A APPENDIX

A.1 STATISTICS OF BENCHMARK DATA SETS

In this section, we summarize the statistics of the four natural language processing datasets used in our experiments in Table 2. For SQuAD, since the dataset does not annotate supporting facts, we approximately estimate supporting facts by counting tokens that are part of a bigram that appears in the question.

Data	Train	Dev	# of Spans	# of Supporting facts
HotpotQA(Yang et al., 2018)†	90,564	7,405	14.45/11/17/2/96◇	2.38/2/3/2/12
MultiRC(Khashabi et al., 2018)†	5,131	953	14.72/12/18/6/41	2.32/2/2/2/6
DocRED(Yao et al., 2019)†	38,180	12,323	8.24/6/10/3/25	1.67/1/2/1/11
SQuAD(Rajpurkar et al., 2016)⋆	87,599	10,570	156.26/114/186/25/853	7.74/3/11/0/82‡

† Sentence-level span and supporting facts, and MultiRC is the second version and available at https://cogcomp.seas.upenn.edu/multirc/, where the column "train" and "dev" w.r.t. MultiRC reports the number of questions in training data and dev data, respectively.
⋆ Token-level span, and the statistics of # of spans/supporting facts are collected by setting span = 1 token;
◇ This is collected based on the top 4 documents selected from 10 candidate documents by document retriever. The five numbers correspond to Average, 25% Percentile, 75% Percentile, Min and Max, respectively.
‡ We define the supporting facts as the spans of the context which have bigrams appearing in the question.

Table 2: Statistics of Benchmark Data sets

A.2 EXPERIMENTAL SETUP FOR DIFFERENT TASKS

This section introduces the detailed implementations of our methods on four benchmark data sets, as well as the hyper-parameter setting for model optimization and baselines to compare with our implementations.

A.2.1 HOTPOTQA

Implementation details. The objective of HotpotQA is answering questions from a set of 10 paragraphs where two paragraphs are relevant to the question and the rest are distractors. HotpotQA presents two tasks: answer span prediction and evidence sentence (i.e., supporting fact) prediction. Our HotpotQA model consists of two stages: the first stage selects top 4 paragraphs from 10 candidates by a retrieval model. The second stage finds the final answer span and evidence over the selected 4 paragraphs. We particularly feed the following input format to encoder: "[CLS] question [SEP] $\text{sent}_{1,1}$ [SEP] $\text{sent}_{1,2} \cdots$ [SEP] $\text{sent}_{4,1}$ [SEP] $\text{sent}_{4,2} \cdots$ [SEP]". And we apply the proposed span drop methods over all the sentences except for supporting facts.

For answer span prediction, we use the answer span prediction model in (Devlin et al., 2019) with an additional task of question type (yes/no/span) classification head over the first special token ([CLS]). For evidence extraction, we apply two-layer MLPs on top of the representations corresponding to sentence and paragraph to get the corresponding evidence prediction scores and use binary cross entropy loss to train the model. Finally, we combine answer span, question type, sentence evidence, and relevant paragraph losses and train the model in a multitask way using linear combination of losses. The hyper-parameter search space for our models on HotpotQA is given in Table 3.

Baselines. We compare our implementation of HotpotQA model over Electra with the following strong baselines: 1) RoBERTa (Liu et al., 2019b) based model, 2) long sequence encoder Longformer (Beltagy et al., 2020) based model and 3) SAE (Tu et al., 2020) which combines graph neural network and pretrained language models for multi-hop question answer and is the current SOTA model on HotpotQA. The numbers reported in our paper for the first two models come from (Zaheer et al., 2020).

A.2.2 MULTIRC

Implementation details. MultiRC is a multi-choice question answer task, which supplies a set of alternatives or possible answers to the question and requires to select the best answer(s) based on

Parameter name	HotpotQA	MultiRC	DocRED	SQuAD
Batch size	{4, 8}	{8, 16}	{4, 8}	{16, 32}
Learning rate	{3e-5, 2e-5, 1e-5, 1e-4}	{3e-5, 2e-5, 1e-5, 1e-4}	{1e-4, 5e-5}	{1e-4, 5e-5}
Span Drop ratio	{0.1, 0.15, 0.2, 0.25}	{0.1, 0.15, 0.2, 0.25}	{0.05, 0.1, 0.15, 0.2}	{0.03, 0.05, 0.1, 0.15, 0.2, 0.25}
Optimizer	AdamW	AdamW	AdamW	AdamW
Epochs	10	15	{10,20,30}	{2,4,6,8}

Table 3: Hyper-parameter search space for models on different benchmarks

multiple sentences while a few of these sentences (i.e., supporting facts) are relevant to the questions. For given example of k candidate answer and n sentences, we first feed the following input to the encoder: "[CLS] question [SEP] answer$_1$ [SEP] answer$_2$ [SEP] $\cdots$ answer$_k$ [SEP] sentence$_1$ [SEP] sentence$_2$ [SEP] $\cdots$ sentence$_n$ [SEP]". For answer prediction, we apply two-layer MLPs on top of the representations corresponding to candidate answers and sentences to get the corresponding answer and sentences scores, and use a combination of two binary cross entropy losses to train the model in a multi-task way. We apply our span drop methods over all input sentences except for the evidence sentences. The hyper-parameter search space for our MultiRC models is given in Table 3.

Baselines. We compare our implementation of MultiRC model over Electra with the following baselines: 1) BERT based model, 2) RoBERTa (Liu et al., 2019b) based model and 3) REPT (RoBERTa-base) trained with new additional training tasks (Jiao et al., 2021).

A.2.3 Relation extraction task: DocRED

Implementation details. DocRED is document-level relation extraction which consists of two tasks: relation prediction of a given pair of entities and the evidence prediction. We construct the entity-guided inputs to the encoder following prior work Huang et al. (2020). Each training example is organized by concatenating the head entity, together with the n sentences in the document as "[CLS] head entity [SEP] sentence$_1$ [SEP] sentence$_2$ [SEP] $\cdots$ sentence$_n$ [SEP]".

For both relation extraction and evidence prediction, we apply a biaffine transformation that combines entity representations and entity/sentence representations, respectively, and score them using the adaptive threshold loss proposed by Zhou et al. (2021b). We train the model in a multi-task setting by using a linear combination of relation extraction and evidence prediction losses. And we apply the proposed span drop methods over all input sentences except for those that serve as evidence for entity relations with the head entity. Please refer to Table 3 for the hyper-parameter search space of our DocRED models.

Baselines. We compare against a set of strong baselines w.r.t. document-level relation including: 1) E2GRE (Huang et al., 2020), 2) ATLOP (Zhou et al., 2021b) and 3) SSAN (Xu et al., 2021).

A.2.4 SQuAD

Implementation details. SQuAD aims at extracting a segment of text from a given paragraph as answer to the question. We format the input example as "[CLS] question [SEP] paragraph [SEP]" and feed to the encoder. Following prior work, we employ a span prediction mechanism where the end of the span is conditioned on the representation of the start of the span, originally presented in the XLNet paper (Yang et al., 2019). And we apply our proposed span drop methods over the paragraph before feeding it into the model for training, where spans that contain question bigrams are preserved. The hyper-parameter space to optimize the SQuAD model is shown in Table 3.

Baselines. We implemented models for SQuAD based on ELECTRA (Clark et al., 2019) and compare them with existing model implementations for SQuAD based on 1) ELECTRA (Clark et al., 2019), 2) RoBERTa (Liu et al., 2019b) and 3) XLNet (Yang et al., 2019), respectively.