Unsupervised Relation Extraction from Language Models using Constrained Cloze Completion

Anonymous EMNLP submission

Abstract

We show that state-of-the-art self-supervised language models can be readily used to extract relations from a corpus without the need to train a fine-tuned extractive head. We introduce RE-Flex, a simple framework that performs constrained cloze completion over pretrained language models to perform unsupe rvised relation extraction. RE-Flex uses contextual matching to ensure that language model predictions matches supporting evidence from the input corpus that is relevant to a target relation. We perform an extensive experimental study over multiple relation extraction benchmarks and demonstrate that RE-Flex outperforms competing unsupervised relation extraction methods based on pretrained language models by up to 27.8 F_1 points compared to the next-best method. Our results show that constrained inference queries against a language model can enable accurate unsupervised relation extraction.

1 Introduction

Relation extraction is a fundamental problem in constructing knowledge bases from unstructured text. The goal of relational extraction is to identify mentions of relational facts (i.e., binary relations between entities of interest) in a text corpus. Traditionally, relation extraction systems leverage supervised machine learning approaches to train specialized extraction model for different relations (Dong et al., 2014; Shin et al., 2015). However, advances in natural language understanding models, and specifically contextual models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), have shifted the focus towards *general relation extraction* where a single natural language model is used for extraction across different relations (Levy et al., 2017).

A key idea behind general relation extraction is to leverage question answering (QA) models and use the reading comprehension capabilities of modern natural language models to identify relation mentions in text. For example, the relation `drafted_by` can be completed for the subject `Stephen Curry` by answering the question `Who drafted Stephen Curry?` State-of-the-art results, here, leverage fine-tuned QA models over self-supervised contextual representations (Devlin et al., 2018; Radford et al., 2018). Initial approaches (Levy et al., 2017) learn these *extractive QA models* by exploiting annotated question-answer pairs and following a supervised setting. While effective in domains related to the annotated question-answer data, supervised extractive QA approaches can fail to generalize to new domains for which annotations are not available (Dhingra et al., 2018). For this reason, more recent approaches (Lewis et al., 2019) propose to use automatically generated question-answer pairs for training and adopt a weakly-supervised setting (Lewis et al., 2019). However, noisy or inaccurate training data leads to a significant drop in performance.

In this work, we revisit the problem of general relation extraction and show that one can perform unsupervised relation extraction by directly using the generative ability of self-supervised contextual language models and without training a fine-tuned QA model. We build upon the recent observation that modern language models encode the semantic information captured in text and are capable of generating answers to relational queries by answering cloze queries that represent a relation (Petroni et al., 2019). For instance, the previous extraction example can be transformed to the cloze query `Stephen Curry was drafted by [MASK]` and the language model can be used to predict the most probable value for the masked token. Further, recent works (Radford et al., 2019; Petroni et al., 2020) shows that prefixing cloze queries with relevant information, i.e., *relevant* *context*, can improve extraction accuracy by utilizing the models' reading comprehension ability (Radford et al., 2019; Petroni et al., 2020). While promising, we show that an out-of-box application of these methods to general relation extraction falls short of extractive QA models. The core limitation is that of *factual generation*: language models do not memorize general factual information (Petroni et al., 2019), and are liable to predict off-topic or non-factual tokens (See et al., 2017).

We proposed a novel two-pronged approach that ensures factual predictions from a contextual language model. First, given an extractive relational cloze query and an associated context, we propose a method to restrict the model's answer to the query to be factual information in the associated context. Our methods introduces a context-constrained inference procedure over language models and does not require altering the pre-training algorithm. The method relies on redistributing the probability mass of the language model's initial prediction to tokens only present in the context. By restricting the model's inference to be present in the context, we ensure a factual response to a relational cloze query. This strategy is similar to methods used in unsupervised neural summarization (Zhou and Rush, 2019) to ensure factual summary generation. Second, we introduce an unsupervised solution to determining whether the context associated with the query contains an answer to a relational query. We propose an information theoretic scoring function to measure how well a relation is represented in a given context, then cluster contexts into "accept" and "reject" categories, denoting whether the contexts express the relation or not.

These are the two components of our unsupervised relation extraction system, RE-Flex. We show that RE-Flex enables fully unsupervised relation extraction using self-supervised language models. We present an extensive experimental evaluation of RE-Flex against state of the art general relation extraction methods across several settings. We demonstrate that RE-Flex outperforms methods that rely on weakly supervised QA models (Dhingra et al., 2018; Lewis et al., 2019) by up to 27.8 F_1 points compared to the next-best method, while it can even outperform methods that rely on supervised QA models (Levy et al., 2017) by up to 12.4 F_1 points in certain settings. Our results demonstrate that by constraining language generation, RE-Flex yields accurate unsupervised relation extractions.

2 Related Work

Typical relation extraction relies on rule-based methods (Soderland et al., 1995) and supervised machine learning models that target specific relation types (Hoffmann et al., 2011; Dong et al., 2014; Shin et al., 2015). These approaches are limited to predefined relations and do not extend to relations that are not specified during training. To alleviate this problem, open information extraction (OpenIE) (Banko et al., 2007) proposes to represent relations as unstructured text. However, in OpenIE different phrasings of the same relation can be treated as different relations, leading to redundant extractions. To address this issue, Universal Schema (Riedel et al., 2013) uses matrix factorization to link OpenIE relations to an existing knowledge base to distill extracted relations.

More recently, question answering has become a popular method to extract spans from text. Levy et al. (2017) showed that casting relation extraction as a QA problem can enable new, unseen relations to be extracted without additional training. Advances in large self-supervised language models (Radford et al., 2018) have enabled QA models to achieve human level performance on some datasets (Rajpurkar et al., 2016). Because these models are trained on a slot-filling objective, there has been a branch between methods that use a QA head to extract spans from input, and methods that use token generation capability of language models to perform information extraction. Both are relevant to our work.

Many QA-based methods have been proposed to identify spans from text. Das et al. (2018) present a reading comprehension model based on the architecture of Chen et al. (2017) to track the dynamic state of a knowledge graph as the model reads the text. Li et al. (2019) proposes a multi-turn QA system to extract relational fact triplets. Xiong et al. (2019) maps evidence from a knowledge base to natural language questions to improve performance in the general QA setting. Most relevant QA systems to our work are the recent works of Lewis et al. (2019) and Dhingra et al. (2018), which propose weak supervision algorithms to generate QA pairs over new corpora, and train models on these QA pairs.

There are also many generative methods that rely only on a language model to generate the answer to queries. Radford et al. (2019) show that self-supervised language models can generate an-

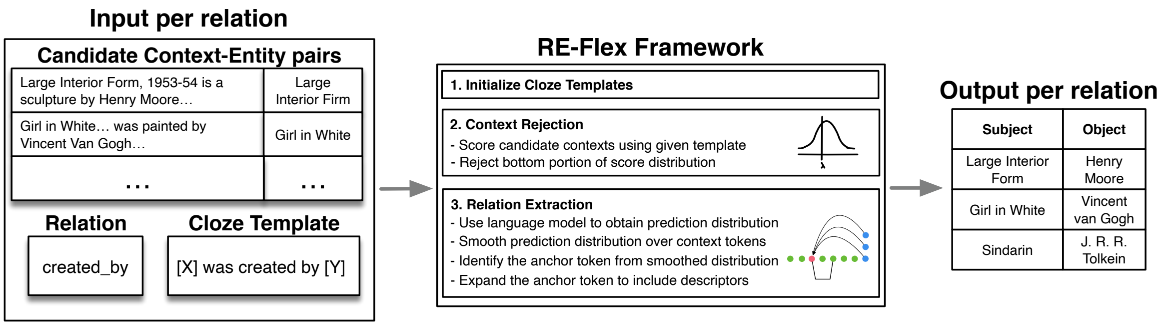

Figure 1: The RE-Flex Framework Overview

swers to questions. Petroni et al. (2019) show that given natural language cloze templates that represent relations, masked language models (Devlin et al., 2018) can answer relational queries directly. Petroni et al. (2020) extends on this work to show that retrieving factual evidence to associate with relation queries can further benefit answer generation. Logan et al. (2019) present a knowledge graph language model that can choose between outputting tokens from a base vocabulary, or entities from a linked knowledge base. Bosselut et al. (2019) show that language models can generate commonsense knowledge bases if pretrained on another corpus and fine-tuned on a commonsense knowledge base.

3 Problem Statement

We consider a slot filling form of relation extraction: given incomplete relations, we must complete the relations using evidence from an underlying text source. We assume a set of input relations R. For each relation $r \in R$, we assume access to a collection of entity-context candidate pairs. Let EC_r denote this collection for relation r. We consider each pair $(e, c) \in EC_r$ to be candidate evidence that some span in c completes relation r for the given entity mention e. If we consider the context to be composed of a sequence of tokens $c = (c_1, c_2, \dots, c_n)$, we must return some subsequence $a = (c_i, ..., c_{i+m})$ such that the relation $r(e, a)$ holds, or $\varnothing$ if c does not express the relation for the given entity.

Furthermore, we represent each relation with a cloze template, a natural language representation of what the relation is attempting to capture. We define a cloze template for relation r to be a sequence of tokens $t = (t_1, \dots, t_{sub}, \dots, t_{obj}, \dots, t_k)$, where t_{sub} and t_{obj} are special tokens denoting the expected locations of the subject and object entities of the relation. For each (e, c), we substitute the special token t_{sub} with e. Let $t(e) = (t_1, \dots, e, \dots, t_{obj}, \dots, t_k)$ denote this substitution. We form our final cloze query by concatenating the context c to the cloze task $t(e)$ and denote the close query $q(e, c) = [c, t(e)]$.

Given a cloze query $q(e, c)$, we express relation extraction as the following inference task: predict if there is a subsequence of the context c that correctly completes the special token t_{obj} in the close task $t(e)$, otherwise return $\varnothing$. As an example, consider the relation `drafted_by`. An example candidate entity-context pair in the pair set for the relation is (`Stephen Curry`, `The Warriors drafted Steph Curry.`). Using the relational template t_{sub} `was drafted by` t_{obj}, we form our full cloze query for the pair: `The Warriors drafted Steph Curry. Stephen Curry was drafted by` t_{obj}.

4 The RE-Flex Framework

An overview of RE-Flex is shown in Figure 1. Given a target relation, RE-Flex assumes as input a set of entities, a set of candidate contexts, and a cloze template expressing the relation. The output of RE-Flex is a table containing instances of this relation for the input entities. RE-Flex is built around two key parts: 1) context rejection and 2) anchor token identification and token expansion. In the first part, RE-Flex determines if the cloze query for a candidate entity-context pair does not contain a valid mention of the target relation, and hence, we must return $\varnothing$. In the second part, given the valid entity-context pairs for the target relation, RE-Flex identifies the subsequence in the corresponding context that completes the relation for the given entity. We describe each part next.

4.1 Context Rejection

For each relation r, we must determine which of the candidate pairs $(e, c) \in EC_r$ express relation r for entity e, and return $\varnothing$ for those that do not. The problem can be naturally considered as a clustering problem, where we group elements of EC_r into an accept cluster I_c or a reject cluster I_{-c}. Given this regime, we must develop a general method to determine how well a given entity-context pair (e, c) expresses a target relation. Using the natural language representation of the relation, we formulate a scoring function to measure how much each context expresses the relation. We then determine a threshold on these scores to partition the pairs.

We propose the following mechanism: First, we leverage the fact that the cloze template t for a target relation r is the natural language representation of the relation and assume that it captures the intention of the relation. We formulate a scoring function $f(c, t(e))$ which takes as input a context c and $t(e)$—the cloze template where we have substituted $t_{sub} = e$—and returns a measurement of how well each token in the template is captured in a given context. Second, for some threshold ϵ, if $f(c, t(e)) > \epsilon$, we assign the corresponding pair (e, c) to I_c, and to I_{-c} otherwise.

To design f, we reason about co-occurrence: if each word in the template co-occurs many times with any word in the context, the relation is likely to be expressed. We define f as follows:

$$f(c, t(e)) = \frac{1}{|t(e)|} \sum_{i=0}^{|t(e)|} \max_{j \in [1,|c|]} \text{PMI}(t(e)[i], c[j])$$

where PMI is the Pointwise Mutual Information (PMI) (Church and Hanks, 1990), $|t(e)|$ and $|c|$ are the total number of tokens in the cloze task $t(e)$ and the context c respectively, $t(e)[i]$ denotes the token in position i of the cloze task $t(e)$, and $c[j]$ denotes the token in position j of context c. For two words x and y, PMI is defined as $\text{PMI}(x, y) = \log \frac{p_q(x,y)}{p(x)p(y)}$, where $p_q(x, y)$ is the probability that x and y co-occur in a q-gram in the corpus and $p(x)$ is the marginal probability of x occurring in the corpus; we set $q = 5$.

We estimate PMI using the cosine similarity between the word embeddings produced by optimizing the skip-gram objective over a target corpus (Mikolov et al., 2013). This approach does not suffer from missing values in the PMI matrix, as an empirical estimate of the PMI matrix might (Levy and Goldberg, 2014). More formally we have: Let $v_x \in \mathbb{R}^d$ and $v_y \in \mathbb{R}^d$ be the word embeddings for two words x and y produced by training the model described in Mikolov et al. (2013). As proven in Arora et al. (2016), we have that:

$$\text{PMI}(x, y) \approx \frac{\langle v_x, v_y \rangle}{||v_x|| ||v_y||}$$

Given this metric, we now turn to how ϵ is computed. We use a simple inlier detection method to determine ϵ. We assume that entity-candidate contexts for each relation r are relatively well-aligned, i.e., the majority of elements in EC_r contain a true mention of relation r for the entity associated with each element. Let Q_r denote the set of all possible correct entity-context pairs for r. We assume that for any valid pair (e, c) the score $f(c, t(e))$ follows a normal distribution $\mathcal{N}(\mu_r, \sigma_r^2)$, and hence, we expect that for most entity-context pairs the similarity scores to the cloze task associated with the relation will be centered around the mean μ_r. Given the above modeling assumptions, we estimate μ_r and σ_r^2 as follows:

$$\mu_t = \frac{1}{|EC_r|} \sum_{(e,c) \in EC_r} f(c, t(e))$$

$$\sigma_t^2 = \frac{1}{|EC_r|} \sum_{(e,c) \in EC_r} (f(c, t(e)) - \mu_t)^2$$

Given the above, we let ϵ is $\epsilon = \mu_r - \lambda \sigma_r$ where λ is a hyperparameter. We assign all (e_r, c_r) pairs to I_c if $f(c_r, t_r) > \epsilon$, and assign the rest to I_{-c}. For all pairs in I_{-c}, we return $\varnothing$.

4.2 Relation Extraction

We discuss how RE-Flex performs relation extraction given a valid entity-pair context. For this part, we assume access to a pre-trained contextual language model—in RE-Flex we use RoBERTa (Liu et al., 2019). For a valid entity-context pair (e, c) for relation r, we construct the cloze query $q(e, c) = [c, t(e)]$ by replacing the subject mask token t_{sub} in the relation cloze template t with e, and given the sequence $q(e, c)$ we identify the span of tokens α in c that should replace the object mask token t_{obj} in $t(e)$ to correctly complete relation r for entity e.

At a high-level, we follow the next process to identify span α: first, we consider the raw predictions of the pre-trained model for t_{obj}, and smoothen the scores of these predictions by restricting valid predictions to correspond only to tokens present in the context c; we pick the context token with the highest final score, which we refer to as the *the anchor token*. Second, given the anchor token in c, we return an expanded span from c that contains descriptors of the anchor token. We describe each of these two steps next.

Anchor token identification We focus on the first step described above. Given an entity-context pair (e, c) that contains a true mention of relation r, the desired answer to the cloze query $q(e, c)$ corresponds to a span of tokens α in c. The task of anchor token identification is to identify any word in span α. To identify such a token, we constraint the inferences of the pre-trained model to tokens in the context c.

Given the cloze query $q(e, c) = [c, t(e)]$, also denoted hereafter q for simplicity, we first use the pre-trained model, denoted hereafter by M, to obtain a prediction for the masked token t_{obj} (see Section 3). Let V denote the vocabulary of all tokens present in the domain of consideration. For each token $v \in V$, we can use model M to obtain a probability that v should be used to complete the masked token t_{obj}. Let $p_{q,M}(v) = p(t_{obj} = v; q(e, c), M)$ denote this probability for token v.

To obtain a factual prediction, we reassign the above probability mass to the tokens that are contained only in the context c. We leverage the contextual model M for this step. For the token at each position in the context sequence c, we find all tokens in V that are semantically compatible with it, given the cloze query $q(e, c)$, and assign their mass to b. Consider the i-th position in the context c. We define the new probability mass for token $c[i]$, denoted by $z_{q,M}(c[i])$, as:

$$z_{q,M}(c[i]) = \sum_{v \in V} p_{q,M}(v) \cdot D(c[i], v)$$

where $D(c[i], v)$ is a non-negative normalized score indicating the semantic compatibility between tokens $c[i]$ and v. We have:

$$D(c[i], v) = \frac{\exp(d(c[i], v))}{\sum_{j=1}^{|c|} \exp(d(c[j], v))}$$

where the unnormalized scores $d(c[i], v)$ are defined using the contextual embeddings obtained by model M.

More formally, let $q^{e,c}(v)$ be the sequence corresponding to the cloze query $q(e, c)$ after we replace the masked object token t_{obj} in the cloze template of the target relation with some token $v \in V$. That is for context $c = \{c_1, c_2, \ldots, c_n\}$, entity e, and the cloze template $t = \{t_1, \ldots, t_{sub}, \ldots, t_{obj}, \ldots, t_k\}$, we have $q^{e,c}(v) = \{c_1, \ldots, c_n, t_1, \ldots, e, \ldots, v, \ldots, t_k\}$. Given model M and sequence $q^{e,c}(v)$, let $M(q^{e,c}(v))[k] \in \mathbb{R}^d$ be the contextual embedding returned by M for the token at the k-th position of sequence $q^{e,c}(v)$. We define the unnormalized score $d(c[i], v)$ as:

$$d(c[i], v) = \cos(M(q^{e,c}(v))[i], M(q^{e,c}(v))[obj])$$

where $\cos(A, B)$ denotes the cosine similarity between two vectors, and obj denotes the position of object token set to v in sequence $q^{e,c}(v)$.

An exact computation of $z_{q,M}(c[i])$ would require $|V|$ forward passes. Instead, we propose to approximate $z_{q,M}(c[i])$. In practice, the language model's output distribution over the vocabulary has low entropy. Thus, we expect $p_{q,M}(v)$ to be zero for most $v \in V$. Therefore, we can approximate $z_{q,M}(c[i])$ by only summing over the top-k tokens for the probability mass $p_{q,M}$. We define a set of proposal tokens $\tilde{V}$ to be these top-k tokens. Empirically, we find that filtering out punctuation from $\tilde{V}$ also increases performance. We take the position of the anchor token in c, denoted by a_{out} to be:

$$a_{out} = \underset{i \in \{1,\ldots,|c|\}}{\arg\max} \sum_{v \in \tilde{V}} p_{q,M}(v) \cdot D(c[i], v)$$

This approximation only requires $k + 1$ forward passes to compute the final prediction. We examine the effect of setting different k in Appendix E.

Anchor token expansion We use a simple mechanism to expand the single-token anchor to a multi-token span. Given an off-the-shelf NER model, we do the following: if the anchor word is within a named entity, return the entire entity. Otherwise, return just the anchor word. While this approach allows us to support multi-token answers, its quality is highly correlated to that of the NER model. In practice, we do not find this to be a limiting factor because most entities tend to span few tokens. We experimentally evaluate the effect of using NER to obtain multi-token spans in Appendix E. We choose this approach as our focus is on studying if language models can be used directly for relation extraction and not decoding.

5 Experimental Evaluation

We compare RE-Flex against several competing relation extraction methods on four relation extraction benchmarks. The main points we seek to validate are: (1) how accurately can RE-Flex extract relations by utilizing contextual evidence, (2) how does RE-Flex compare to different categories of extractive models.

5.1 Experimental Setup

We describe the benchmarks, metrics, and methods we use in our evaluation. We discuss implementation details in Appendix D.

5.1.1 Datasets and Benchmarks

We consider four relation extraction benchmarks. The first two, T-REx (Elsahar et al., 2018) and Google-RE[1], are datasets previously used to evaluate unsupervised QA methods (Petroni et al., 2020), and are part of the LAMA probe (Petroni et al., 2019). We also consider the Zero-Shot Relation Extraction (ZSRE) benchmark (Levy et al., 2017), which is a dataset originally used to show that reading comprehension models can be extended to general relation extraction. Finally, we adapt the TAC Relation Extraction Dataset (TACRED) (Zhang et al., 2017a) to the slot filling setting utilizing a protocol similar to that used in Levy et al. (2017). We present the adaptation procedure, as well as a full table of benchmark characteristics in Appendix C. For the T-REx and Google-RE datasets *all* inputs correspond to entity-context pairs that contain a valid relation mention. On the other hand, ZSRE and TACRED contain invalid inputs for which the extraction models should return $\varnothing$. We refer to the first two datasets as the LAMA benchmarks, while the latter two are general relation extraction benchmarks.

5.1.2 Competing Methods

We consider three classes of competing methods: 1) models that rely on the generative ability of language models, 2) weakly-supervised QA models trained on an algorithmically aligned set of question-answer pairs, and 3) supervised QA models trained on human annotated question-answer pairs. We go into further details about these methods in Appendix D.

[1] `https://code.google.com/archive/p/relation-extraction-corpus/`

Generative Methods We compare to the naive cloze completion (NC) method of Petroni et al. (2019), which queries a masked language model to complete a cloze template representing a relation, without an associated context. We also consider the method of Petroni et al. (2020) (GD), which concatenates the context to the cloze template, and greedily decodes an answer to the relational query. This method is the same as that used in Radford et al. (2019) to show language models are unsupervised task learners. We use the RoBERTa language model (Liu et al., 2019) for both these baselines.

Weakly-supervised QA Methods We compare against two proposed weakly-supervised QA methods. The first method (Lewis et al., 2019) (UE-QA) uses a machine translation model to create questions from text using an off-the-shelf NER model, then trains a question answering head on the generated data to extract spans from text. The second method (Dhingra et al., 2018) (SE-QA) is a semi-supervised approach to QA. It also uses an NER model to generate cloze-style question-answer pairs and then trains a QA model on these pairs. Authors provide generated data for both methods, which we use to train a BERT-Large QA model (Devlin et al., 2018).

Supervised QA Methods Finally, we compare against three supervised QA models trained on human-annotated question-answer pairs. We train BiDAF (Seo et al., 2016), extended to be able to predict no answer (Levy et al., 2017) on Squad 2.0 (Rajpurkar et al., 2018). Additionally, we train BERT-Large on Squad 2.0 (B-Squad) and the training set of ZSRE (B-ZSRE). For Google-RE and T-REx, we do not allow these models to return $\varnothing$.

5.1.3 Metrics

We follow standard metrics from Squad 1.0 (Rajpurkar et al., 2016) and evaluate the quality of each extraction using two metrics: *Exact Match (EM)* and F_1-score. Exact match assigns a score of 1.0 when the extracted span matches exactly the ground truth span, or 0.0 otherwise. F_1 treats the extracted span as a set and considers the precision and recall (at the token level) between tokens in the predicted span and tokens in the ground truth span. For each relation, we compute the average EM and F_1 scores and then average these scores across relations. In the last step, we do not assign weights to different relations.

Category	Dataset	Metric	RE-Flex	Generative		Weakly-supervised		Supervised		
				NC	GD	UE-QA	SE-QA	BiDAF	B-Squad	B-ZSRE
NCR	T-REx	EM	**56.0**	22.9	43.2	14.2	21.2	35.5	42.2	46.2
		F_1	**56.0**	22.9	43.7	19.3	28.2	46.6	52.0	52.7
	Google-RE	EM	**87.2**	5.6	75.6	51.9	19.7	53.9	52.9	70.5
		F_1	**87.2**	5.6	75.7	65.0	25.3	74.8	76.5	75.8
CR	ZSRE	EM	43.7	14.1	27.7	16.7	17.3	40.2	66.5	**82.0**
		F_1	46.9	14.9	30.7	23.7	24.9	49.0	74.1	**84.8**
	TACRED	EM	49.4	9.0	25.6	17.5	13.9	49.3	**56.9**	54.4
		F_1	50.1	9.0	26.3	22.0	18.6	53.7	**61.1**	55.3

Table 1: Performance for all methods on the four benchmarks. Datasets are divided into two categories, no context rejection (NCR) and context rejection (CR), denoting whether the dataset requires that ∅ be returned for any examples. Bold values highlight the best method. Red and blue values denote worse or better performance than RE-Flex, respectively.

5.2 End-to-end Comparisons

We evaluate the performance of RE-Flex against all competing methods for different benchmarks. The results are shown in Table 1.

Method	ZSRE		TACRED	
	EM	F_1	EM	F_1
No rejection	40.0	43.4	39.1	39.5
With rejection	43.7	46.9	49.4	50.1

Table 2: Context rejection ablation on ZSRE.

5.2.1 LAMA Benchmarks

We focus on the LAMA benchmarks, which consist of the Google-RE and T-REx datasets. For these benchmarks, the context always contains the answer to the relational query, and the answer is a single token. We analyze the performance of RE-Flex against each group of baselines.

Comparison to Generative Methods We first compare the performance of RE-Flex to that of the generative methods NC and GD. We see that RE-Flex outperforms NC by 33.1 F_1 on T-REx and 81.6 F_1 on Google-RE. We see that GD also outperforms NC. This observation suggests that retrieving relevant contexts and associating them with relational queries significantly increases the performance of generative relation extraction methods, as opposed to relying on the model's memory.

Compared to GD, RE-Flex shows an improvement of 12.3 F_1 on T-REx and 12.3 F_1 on Google-RE. We attribute this gain on RE-Flex's ability to constrain the language model's generation to tokens only present in the context.

Takeaway: Restricting language model inference ensures more factual predictions, and is key to accurate relation extraction when using the contextual language model directly.

Comparison to Weakly-supervised Methods We compare RE-Flex to UE-QA and SE-QA, which both construct a weakly-aligned noisy training dataset and fine-tune an extractive QA head on the produced examples. RE-Flex outperforms both approaches, yielding improvements of 27.8 F_1 on T-REx and 22.2 F_1 on Google-RE compared to the best performing method for each dataset.

Additionally, we see that, on these benchmarks, GD (despite yielding worse results than RE-Flex) also outperforms UE-QA and SE-QA. This result suggests that training on noisy training data can severely hamper downstream performance.

Takeaway: Using weak-alignment to train a QA head often leads to poor results, and it is better to use the model's generative ability instead. Below, we show that this behavior extends to general relation extraction benchmarks.

Comparison to Supervised Methods We find the surprising result that RE-Flex is better than all supervised methods. We believe the results can be attributed to the fact that the language model is able to capture the subset of relations in these datasets quite well. This finding is also supported by the fact that GD also yields comparable accuracy to the supervised methods.

As we examine below, this behavior is not as pronounced when considering the general relation extraction setting. Still, we are able to assert that for specific relation subsets, our inference procedure is able to outperform standard QA models.

Takeaway: Our findings strongly support that contextual models capture certain semantic relations (Petroni et al., 2019, 2020), but to outperform the performance of supervised models we still need RE-Flex's fine-tuned inference procedure.

5.2.2 General Relation Benchmarks

We now focus on ZSRE and TACRED, which are more reflective of our problem statement. Here, we must assert whether a candidate context contains a true expression of the relation, and produce multiple token spans as answers.

Comparison to Generative and Weakly-supervised Methods We see that RE-Flex significantly outperforms all generative and weakly-supervised methods on these benchmarks. We outperform the next best method by 22.0 F_1 on ZSRE and by 28.1 F_1 on TACRED. In this realistic context, using the contextual language model without fine-tuning the corresponding inferences falls short, while a noisily trained QA head also exhibits poor performance. To understand if these results are to be attributed to RE-Flex's ability to reject contexts, we ablate the performance of RE-Flex with and without enabling context rejection (Section 4.1). The results are shown in Figure 2. We see that context rejection leads to increased performance. For example, in TACRED it boosts RE-Flex's F_1 score by more than 10 points. We also see that even without the context rejection, RE-Flex outperforms these classes of methods by up to 13.2 F_1 compared to the next best method. This finding suggests that the combination of fine-tuned inference and context rejection leads to good performance.

Takeaway: In addition to restricted inference, incorporating context rejection is necessary for the general relation extraction setting. This finding is consistent with that for the LAMA benchmarks.

Comparison to Supervised Methods We compare to supervised QA baselines on the general relation extraction benchmarks. Here, all competing approaches are trained on human annotated QA pairs. We find that RE-Flex performs comparably to BiDAF but falls short of the fine-tuned BERT-based QA models. Recall that BiDAF relies on a simpler attention-flow model, and does not use self-supervised language representations, as BERT does. The best performing BERT baselines see an average improvement of 37.9 F_1 on ZSRE and 10 F_1 on TACRED compared to RE-Flex. However, as we show next, there is a significant number of relations for which RE-Flex outperforms the BERT-based baselines for even up to 40 F_1 points in TACRED and up to 60 F_1 points in ZSRE.

To better understand RE-Flex's behavior beyond the averaged F_1, we record the difference in F_1 scores between RE-Flex and each BERT baseline on a per relation basis. Histograms of these results can be found in Figure 2. On TACRED, RE-Flex outperforms the best method for 20% of relations and comes within 20.0 F_1 for 26% of relations. For ZSRE, RE-Flex outperforms the best method for 6% of relations, and comes within 20.0 F_1 for another 12% of relations. These results show that for certain relations, RE-Flex can perform competitively or even better with supervised methods. We note that the relations which RE-Flex performs better than the baselines tend to be simple many-to-one relations which are likely to be clearly stated in succinct ways. For example, RE-Flex outperforms baselines on the `cause_of_death` and `religious_affiliation` relations.

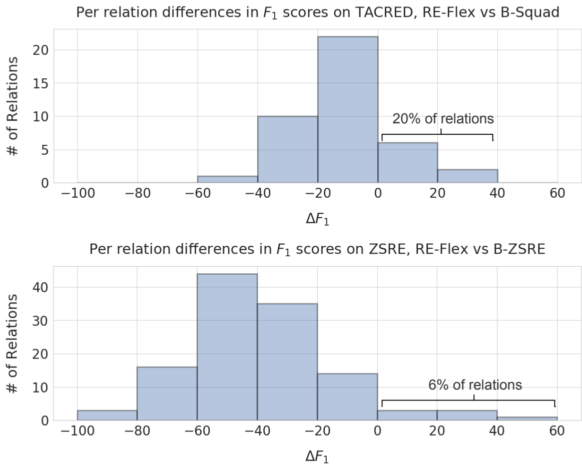

Figure 2: Histogram breakdown of differences between F_1 performances between RE-Flex and the best performing supervised methods for each of the ZSRE and TACRED benchmarks. We see that for many cases the unsupervised approach of RE-Flex outperforms the fully-supervised BERT-based baselines.

Finally, we note the performance drop of B-ZSRE when applied to the TACRED dataset. Both QA models perform similarly on TACRED, which does not have a QA training set associated with it. This shows that supervised QA models exhibit some bias towards the underlying corpus they are trained on, which supports claims in previous work (Dhingra et al., 2018). We further expand on this result in Appendix F.

Takeaway: We find a significant number of relations for which RE-Flex outperforms even fully-supervised QA models. These relations tend to be simple many-to-one relations, where the fact is clearly expressed in the context.

6 Conclusion

We introduced RE-Flex, a simple framework that constrains the inference of self-supervised language models after they have been trained. We perform an extensive experimental study over multiple relation extraction benchmarks and demonstrate that RE-Flex outperforms competing relation extraction methods by up to 27.8 F_1 points compared to the next-best unsupervised method.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A Latent Variable Model Approach to PMI-based Word Embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Ijcai*, volume 7, pages 2670–2676.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. *arXiv:1704.00051 [cs]*. ArXiv: 1704.00051.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. *arXiv:1810.05682 [cs]*. ArXiv: 1810.05682.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and Effective Semi-Supervised Question Answering. 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 582–587, New Orleans, Louisiana. Association for Computational Linguistics.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 541–550, Stroudsburg, PA, USA. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised Question Answering by Cloze Translation. *arXiv:1906.04980 [cs]*. ArXiv: 1906.04980.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. *arXiv:1905.05529 [cs]*. ArXiv: 1905.05529.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *arXiv:1704.04368 [cs]*. ArXiv: 1704.04368.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. 2015. Incremental knowledge base construction using deepdive. *Proceedings of the VLDB Endowment*, 8(11):1310–1321.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. 1995. Crystal inducing a conceptual dictionary. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1314–1319, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264, Florence, Italy. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017a. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017b. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Jiawei Zhou and Alexander Rush. 2019. Simple Unsupervised Summarization by Contextual Matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy. Association for Computational Linguistics.

A Implementation Details

We set RE-Flex's top-k parameter (see Section 4.2) to 16. We tune the λ parameter, when applicable, on the provided development sets of the datasets. Additionally, we tune whether to use the NER expansion, again using the development sets of the datasets. These hyperparameters are tuned using a standard grid search. We use Fairseq's implementation of RoBERTa-large[2] as our self-supervised language model. For the embeddings of the context rejection mechanism, we use the FastText library (Bojanowski et al., 2017). For the token embeddings of the anchor identification model, we first collect an embedding for each subword (RoBERTa uses byte-pair subword encodings) by flattening the output representation of all of the RoBERTa-large decoder layers for each subword into a single vector. Because we operate on the token and not the subword level, we obtain a token representation by averaging all subword vector embeddings that compose a token. Examining the effect of our embedding choices is out of the scope of this work, and we leave it as a future analysis.

As stated in our construction of $\tilde{B}$ (Section 4.2), we filter any punctuation predicted. For named entity recognition and noun phrase chunking (used for identifying multi-token extractions in RE-Flex), we use the `en_web_core_lg` model of the spaCy library[3]. We train and run all models on a single NVIDIA V100 32GB memory GPU.

B Qualitative Results

We provide few qualitative examples of extractions from ZSRE obtained by the different methods. The examples are shown in Figure 3. The first two examples highlight two accurate extractions from RE-Flex, while the third example an incorrect extraction. These examples also highlight the weakness of the UE-QA and SE-QA methods: many times they extract an incorrect large sequence from the input context.

C Dataset Details

TACRED adaptation to slot filling Relation Extraction Dataset (Zhang et al., 2017b) is a relation classification dataset. The original task is to predict the relation of a subject and object pair given a supporting context. There are 41 possible relations, with an additional relation labelled "no_relation" to denote an example whose sentence does not express the relation between the subject and object. We convert the dataset to our slot filling setting by considering the subject and relation known for each example, and setting the task to predict the object. Following the established process of Levy et al. (2017) for adding realistic negative examples, we distribute all examples labelled `no_relation` to relations sharing the same head entity, and set the target object for each to be $\varnothing$.

Dataset characteristics A table of dataset characteristics can be found in Table 3.

D Competing Methods Implementation Details

All generative baselines are implemented using Fairseq (Ott et al., 2019). Following the implementation of (Lewis et al., 2019), we train a BERT-Large model on the provided training datasets of (Lewis et al., 2019) and (Dhingra et al., 2018) for the UE-QA and SE-QA baselines. These training

R: production company(Lawless Range, ?)
T: Lawless Range was produced by [Y]
Q: Which production company is involved with lawless range
C: Lawless Range is a 1935 Western film released by Republic Pictures, directed by Robert N. Bradbury and starring John Wayne.
RE-Flex: Republic Pictures
UE-QA: Republic Pictures, directed by Robert N. Bradbury
SE-QA: 1935 Western film released by Republic Pictures, directed by Robert N. Bradbury and starring John Wayne.
B-Squad: Republic Pictures,
B-ZSRE: Republic Pictures,

R: military branch(David Semple, ?)
T: David Semple served in the [Y]
Q: Who did David Semple serve for?
C: Lieutenant-Colonel Sir David Semple MD (1856 -- 1937) was a British Army officer who founded the Pasteur Institute at Kasauli in the Indian state of Himachal Pradesh.
RE-Flex: British Army
UE-QA: the Pasteur Institute at Kasauli in the Indian state of Himachal Pradesh.
SE-QA: Himachal Pradesh.
B-Squad: British Army
B-ZSRE: Pasteur

R: publisher(FIFA Soccer 95, ?)
T: FIFA Soccer 95 is published by [Y]
Q: What company published FIFA Soccer 95?
C: FIFA Soccer 95 is a 1994 sports video game developed by EA Canada's Extended Play Productions team and published by Electronic Arts.
RE-Flex: EA Canada's
UE-QA: Extended Play Productions
SE-QA: Electronic Arts.
B-Squad: Electronic Arts.
B-ZSRE: Electronic Arts.

Figure 3: Example extractions from ZSRE for the different methods. Here, **R** indicates the target relation, **T** the cloze template used, **Q** the corresponding question required by the QA-based models, and **C** the provided context for the extraction task.

[2] https://github.com/pytorch/fairseq
[3] https://spacy.io/

Benchmark	Relation in Context	Extraction Type	Underlying Corpus	# of Relations	Total Samples
T-REx	Implicit Relation Mention	Single-Token	Wikipedia	41	34,039
Google-RE	Exact Relation Mention	Single-Token	Wikipedia	3	5,528
ZSRE	Possibly Irrelevant Context	Multi-Token	Wikipedia	120	42,635
TACRED	Possibly Irrelevant Context	Multi-Token	TAC-KBP	41	6,357

Table 3: We consider four benchmarks that vary with respect to the type of target extractions, the quality of context to relation alignment, and the underlying corpus.

Method	ZSRE		TACRED	
	EM	F_1	EM	F_1
No expansion	42.4	46.1	49.6	50.3
NER expansion	36.4	39.2	42.9	43.3
Tuned expansion	43.7	46.9	49.4	50.1

Table 4: Effect of token expansion on ZSRE dataset.

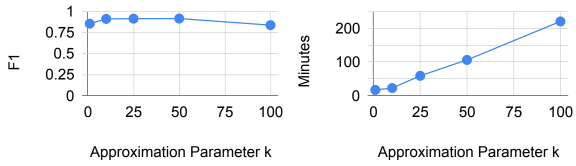

Figure 4: Effect of parameter k on F_1 for Google-RE.

datasets are collected over a snapshot of Wikipedia, which is the underlying corpus of three of our four benchmarks. We use the HuggingFace Transformers library (Wolf et al., 2019) for our implementation of all QA models except BiDAF, for which we use a slightly altered version of the original author's code (Levy et al., 2017).

E Microbenchmarks

We evaluate the effect of different components of RE-Flex on its end-to-end performance.

Context rejection analysis We first examine the effect of RE-Flex's context rejection mechanism. In Table 2, we measure the performance with and without context rejection on the datasets which require context rejection. We find that on the ZSRE dataset, the rejection increases F_1 by 3.5. On TACRED, F_1 increases by 10.6 F_1 with context rejection. In both cases, context rejection positively impacts performance.

Anchor expansion analysis We examine the effect of expanding the anchor token in RE-Flex. To examine this behavior in more details, we evaluate RE-Flex by considering single-token only extractions, multi-token extractions using NER expansion, and a tuned expansion that chooses either to expand or not to expand based on performance on the development set for each dataset. The results are shown in Table 4. We see that with tuned expansion, F_1 increases by 0.8 F_1 on ZSRE, and decreases by 0.2 F_1 on TACRED. In fact, utilizing NER expansion for all relations leads to a decrease of 6.9 F_1 on ZSRE and 7.0 F_1 on TACRED. We conclude that what additional information to include in a prediction is determined by the information need of each relation, and meeting this need for general relations is left for future work.

Approximation analysis We examine the trade-offs between performance, runtime, and the approximation parameter k described in Section 4.2. We set the batch size to 1 to for this analysis. Results for the three Google-RE relations are shown in Figure 4. Our measurements show that our choice of $k = 16$ leads to high-quality results while having an acceptable runtime.

F Biases of QA Models

Given that RE-Flex outperforms all supervised methods for T-Rex and Google-RE, we perform a detailed analysis to understand the reason behind this limitation of QA models. We suspect these results can be partially attributed to the construction of these settings, where the expected response is a single token; however QA models are more likely to predict multi-token spans because their training data is biased towards longer spans.

We have the following finding from our results: B-ZSRE, which is trained on entity length answer spans, performs better than the B-Squad baseline by 17.6 EM. As both models are the exact same architecture, but trained on different QA datasets, we can attribute this difference to biases in span length. We further verify this span length bias by conducting an error breakdown on these datasets. For each QA model, we consider each example

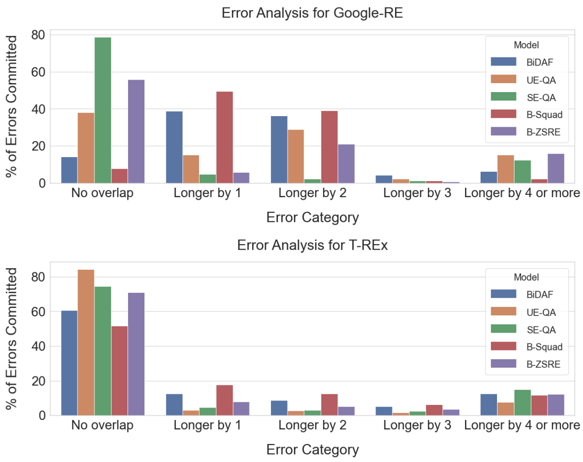

Figure 5: Error categorization of QA-based methods. For RE-Flex all errors belong to the No overlap group.

which returns an EM of 0, and classify the example based on whether the predict has no overlap with the ground truth, or by how much longer the prediction is.

We present the results in Figure 5. We see that on Google-RE, the majority of the errors committed by BiDAF and B-Squad, both trained on Squad 2.0, are because the predictions are longer than the expected answer by one or two tokens. B-ZSRE does not exhibit these error ratios, instead primarily missing the answer entirely. On T-REx, all models primarily miss the ground truth entirely. We attribute this finding to the fact that evidence in T-REx is weaker and does not have explicit lexical clues to select answer spans. Training these models using contexts with weaker evidence might improve relation extraction performance.

Takeaway: Supervised QA models are biased towards the span lengths in their training set, and struggle when given weaker evidence contexts.