The KITMUS Test for Knowledge Integration from Multiple Sources

Anonymous ACL submission

Abstract

Natural language understanding models 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 tasks that require reasoning over multiple facts and an accompanying dataset with individual subtasks that we vary in order to control the knowledge source of relevant facts. 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 train 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.

1 Introduction

Progress on natural language understanding (NLU) benchmarks has recently been driven by pretrained large language models (LLMs), which may 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 train-time knowledge contained in their parameters, usually acquired via pretraining.

Recent work suggests that models can use train-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 was only supplied at inference time, because of, e.g., time sensitivity or instance specificity. Consider the passage "John saw the 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 train-time and inference-time knowledge.

> ***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.

Figure 1: Example from KITMUS. To resolve the pronoun "he," a model needs to draw on entity-specific knowledge about an entity's occupation as well as on background knowledge about the occupation itself.

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 the knowledge sources and reason over them on the fly. For example, pretrained parameters might contain the knowledge that Donald Trump is the president of 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.

We know little about how models make use of multiple knowledge sources. Drawing on recent work examining the effects of knowledge conflicts across different knowledge sources (Longpre et al., 2021), we aim to more broadly examine the behaviour of NLU models in the presence of different knowledge sources. We introduce a coreference resolution task designed to probe models' ability to draw on knowledge available in different knowl-

edge sources, including in the presence of varying numbers of entities and noise. Unlike Longpre et al. (2021), where the focus is on conflicting facts, we control for when complementary information is made available to models.

Specifically, in our task, the resolution of a given pronoun requires two knowledge types as shown in Figure 1: (1) entity-specific knowledge, such as "Sevin is a judge" and (2) background knowledge, such as "Judges decide cases in law courts". Background knowledge is usually learned during the pretraining of LLMs and therefore considered train-time knowledge, 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 different sources. We evaluate a model's ability to integrate and reason over the two knowledge types given in two knowledge sources.

We propose KITMUS, a test suite containing instances of our task. Similar to how a litmus test checks for acidity, the KITMUS test evaluates Knowledge InTegration from MUltiple Sources. KITMUS's distinguishing feature is that it contains texts in which we methodically vary the mapping of the knowledge types to the knowledge sources, which allows us to pinpoint the specific strengths and limitations of models. We also analyze the behaviour of models when the knowledge is contained only in the instance by introducing variants where a model needs to reason over fictional knowledge, which is presumably not contained in the parameters. Unlike previous works, where the 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. We validate in a human evaluation study that both background and entity-specific knowledge are required to perform well on KITMUS and that the automatically generated labels are consistent with human annotation[1].

We evaluate state-of-the-art coreference resolution models on the KITMUS test suite. In our experiments, many established models appear unable to integrate knowledge from two different knowledge sources and reason over them without task-specific training. 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 train time and at inference time. However, we find that the ability to integrate knowledge from different sources seems to the 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 providing background knowledge only at inference time is not sufficient.

[1]Code for generation and evaluation will be made available on GitHub.

2 Related Work

Coreference resolution as a reasoning task: Coreference resolution is the task of determining which mentions in a text corefer. In the general case, which is presented in large coreference resolution datasets such as Ontonotes (Pradhan et al., 2012), this can mostly be accomplished by exploiting shallow cues such as gender, position, and number cues (Durrett and Klein, 2013). There has been extensive work to study NLU models' ability to exploit linguistic knowledge that involves these shallow cues, 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 smaller 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. Moreover, KITMUS involves a more diverse set of knowledge. While WSC and KnowRef focus on abstract external knowledge that is valid independent of the specific entities involved (Emami et al., 2019), KITMUS focuses on both entity-specific and entity-agnostic knowledge.

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.

In the wake of the WSC, several NLU datasets such as bAbi (Weston et al., 2015) and OpenBookQA (Mihaylov et al., 2018) were proposed that demand reasoning over knowledge (Mishra et al., 2018; Mitra et al., 2019). Longpre et al. (2021) recognized the distinction between train-time and inference-time knowledge, which they call parametric and contextual knowledge. The latter is usually retrieved at inference time from an unstructured (Koupaee and Wang, 2018) or structured (Rebele et al., 2016; Liu and Singh, 2004; Singh, 2002) knowledge base.

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, drawing from Onoe et al. (2021). In their dataset, they pose fact-checking tasks that require combining entity knowledge with commonsense knowledge. However, in contrast to our dataset, they do not provide the required knowledge, and expect models to either use only train-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 of KITMUS is generated and provided in a controlled setting.

Reasoning over knowledge with Transformers: Clark et al. (2020) study the limits of reasoning in transformer models. They investigate an approach where classical logic facts and rules are stated using natural language instead of a formal representation, and train transformers to reason over these types of sentences.

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 Clark et al. (2020)'s in that the prediction target is the resolution of pronoun coreferences within a text. This requires detecting multiple candidate mentions and identifying those that corefer with a pronoun using both train-time and inference-time knowledge. In contrast, the prediction target of Clark et al. (2020) is the boolean decision if a claim is consistent with a set of facts and rules. Our experiments corroborate the results of Clark et al. (2020) as we find that the Transformer-based model BERT4Coref (Joshi et al., 2019) can reason effectively over inference-time knowledge.

3 The KITMUS Test Suite

In this work, we evaluate the knowledge integration capability of NLU models in the presence of two knowledge sources: 1) train-time: knowledge accumulated in the parameters during (pre-)training and 2) inference-time: knowledge observed in the instance.

We formulate a coreference task whose resolution 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 choose the following knowledge types:

– **Entity-specific**: occupation of an entity e.g., "Telles is a firefighter."
– **Background**: situation typical for an occupation e.g., "a firefighter is putting out fires."

For example, consider the following task to predict whether Telles or Drayer is the correct antecedent of the pronoun "she".

> **Telles** is a firefighter. **Drayer** is a bus driver. **Telles** and **Drayer** met at the sports bar. After a long day at work putting out fires, **she** was happy to relax. [Correct answer: **Telles**]

Here, the occupations are *firefighter* and *bus driver*, and the situational cue is *putting out fires*. Both the knowledge types are required in order to resolve this coreference. An illustration of this knowledge schema can be found in Figure 2.

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-TRAIN: 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 only available in the instance

Each instance of the task consist of two texts that are concatenated: a knowledge text—containing the inference-time knowledge that models are given access to—and a task text—consisting of the coreference task that models solve.

3.1 BACKGROUND-TRAIN

In this variant, entity-specific knowledge is provided at inference time and background knowledge about occupations is assumed to be train-time

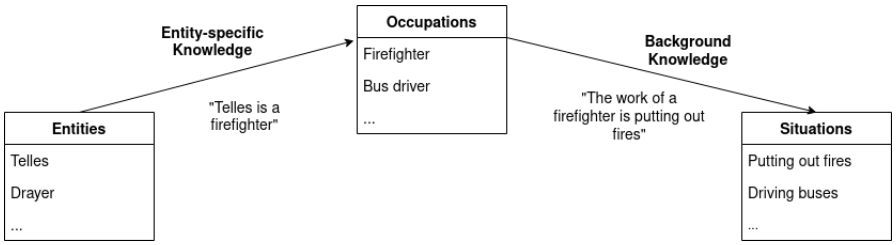

Figure 2: Schema of different knowledge types in KITMUS.

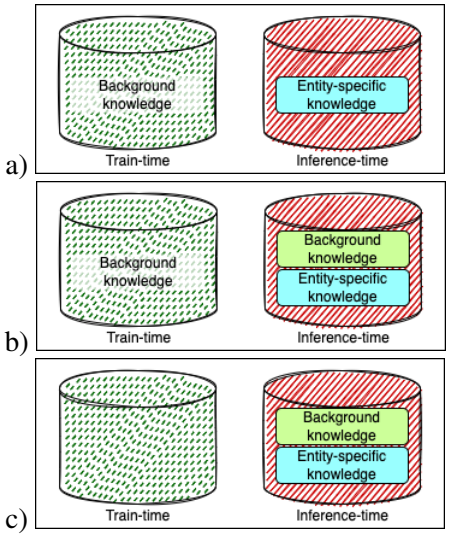

Figure 3: Variants of KITMUS based on the source of background knowledge: (a) BACKGROUND-TRAIN (b) BACKGROUND-BOTH (c) BACKGROUND-INFERENCE

knowledge since information such as "the work of a firefighter is putting out fires" is likely to have been observed during pretraining. An example is shown in Section 3. Here, the entity-specific knowledge about **Telles** and **Drayer** is inference-time; however, the knowledge about the jobs of a firefighter and a bus driver is train-time. By evaluating on this variant, we evaluate whether models have the ability to integrate and reason over both train-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:

> **Telles** is a firefighter. The work of a firefighter is putting out fires. **Drayer** is a bus driver. The work of a bus driver is driving buses. **Telles** and **Drayer** met at the sports bar. After a long day at work putting out fires, **she** was happy to relax.

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 train-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 such as "the work of a baker is baking bread", which is likely to have been observed. The entities in all variants are always fictional, which ensures that entity-specific knowledge about them has not been observed at train time. Thus, in this variant, both knowledge types are fictional and not contained in the pretrained parameters.

Background knowledge about occupations maps occupations to situations that are typical for the occupation, such as "baker" and "baking bread". 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 subvariants: 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 section 4.2. Example texts resulting from different forms of fictionality can be seen in Table 1.

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 if noise is present in the instance.

The dataset entries are generated using English-language templates. These templates are designed to control for variables pertaining to entities, occupations, and situations.

Var.	Occupation	Situation	Example
BB	Real	Real	The work of a *firefighter* is *putting out fires*. Whyte is a firefighter[...]After a long day at work putting out fires, he was happy to relax.
BI	Real	CharFict	The work of a *firefighter* is *ehemting smorbtly*. Whyte is a firefighter[...]After a long day at work ehemting smorbtly, he was happy to relax.
BI	Real	WordFict	The work of a *firefighter* is *controlling the pool of an aircraft by using its directional flight controls*. Whyte is a firefighter[...]After a long day at work studying the stars and the drink, he was happy to relax.
BI	CharFict	Real	The work of a *mirituer* is *putting out fires*. Whyte is a mirituer[...]After a long day at work putting out fires, he was happy to relax.
BI	CharFict	CharFict	The work of a *mirituer* is *ehemting smorbtly*. Whyte is a mirituer[...]After a long day at work ehemting smorbtly, he was happy to relax.
BI	CharFict	WordFict	The work of a *mirituer* is *controlling the pool of an aircraft by using its directional flight controls*. Whyte is a mirituer. [...]After a long day at work controlling the pool of an aircraft by using its directional flight controls, he was happy to relax.

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

Each entry is structured to first (1) introduce the entities, (2) then place them in the same location, and (3) finally, have one of them remember a situation related to their occupation. The `noise` is a statement about the location intended to act as a distractor to increase the task difficulty. It both makes the distance between pronoun and antecedents variable and the evaluation of reasoning abilities for NLU models more challenging. The template for a task text with two entities is:

> ⟨`entity`$_A$⟩ and ⟨`entity`$_B$⟩ met at ⟨`location`⟩. ⟨`noise`⟩. After a long day at work ⟨`situation`⟩, ⟨`pronoun`⟩ was happy to relax.

The knowledge text maps entities to their respective occupations using the phrase "is a". The template for providing inference-time entity-specific knowledge about two entities is:

> ⟨`entity`$_A$⟩ is a ⟨`occupation`$_A$⟩.
> ⟨`entity`$_B$⟩ is a ⟨`occupation`$_B$⟩.

4.1 Resource Pools

We generate texts by randomly sampling from predefined sets of named entities, occupations, situations, locations, and pronouns. We ensure that texts included in the train, validation, and test splits are drawn from non-overlapping subsets of names, occupations, locations, and noise statements.

Entities are sampled from a pool of the 20,000 most frequent last names from the 2010 U.S. census (United States Census Bureau, 2021). 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 train-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 (Indeed, 2021) and the US Labor census data (US Labor Census, 2021). 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 can 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 (Happier Human, 2019). We manually filter out locations that could provide inadvertent surface cues related to the entities' occupation, nationality, or gender.

Noise statements are sampled 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. The sentences are generated using GPT-2 (Radford et al., 2019) and manually verified not to include cues related to any entity or 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.

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 substask has train, validation and test splits with 2000, 400, and 2000 examples respectively. The size of KITMUS is similar to that of the GAP dataset (Webster et al., 2018), but is smaller compared to Ontonotes (Pradhan et al., 2012).

4.2 Fictional Occupations

In order to create fictional background knowledge that maps occupations to situations, we create fictional occupations and fictional situations. Following the work of Malkin et al. (2021), we generate 60 names of fictional occupation by sampling from a character-level LSTM language model with temperature 0.5. 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 and eliminate unpronounceable or pre-existing words in the English.

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.

4.3 Dataset Formats

We provide the test suite in two 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 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.

The GAP format operates on character indices rather than token indices and allows for the annotation of only two entities and only one mention per entity (excluding the pronoun). This means that only a single mention of an entity in the task text is annotated.

4.4 Human Validation

To assess the quality of KITMUS, we conducted a small human study with six participants. For this, we created a questionnaire by randomly selecting 5 instances from each subtask—tasks with two, three, and four entities—of the BACKGROUND-TRAIN variant. Additionally, from each subtask, we include 5 instances without any knowledge text. All the participants answered a total of 60 instances presented to them in a random order.

When the knowledge text was provided, most of the participants were able to identify the correct antecedent. Without knowledge text—no entity-specific knowledge—all participants indicated that the instances cannot be answered. This suggests that there are no inadvertent cues that can be exploited by humans to solve the task. The high inter-annotator agreement (0.994 as measured by Fleiss' kappa (Fleiss et al., 2003)) shows that the test suite has a high internal validity. The agreement of the participants with the automatically produced labels indicates that the data generation process is generally sound. The questionnaire and additional details can be found in Appendix A.1.

5 Experimental Setup

We evaluate existing coreference resolution models on the KITMUS test-suite.

5.1 Model Selection

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 train all models on the train split of KITMUS and use their default hyperparameters. The training details are in Appendix A.2. 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 train 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.

5.3 Evaluation

We test 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.

Additionally, we compare against two baselines: 1) human: the majority decision of human validation study participants and 2) random: random choice among the gold candidate mentions.

6 Results and Discussion

6.1 BACKGROUND-TRAIN

Table 2a shows that none of the evaluated models are able to outperform the random baseline without task-specific training on KITMUS. When trained on KITMUS, BERT4Coref (Joshi et al., 2019) and C2F (Lee et al., 2018) perform significantly better than random, as can be seen in Table 2b. The high performance of BERT4Coref and C2F on the BACKGROUND-TRAIN 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.

One possible reason for the poor performance of Ontonotes-trained models is that when trained on general coreference resolution datasets—like Ontonotes—they learn to exploit surface cues, which does not transfer to KITMUS where such cues are removed. Another explanation is that the structure of the texts in KITMUS, which is designed to place knowledge in specific knowledge sources, differs from that of Ontonotes. This might affect models' abilities to form useful representations. However, it is worth noting that the human study participants could solve the task without difficulties.

We observe that success in solving the task seems to coincide with the acceptance of input in the CoNLL format (Pradhan et al., 2012), while those models that accept the GAP format (Webster et al., 2018) perform poorly. This could be due to the lack of mention annotations in the knowledge text in the GAP format.

Furthermore, BERT4Coref seems to consistently outperform C2F. One reason for the better performance of BERT4Coref might be 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).

Performance of all models 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. In order to explore the effect of the noise statements, we conduct additional experiments on the BACKGROUND-TRAIN variant without noise. The removal of noise does not result in a significant performance change, as shown in Appendix A.3.

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-TRAIN variant of the task. In Table 3, we report results on the four-entity subtask, which Table 2 suggests to be the most

Model	2 Entities	3 Entities	4 Entities
BERT4Coref	**0.37**	**0.21**	0.11
C2F	0.34	0.17	**0.12**
Stfd. Neural	0.12	0.06	0.06
Stfd. Stat.	0.01	0.01	0.00
Random	0.50	0.33	0.25

(a) Ontonotes-trained

Model	2 Entities	3 Entities	4 Entities
BERT4Coref	**1.00**	**0.97**	**0.95**
C2F	0.64	0.51	0.47
GREP†	0.51	-	-
PeTra†	0.02	-	-
Random	0.50	0.33	0.25

(b) KITMUS-trained

Table 2: Accuracy on BACKGROUND-TRAIN variant of KITMUS. Models marked with † operate on GAP format, all other models operate on the CoNLL format. F1 scores shown in Appendix A.3 track the accuracy scores.

Var.	Occupation	Situation	C2F	BERT4Coref
BB		Real	0.45	0.95
BI	Real	CharFict	0.24	0.26
BI		WordFict	0.18	0.57
BI		Real	0.30	0.37
BI	CharFict	CharFict	0.27	0.23
BI		WordFict	0.22	0.27

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

challenging.

The performance of both the models on BACKGROUND-BOTH and BACKGROUND-TRAIN variants is comparable. This indicates that redundantly providing background knowledge both at train time and in inference-time inputs does not increase models' ability to absorb knowledge.

In the experiments on BACKGROUND-INFERENCE variant, models seem unable to integrate fictional background knowledge—fictional occupations and situations—observed at inference time. However, experiments on the other variants indicate that models are able to integrate fictional entity-specific knowledge observed at inference time reliably. This suggests that the 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 whether it is fictional or real. One possible explanation could be that LLMs observed different frequencies of unseen entities, occupations, situations during pretraining, which result in a difference in their ability to adapt to unseen fictional instances of those categories.

BERT4Coref in particular seems to perform consistently poorly on character-level fictional situations compared to real and word-level fictional situations. 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 the presence of out-of-vocabulary words decreases performance of neural models for NLU tasks (Schick and Schütze, 2020; Moon and Okazaki, 2020; He et al., 2021).

7 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, and controlled for the knowledge sources that the knowledge is available in.

Our results show that with task-specific training, some models have the ability to reason over both knowledge observed at train time and at inference time. For these models, knowledge can be integrated by concatenating textual knowledge to the model inputs. However, redundant information in multiple knowledge sources does not lead to further performance improvements. Furthermore, the ability of models to integrate inference-time knowledge on-the-fly seems to depend on its knowledge type. Our findings imply that supplying additional information (e.g., from a retriever) at inference time to NLU models can be successful even if the knowledge required for the task has not been observed before. However, for tasks similar to ours, adding information on-the-fly might not work in a zero-shot setting without task-specific training.

In future work, we would like expand the KITMUS test suite with different knowledge types and more naturalistic noise.

8 Ethical Considerations

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).

While KITMUS is intended as a diagnostic tool, users should be aware of these biases and the possibility of other 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), also included in Appendix A.4.

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.

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.

Happier Human. 2019. How to Meet People: 47 Best Places for Making New Friends. `https://www.census.gov/topics/population/genealogy/data/2010_surnames.html`. Online; accessed 19 March 2022.

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.

Indeed. 2021. 23 Common Jobs (With Salary Data and Primary Duties). `https://ca.indeed.com/career-advice/finding-a-job/common-jobs`. Online; accessed 19 March 2022.

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.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

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*.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

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.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and

models for process paragraph comprehension. *arXiv preprint arXiv:1805.06975*.

Arindam Mitra, Peter Clark, Oyvind Tafjord, and Chitta Baral. 2019. Declarative question answering over knowledge bases containing natural language text with answer set programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3003–3010.

Sangwhan Moon and Naoaki Okazaki. 2020. Patchbert: Just-in-time, out-of-vocabulary patching. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (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.

Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, pages 177–185. Springer.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774.

Push Singh. 2002. The open mind common sense project. *KurzweilAI. net*.

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.

United States Census Bureau. 2021. Frequently Occurring Surnames from the 2010 Census. `https://www.census.gov/topics/population/genealogy/data/2010_surnames.html`. Online; accessed 21 March 2022.

US Labor Census. 2021. Employment by detailed occupation. `https://www.bls.gov/emp/tables/emp-by-detailed-occupation.htm`. Online; accessed 19 March 2022.

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.

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.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

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.

A Appendix

A.1 Human Validation

The participants are graduate students with fluency in English which were recruited via an open call. The participants were compensated with the equivalent of 12 USD for their participation.[2] The study was approved by a university's ethics review board and the participants gave their written consent via a form.

The participants are tasked to resolve the coreferences in a randomly sampled subset of KITMUS texts. The task is presented to the participants in 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 contains 60 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 60 questions where the participants have to choose one option among all the entity names and "can't say" indicating that the task could not be solved.

A.2 Experiment Details

We train all models on Nvidia Quadro RTX 8000 GPUs in a compute cluster infrastructure. 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.

A.3 Additional Results

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.

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

Model	Train Data	2 Ent.	3 Ent.	4 Ent.
PeTra	KITMUS	0.66	-	-
GREP		0.51	-	-
BERT4Coref		1.0	0.97	0.95
C2F		0.64	0.51	0.48
BERT4Coref	Ontonotes	0.43	0.27	0.16
C2F		0.47	0.31	0.25
Stfd. Neural		0.20	0.10	0.10
Stfd. Stat.		0.02	0.02	0.00
Random	-	0.50	0.33	0.25

Table 4: Antecedent F1 on BACKGROUND-TRAIN variant of KITMUS.

Var.	Occupation	Situation	C2F	BERT4Coref
BB	Real	Real	0.45	0.95
BI		CharFict	0.24	0.27
BI		WordFict	0.19	0.57
BI	CharFict	Real	0.30	0.38
BI		CharFict	0.27	0.24
BI		WordFict	0.22	0.27

Table 5: KITMUS-trained F1 Score on BACKGROUND-BOTH (BB) and BACKGROUND-INFERENCE (BI) variants of KITMUS with four entities. Random performance is 0.25.

Model	Train Data	2 Ent.	3 Ent.	4 Ent.
PeTra	KITMUS	0.02	-	-
GREP		0.48	-	-
BERT4Coref		0.99	0.97	0.92
C2F		0.66	0.49	0.43
BERT4Coref	Ontonotes	0.42	0.23	0.13
C2F		0.33	0.19	0.13
Stfd. Neural		0.20	0.11	0.09
Stfd. Stat.		0.03	0.02	0.01
Random	-	0.50	0.33	0.25

Table 6: Pronoun accuracy on BACKGROUND-TRAIN variant of KITMUS with no noise statements added.

A.4 Datasheet

A.4.1 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 pa-

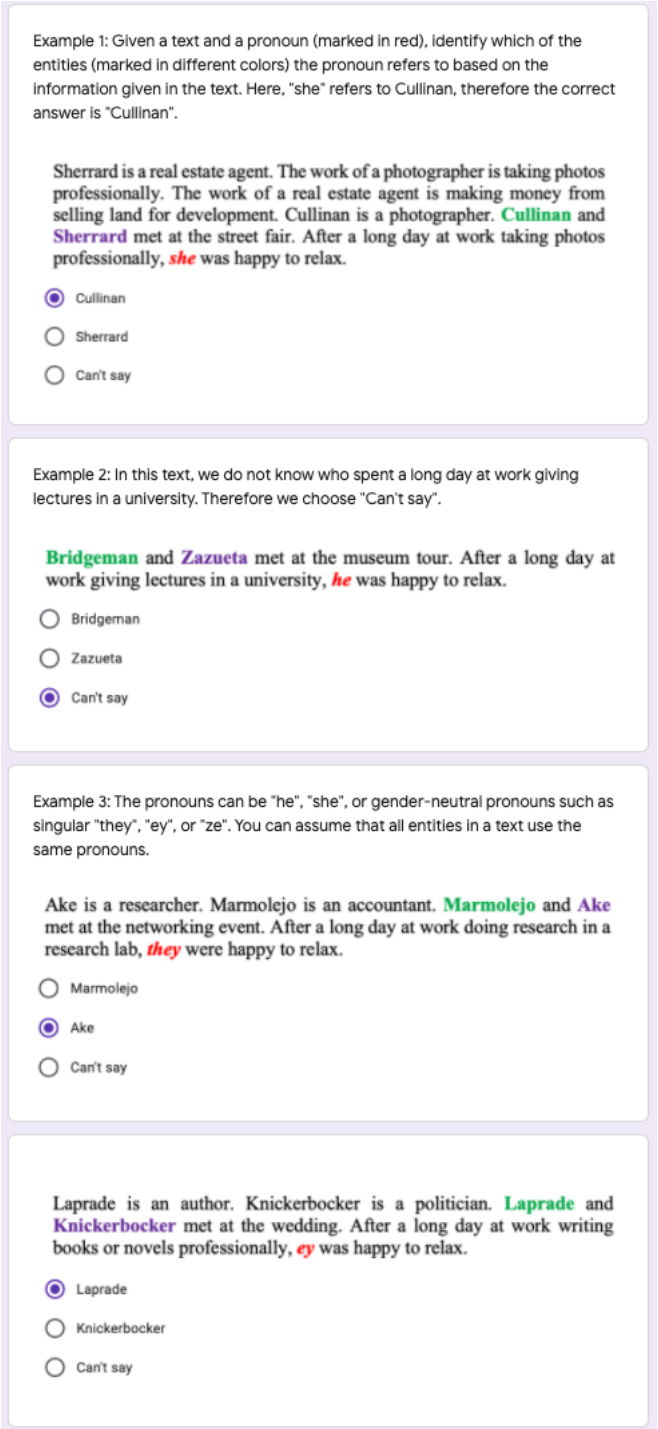

Figure 4: Task introduction with examples for the participants of human validation.

per (details omitted during review for anonymity).

Who funded the creation of the dataset?

Funding was provided by multiple sources (details omitted during review for anonymity).

Any other comments?

None.

A.4.2 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-TRAIN (also without noise), BACKGROUND-BOTH, and 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 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.4.3 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).

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.4.4 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.4.5 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.4.6 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 (details omitted for anonymity).

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.4.7 Maintenance

Who will be supporting/hosting/maintaining the dataset?

The first author(s) will support and maintain the dataset (details omitted for anonymity).

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

Omitted for anonymity.

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.