it will be distributed under the license specified in the GitHub repository for the dataset. In the repository, we will also request to cite the corresponding paper if the dataset is used.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances?

None that we are aware of.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?

None that we are aware of.

Any other comments?

No.

A.7.7 Maintenance

Who will be supporting/hosting/maintaining the dataset?

The first authors will support and maintain the dataset.

How can the owner/curator/manager of the dataset be contacted?

Contact the first authors.

Is there an erratum?

No. Future updates and known errors will be specified in the `README.md` of the repository.

Will the dataset be updated?

Currently, no updates are planned.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?

Not applicable, since the entities are fictional.

Will older versions of the dataset continue to be supported/hosted/maintained?

In the case of updates, the original version of the dataset will always be available on GitHub via a tagged release.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?

Suggestions for the augmentation of the dataset can be made via GitHub pull requests.

Any other comments?

None.

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?
A.8.2

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

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

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
A.8.6

☑ 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)?
A.8.5

☑ 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?
A.8.2

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
A.8.2

☑ 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.
A.8.2

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?
A.6

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.)?
A.8.4

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
4.4

☑ 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.?
A.5

☑ 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)?
A.5

☑ 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?
A.5

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
A.5

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
A.5