Rethinking Label Smoothing on Multi-hop Question Answering

Zhangyue Yin[1,2*], **Yuxin Wang**[1,2*], **Yiguang Wu**[1,2], **Hang Yan**[1,2], **Xiannian Hu**[1,2], **Xinyu Zhang**[3], **Zhao Cao**[3], **Xuanjing Huang**[1,2], **Xipeng Qiu**[1,2 †]
[1]School of Computer Science, Fudan University
[2]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
[3]Huawei Poisson Lab
{yinzy21,wangyuxin21,xnhu21}@m.fudan.edu.cn
{ygwu20,hyan19,xjhuang,xpqiu}@fudan.edu.cn
{zhangxinyu35,caozhao1}@huawei.com

Abstract

Label smoothing is a regularization technique widely used in supervised learning to improve the generalization of models on various tasks, such as image classification and machine translation. However, the effectiveness of label smoothing in multi-hop question answering (MHQA) has yet to be well studied. In this paper, we systematically analyze the role of label smoothing on various modules of MHQA and propose F1 smoothing, a novel label smoothing technique specifically designed for machine reading comprehension (MRC) tasks. We evaluate our method on the HotpotQA dataset and demonstrate its superiority over several strong baselines, including models that utilize complex attention mechanisms. Our results suggest that label smoothing can be effective in MHQA, but the choice of smoothing strategy can significantly affect performance.

1 Introduction

Label smoothing is a regularization technique that has been widely used in supervised learning to improve the generalization of models on various tasks, such as image classification (He et al., 2020b) and machine translation (Gao et al., 2020; Lukasik et al., 2020b). The basic idea of label smoothing is to smooth the distribution of true labels by replacing the one-hot encoding of the labels with a softened version (Szegedy et al., 2016). This encourages the model to be less confident in its predictions and to consider a wider range of possibilities, which can reduce overfitting and improve generalization (Müller et al., 2019; Lukasik et al., 2020a).

Multi-hop question answering (MHQA) is a task that involves answering complex questions by aggregating information from multiple sources. These tasks require a model to perform multiple reasoning steps and to handle varied structures of information. Mainstream QA models for MHQA often consist of a complex pipeline, including a document retriever, a supporting evidence selector, and a module for multi-hop reasoning (Tu et al., 2020; Wu et al., 2021a; Li et al., 2022b). These components work together to accurately retrieve and integrate relevant information from multiple sources in order to provide a correct answer to the given question.

Despite the widespread use of label smoothing in other tasks, the effectiveness of this technique in MHQA has not to be thoroughly investigated. In this paper, we aim to fill this research gap by systematically analyzing the role of label smoothing on various modules of MHQA. We will conduct experiments using various label smoothing strategies and multiple label smoothing techniques, including F1 smoothing, a novel method we propose for machine reading comprehension (MRC) tasks, and evaluate its performance on the HotpotQA dataset (Yang et al., 2018a).

To the best of our knowledge, we are the first to systematically study the effect of label smoothing on MHQA. Our experiments demonstrate that carefully designing label smoothing for each module of MHQA and using the appropriate label smoothing strategy can significantly improve the performance of each module and lead to overall improvements in the model. The code for our approach is available on GitHub[1].

Our main contributions are as follows:

- We conduct a systematic analysis of the impact of label smoothing on various modules of MHQA, including document retrieval, supporting evidence prediction, answer type selection and answer span extraction.
- We propose F1 smoothing, a new label

[*]Equal contribution.
[†]Corresponding author.

[1]https://github.com/yinzhangyue/C2FM

smoothing method that is specifically tailored for MRC tasks.

- We evaluate the proposed method on the HotpotQA dataset and find that it outperforms several strong baseline models and achieves the best results. This demonstrates the effectiveness of our smart label smoothing design in MHQA.

2 Related Work

Label Smoothing Label smoothing is a regularization technique that was introduced in computer vision to improve classification accuracy on ImageNet (Szegedy et al., 2016). The idea behind label smoothing is to prevent the model from becoming too confident in its predictions by slightly modifying the ground truth labels during training. This helps to improve the generalization of the model. Label smoothing has been widely adopted in a variety of natural language processing tasks, including speech recognition (Chorowski and Jaitly, 2017), document retrieval (Penha and Hauff, 2021), dialogue generation (Saha et al., 2021) and neural machine translation (Gao et al., 2020; Lukasik et al., 2020b; Graça et al., 2019). In recent years, label smoothing has also been applied to span-related tasks such as machine reading comprehension and named entity recognition (Zhao et al., 2020a; Zhu and Li, 2022).

Multi-hop Question Answering Multi-hop reading comprehension (MHRC) is a challenging task in the field of machine reading comprehension (MRC) that closely resembles the human thought process in real-world scenarios. As a result, it has become a popular topic in the field of natural language understanding in recent years. To facilitate research in this area, several datasets have been developed, including HotpotQA (Yang et al., 2018a), WikiHop (Welbl et al., 2018), and NarrativeQA (Kočiský et al., 2018). Among these, HotpotQA is particularly representative and challenging, as it not only requires the model to extract the correct answer span from the context but also requires a series of supporting evidence as proof of MHRC. In this paper, we focus on HotpotQA as our primary dataset for studying label smoothing in the context of MHRC.

Recent advances in MHRC have led to the development of several graph-free models, such as QUARK (Groeneveld et al., 2020), C2FReader (Shao et al., 2020), and S2G (Wu et al., 2021b), which have challenged the dominance of previous graph-based approaches like DFGN (Qiu et al., 2019), SAE (Tu et al., 2020), and HGN (Fang et al., 2020). C2FReader (Shao et al., 2020) suggests that the performance difference between graphical attention and self-attention is minimal, while S2G's (Wu et al., 2021b) strong performance demonstrates the potential of not using graphical modeling in MHRC. FE2H (Li et al., 2022a), which uses a two-stage selector and a multi-task reader, currently achieves the best performance on the MHRC task, indicating that pre-trained language models alone may be sufficient for modeling multi-hop reasoning. Motivated by the design of S2G (Wu et al., 2021b) and FE2H (Li et al., 2022a), we propose a simpler model called C2FM that does not include an additional attention module. Our aim is to examine the impact of label smoothing on individual models of MHQA.

3 Our Framework

We first introduce our multi-hop question answering architecture to facilitate our description of label smoothing. Our framework uses a **C**oarse-to-**F**ine retriever and a **M**ulti-task prediction reader (**C2FM**) for answer extraction and supporting evidences prediction. Compared to the complex structural design of S2G (Wu et al., 2021a), the simpler design of C2FM is more conducive for us to investigate the role of label smoothing in each component of the framework.

Figure 1 illustrates the overall framework of C2FM. In the document retrieval module, we use a coarse-to-fine retrieval approach to identify the most relevant documents for a given question. In this example, Doc1 and Doc4 are marked as relevant documents. In the coarse stage, the top k documents (which we set to 3 in this paper) are retrieved, resulting in Doc1, Doc3, and Doc4 as the most likely to contain the answer. In the fine-grained stage, we combine documents two by two and retrieve them based on their relationship to each other, which is crucial for multi-hop QA. Doc1 and Doc4 are the correct combinations that we want to pass on to the downstream multi-task reading module, hence the label 1. This two-stage retrieval process allows our model to capture not only the relationship between the question and the candidate documents, but also the relationship between the most promising candidate documents.

In the reading comprehension module, we use

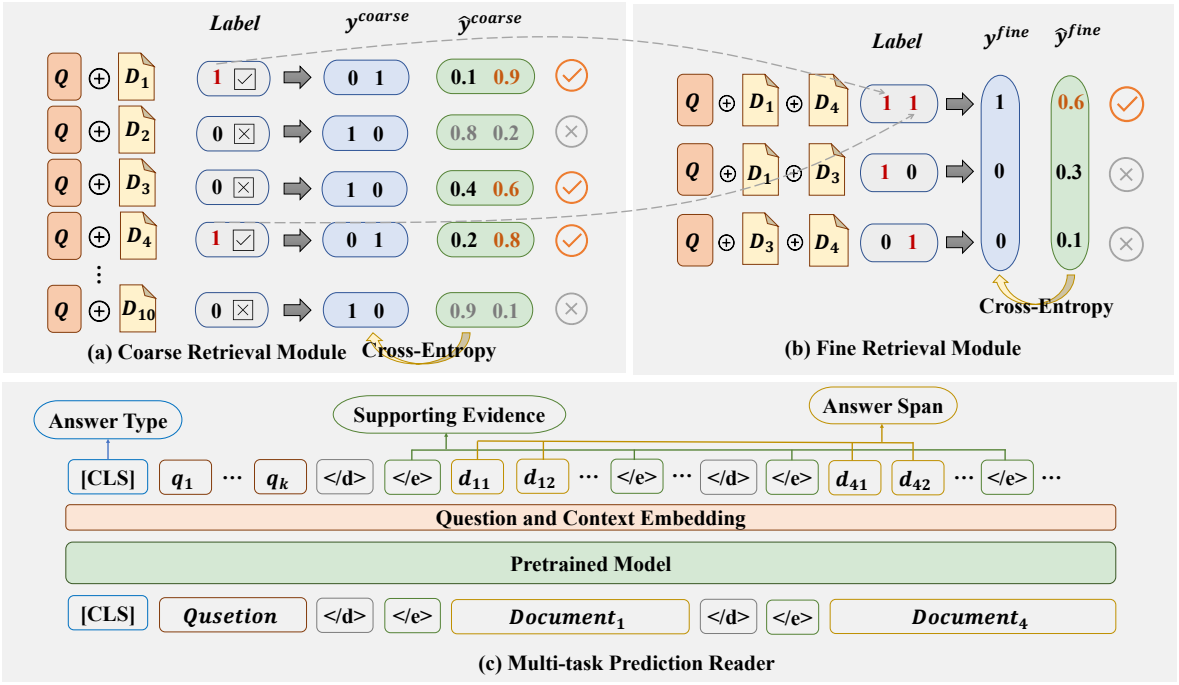

Figure 1: Overview of our C2FM model, which consists of three main modules: Coarse Retrieval, Fine Retrieval, and the Multi-Task Reader.

a multi-task learning approach to simultaneously optimize three goals: answer type selection, answer span extraction, and evidence sentence prediction.

3.1 Coarse-to-Fine Retrieval

Coarse Retrieval Module In the formal MHRC task, each question Q is typically provided with a set of m documents $\{D_1, D_2 \ldots, D_m\}$, only a few of which (two in HotpotQA) are truly relevant to Q. In coarse retrieval, we optimize the classification of each combination of question and document using the Cross-Entropy loss. Specifically, we treat every document D_i as a two-class classification problem.

$$\mathcal{L}_{coarse} = \mathbb{E}[-\frac{1}{M}\sum_{i=1}^{M}(y_i^{coarse} \cdot log(\hat{y}_i^{coarse}) + (1 - y_i^{coarse}) \cdot log(1 - \hat{y}_i^{coarse}))] \quad (1)$$

where $\hat{y}_i^{coarse}$ is the probability predicted by the model and y_i^{coarse} is the one-hot encoded ground-truth distribution.

$$y_i^{coarse} = \begin{cases} 1 & D_i \text{ is a related document.} \\ 0 & D_i \text{ is an unrelated document.} \end{cases} \quad (2)$$

Fine Retrieval Module In fine-grained retrieval, we select the top three relevant articles from the previous step and combine them in pairs to create a set of document pairs $\{C_1, C_2, C_3\}$ with C_3^2 total combinations. We then focus on the interactions between these document pairs, which are essential for multi-hop question answering, and optimize using cross-entropy loss.

$$\mathcal{L}_{fine} = \mathbb{E}[-\sum_{i=1}^{3} y_i^{fine} log(\hat{y}_i^{fine})] \quad (3)$$

We use $\hat{y}_i^{fine}$ to represent the document pair probability predicted by our model and yi^{fine} to represent the one-hot encoded ground-truth distribution.

$$y_i^{fine} = \begin{cases} 1 & C_i \text{ consists of two related documents.} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Thus, when the coarse retrieval fails to retrieve two related documents, y_i^{fine} is **all zero**, indicating that it does not contribute to the model's performance until the coarse retrieval is sufficient. We use a single pre-trained language model as the encoder for both the coarse and fine retrieval steps, and combine the retrieval losses using a weighted sum. In our paper, both λ_1 and λ_2 are set to 1, indicating that the fine retrieval contributes equally to the model's performance as the coarse retrieval

$$\mathcal{L}_{retrieval} = \lambda_1 \mathcal{L}_{coarse} + \lambda_2 \mathcal{L}_{fine}. \quad (5)$$

3.2 Multi-Task Reader

In the reading comprehension module, we use multi-task learning to simultaneously predict Evidence Sentences and extract answer span. In order to better evaluate the role of label smoothing, we do not include an additional attention module in our model. Instead, we focus on the effects of label smoothing on the performance of the main reading comprehension module. In addition, HotpotQA contains samples with yes/no answers. The practice of splicing "yes" and "no" tokens at the beginning of the sequence (Li et al., 2022a) could corrupt the original text's semantic information. To avoid the impact of additional information on label smoothing analysis, we introduce an answer type selection header trained with a cross-entropy loss function.

$$\mathcal{L}_{type} = \mathbb{E}[-\sum_{i=1}^{3} y_i^{type} log(\hat{y}_i^{type})] \quad (6)$$

where $\hat{y}i^{fine}$ denotes the predicted probability distribution over answer types generated by our model, and yi^{fine} represents the corresponding one-hot encoded ground-truth distribution.

$$y_i^{type} = \begin{cases} 0 & \text{answer is no} \\ 1 & \text{answer is yes} \\ 2 & \text{answer is a span} \end{cases} \quad (7)$$

To extract the span of answers, we use a linear prediction layer on the contextual representation to identify the start and end positions of answers, and employ cross-entropy as the loss function. The corresponding loss terms are denoted as $\mathcal{L}_{start}$ and $\mathcal{L}_{end}$, respectively. Similar to previous work in the field, such as S2G (Wu et al., 2021a) and FE2H (Li et al., 2022a), we also inject a special placeholder token "</e>" and use a linear binary classifier on the output of "</e>" to determine whether a sentence is a supporting fact. The classification loss of the supporting facts is denoted as $\mathcal{L}_{sup}$, and we jointly optimize all of these objectives in our model.

$$\mathcal{L}_{reading} = \lambda_3 \mathcal{L}_{type} + \lambda_4(\mathcal{L}_{start} + \mathcal{L}_{end}) + \lambda_5 \mathcal{L}_{sup} \quad (8)$$

Similarly, we set λ_3 and λ_4 and λ_5 all to 1, giving equal importance to each module for multitask learning.

4 Label Smoothing

Label smoothing is a regularization technique that aims to reduce over-fitting in a classifier by modifying the ground truth labels of the training data. In the one-hot setting, the probability of the correct category $q(y|x)$ for a training sample (x, y) is typically defined as 1, while the probabilities of all other categories $q(\neg y|x)$ are defined as 0. The cross-entropy loss function used in this setting is typically defined as follows:

$$\mathcal{L} = -\sum_{k=1}^{K} q(k|x) \log(p(k|x)) \quad (9)$$

where p(k|x) is the probability of the model's prediction for the k-th class. Specifically, label smoothing mixes $q(k|x)$ with a uniform distribution $u(k)$, independent of the training samples, to produce a new distribution $q'(k|x)$.

$$q'(k|x) = (1 - \epsilon)q(k|x) + \epsilon u(k) \quad (10)$$

We denote ϵ as the weight that controls the importance of $q(k|x)$ and $u(k)$ in the resulting distribution. $u(k)$ is construed as $\frac{1}{K}$ of the uniform distribution, where K is the total number of categories. Next, we will explore the role of Label Smoothing in each of the modules in C2FM.

4.1 Label Smoothing in Retrieval Module

Coarse Retrieval We apply Eq.1 to Eq.10. Additionally, SAE (Tu et al., 2020) and S2G (Wu et al., 2021a) both prioritize documents containing answer spans, named gold documents. Therefore, we introduce a answer aware distribution y^{gold} and obtain a new hybird distribution $y_i^{'coarse}$.

$$y_i^{'coarse} = (1 - \epsilon)y_i^{coarse} + \epsilon u(x) + y_i^{gold} \quad (11)$$

where $u(x)$ is a uniform distribution and y^{gold} is defined as follows.

$$y_i^{gold} = \begin{cases} 1 & D_i \text{ contains answer} \\ 0 & D_i \text{ does not contain answer} \end{cases} \quad (12)$$

Fine Retrieval In section 3.1, we described how the loss of fine retrieval is 0 when coarse retrieval fails to retrieve two relevant documents. In addition, we use a pre-trained model to learn both retrieval processes simultaneously. However, when fine retrieval starts training, the model already has some document retrieval capabilities. In this case, applying label smoothing during the fine retrieval

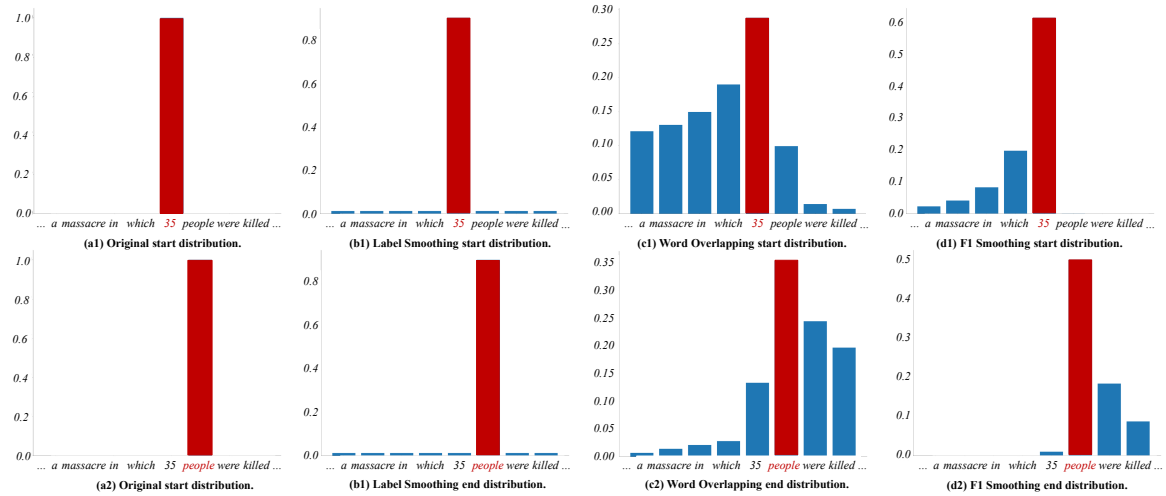

Figure 2: Visualization of original distribution and different label smoothing distributions, including Label Smoothing, Word Overlapping, and F1 Smoothing. The first row shows the distribution of the start token, and the second row shows the distribution of the end token. The gold start and end tokens are highlighted in red.

phase could obscure the goal of model training and potentially hinder the model's performance. Therefore, we choose not to use label smoothing in the fine retrieval phase. For the sake of completeness, we include the results of experiments using label smoothing in the fine retrieval phase in Appendix A.

4.2 Label Smoothing in Reading Module

In multi-task prediction, there are three types of loss: $\mathcal{L}_{type}$, $\mathcal{L}_{sentence}$, and $\mathcal{L}_{span}$. $\mathcal{L}_{type}$ and $\mathcal{L}_{sentence}$ are losses for simple classification tasks, which can be smoothed using normal methods. $\mathcal{L}_{span}$ is the loss for answer extraction, which involves identifying the start and end positions of a span. Due to the specific nature of this task, a different smoothing method may be required to achieve optimal results. Previous research (Zhao et al., 2020a) has explored various label smoothing methods for machine reading comprehension, including normal label smoothing and word overlap smoothing. Motivated by the concept of word overlapping, we propose a more mathematically consistent extension of label smoothing to tasks with F1 scores, named **F1 Smoothing**.

Consider a sample x that contains a context S and an answer a_{gold}. The total length of the context is denoted by L. We use $q_s(t|x)$ to denote the F1 score between a span of arbitrary length starting at position t in S and the ground truth answer a_{gold}. Similarly, $q_e(t|x)$ denotes the F1 score between a span of arbitrary length ending at position t in S and a_{gold}.

$$q_s(t|x) = \sum_{\xi=t}^{L-1} F1\left((t,\xi), a_{gold}\right) \tag{13}$$

$$q_e(t|x) = \sum_{\xi=0}^{t} F1\left((\xi,t), a_{gold}\right) \tag{14}$$

The normalized distributions are noted as $q_s^{'}(t|x)$ and $q_e^{'}(t|x)$, respectively.

$$q_s^{'}(t|x) = \frac{exp(q_s(t|x))}{\sum_{i=0}^{L-1} exp(q_s(i|x))} \tag{15}$$

$$q_e^{'}(t|x) = \frac{exp(q_e(t|x))}{\sum_{i=0}^{L-1} exp(q_e(i|x))} \tag{16}$$

In order to reduce the computational overhead of F1 Smoothing, we present a fast version of the computation in Appendix B. As shown in Figure 2, F1 Smoothing provides a more precise labeling of tokens compared to Word Overlapping, while also decreasing the probability of irrelevant tokens and preventing incorrect guidance of the model during training. This makes F1 Smoothing an effective method for multi-hop question answering tasks.

5 Experiment

5.1 Dataset

We evaluate our approach on the distractor setting of HotpotQA (Yang et al., 2018b), a multi-hop question-answer dataset with 90k training samples, 7.4k validation samples, and 7.4k test samples. Each question in this dataset is provided with

several candidate documents, two of which are relevant to the question, while the others are irrelevant. In addition to this, HotpotQA also provides supporting evidence for each question, encouraging the model to explain the inference path of the multi-hop question-answer. We use the Exact Match (EM) and F1 scores (F1) to evaluate the performance of our approach in terms of related document retrieval, supporting evidence prediction, and answer extraction.

5.2 Implementation Details

Our model is built using the Pre-trained language models (PLMs) provided by HuggingFace's Transformers library (Wolf et al., 2020).

Coarse-to-Fine Retriever We employed the large version of RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2020) as our PLMs and conducted the ablation study on RoBERTa-large (Liu et al., 2019). We used a single RTX3090 GPU, set the number of epochs to 8, and the batch size to 16. For optimizer, we used the AdamW optimizer with a learning rate of 6e-6 and a weight decay of 1e-2.

Multi-Task Reader We utilized the large version of RoBERTa (Liu et al., 2019) and the XXLarge version of DeBERTa (He et al., 2020a) as our PLMs and conducted ablation studies on RoBERTa-large (Liu et al., 2019). For the RoBERTa-large model, we employed an RTX3090 GPU and set the number of epochs to 16 and the batch size to 16. For the DeBERTa-v2-xxlarge model, due to the larger number of parameters, we used an A100 GPU and set the number of epochs to 8 and the batch size to 16. We also utilized the AdamW optimizer with a learning rate of 4e-6 and a weight decay of 1e-2 for optimization.

5.3 Experimental Results

We employ ELECTRA (Clark et al., 2020) as the PLM for the retrieval module and DeBERTa-v2-xxlarge as the PLM for the reading comprehension module. Our model, dubbed C2FM with multiple label smoothing, is tested on the HotpotQA test set with the distractor setting. As shown in Table 1, C2FM with multiple label smoothing achieves an improvement of 0.8% and 0.77% in EM and F1 for answer, and 0.19% and 0.57% in EM and F1 for supporting evidence compared to the C2FM model. Among the label smoothing techniques we experimented with, F1 smoothing yielded the most significant performance improvement and thus we named our model **C2FM with F1 Smoothing**, or C2FM-F1 for short.

We compare the performance of our proposed Coarse-to-Fine Retrieval method, which utilizes ELECTRA as a backbone for training, with three advanced works: SAE, S2G, and FE2H. These methods also employ elaborate selectors to retrieve relevant documents. We evaluate the performance of the document retrieval using the EM and F1 metrics. As shown in Table 2, our Coarse-to-Fine retrieval method outperforms these three strong baselines. Moreover, the performance can be further improved through the use of label smoothing.

In Table 3, we compare the performance of a Multi-task Reader trained with DeBERTa-v2-xxlarge (He et al., 2020a) as the backbone on the documents retrieved by the Coarse-to-Fine Retriever. Our results show that the C2FM model outperforms the strong baseline SAE (Tu et al., 2020) and S2G (Wu et al., 2021b), and the use of label smoothing techniques can further improve model performance. Overall, these results demonstrate the value of the C2FM approach and the potential for further performance improvements through the use of label smoothing.

5.4 Smoothing Analysis

In our study of the role of label smoothing, we used RoBERTa-large (Liu et al., 2019) as the backbone for our model. To ensure the reliability of our experimental results, we conducted multiple runs with different random number seeds (41, 42, 43, and 44) to ensure stability.

Label smoothing contains a hyperparameter ϵ to modify the target label probabilities of a model during training. In order to improve the effectiveness of label smoothing, (Xu et al., 2020) proposed TSLA, a two-stage learning approach that applies label smoothing in the first stage and trains the model in the normal way in the second stage. We propose that Epsilon can also be decayed linearly, similar to the way the learning rate is often decayed. In our experiments, we compared three label smoothing strategies: Constant, TSLA, and Linear Decay. The initial value of Epsilon in our experiments was 0.1, and in the first stage of TSLA, the number of epochs was set to 4. For each epoch in the linear decay strategy, Epsilon was decreased by 0.01.

Coarse Retrieval In our analysis presented in Table 4, we observed that introducing y^{gold} did not

Model	Answer		Supporting	
	EM	F1	EM	F1
Baseline Model (Yang et al., 2018a)	45.60	59.02	20.32	64.49
QFE (Nishida et al., 2019)	53.86	68.06	57.75	84.49
DFGN (Qiu et al., 2019)	56.31	69.69	51.50	81.62
SAE-large (Tu et al., 2020)	66.92	79.62	61.53	86.86
C2F Reader (Shao et al., 2020)	67.98	81.24	60.81	87.63
HGN-large (Fang et al., 2020)	69.22	82.19	62.76	88.47
FE2H on ELECTRA (Li et al., 2022a)	69.54	82.69	64.78	88.71
AMGN+ (Li et al., 2021)	70.53	83.37	63.57	88.83
S2G+EGA (Wu et al., 2021b)	70.92	83.44	63.86	88.68
FE2H on ALBERT (Li et al., 2022a)	71.89	**84.44**	64.98	89.14
C2FM (ours)	71.27	83.57	65.25	88.98
C2FM with F1 Smoothing (ours)	**72.07**	84.34	**65.44**	**89.55**

Table 1: In the distractor setting of the HotpotQA test set, our comparison showed that the C2FM model with various label smoothing methods significantly outperforms the original model. Among the label smoothing methods we evaluated, C2FM with F1 smoothing achieved the best results, surpassing the state-of-the-art performance in the literature. These findings demonstrate the effectiveness of introducing label smoothing in multi-hop question answering tasks.

Model	EM	F1
SAE$_{large}$ (Tu et al., 2020)	91.98	95.76
S2G$_{large}$ (Wu et al., 2021b)	95.77	97.82
FE2H$_{large}$ (Li et al., 2022a)	96.32	98.02
C2FM (ours)	96.50	98.10
w. Label Smoothing	**96.85**	**98.32**

Table 2: Comparison of our coarse-to-fine retriever with previous baselines on HotpotQA dev set. Label smoothing can further enhance model performance.

Model	Answer		Supporting	
	EM	F1	EM	F1
SAE	67.70	80.75	63.30	87.38
S2G	70.80	-	65.70	-
C2FM	71.39	83.84	66.32	89.54
C2FM-F1	71.89	84.65	66.75	90.08

Table 3: Performances of cascade results with label smoothing on the dev set of HotpotQA in the distractor setting.

Setting	F1	EM
Constant	97.94±.04	96.06±.11
w/o $u(x)$	97.91±.09	95.93±.05
w/o y^{gold}	97.88±.08	95.89±.07
TSLA	98.05±.05	96.21±.01
Linear Decay	98.18±.04	96.57±.05

Table 4: Ablation and strategy analysis on Coarse Retrieval Module with Label Smoothing.

Setting	F1	EM
Constant	90.53±.02	66.88±.02
w/o $u(x)$	90.50±.02	66.94±.05
TSLA	90.72±.05	67.42±.05
Linear Decay	90.85±.03	67.63±.04

Table 5: An analysis of the effectiveness of label smoothing for Supporting Evidence Prediction through ablation and strategy evaluation.

significantly improve the performance of the coarse retrieval module. One possible explanation for this result is that the inclusion of y^{gold} may have exacerbated the overconfidence of the model. Our findings on the Constant label smoothing strategy were consistent with (Penha and Hauff, 2021), which showed that it did not significantly improve the retrieval module's performance. However, we found that using the TSLA and Linear Decay strategies effectively stimulated the potential of label smoothing, resulting in improved generalization for the retrieval model.

Supporting Evidence Prediction We evaluated the impact of using different label smoothing strategies on the performance of our model on supporting evidence prediction. As shown in Table 5, regular label smoothing had a negligible effect on the model's performance. On the other hand, the TSLA strategy resulted in an average improvement of 0.22% in EM and 0.48% in F1. The Linear De-

Methods	F1	EM
RoBERTa$_{large}$	69.11±.02	82.21±.03
w. Label Smoothing	69.30±.02	82.56±.09
w. Word Overlapping	69.60±.09	82.68±.13
w. F1 Smoothing	**69.93**±.07	**83.05**±.10

Table 6: Analysis of different label smoothing methods for Answer Span Extraction.

Setting	Accuracy
Constant	99.43±.01
w/o $u(x)$	99.41±.01
TSLA	99.45±.04
Linear Decay	99.44±.02

Table 7: Answer Type Selection results with different smoothing strategies.

cay strategy also yielded positive results, with an average gain of 0.35% in EM and 0.69% in F1. These results suggest that both TSLA and Linear Decay may be effective strategies for improving the performance of supporting evidence prediction through label smoothing.

Answer Span Extraction Table 6 demonstrates the effect of applying different label smoothing techniques on the performance of answer span extraction. Our findings are consistent with previous research (Zhao et al., 2020b), which showed that label smoothing can improve model performance. Specifically, we found that Word Overlapping resulted in an average F1 improvement of 0.3% and an average EM improvement of 0.12%, while F1 Smoothing resulted in an average F1 improvement of 0.63% and an average EM improvement of 0.49%. These results suggest that F1 Smoothing is an effective technique for improving the performance of the reading comprehension module.

Answer Type Selection We implemented label smoothing on the answer type selection task. We evaluated the classification performance using accuracy and the results are shown in Table 7. Despite the relatively simple nature of this task, we achieved a very high accuracy rate. However, we found that the use of label smoothing did not significantly improve the performance of the model.

5.5 Error Analysis

To better understand the role of label smoothing for the overall architecture, we conducted an error analysis following the approach of S2G (Wu et al., 2021b) on our C2FM and C2FM-F1 model. Table 8 shows three types of errors in our model's predictions: incompleteness errors, superfluity errors, and multi-hop reasoning errors. An incompleteness error occurs when the predicted answer span is smaller than the labeled answer span. A superfluity error occurs when the predicted answer span exceeds the labeled answer span. A multi-hop reasoning error occurs when the predicted answer span is offset from the labeled answer span due to errors in the model's multi-hop reasoning. The experimental results indicate that label smoothing was effective in reducing incompleteness, superfluity, and multi-hop reasoning errors by 8.23%, 9.78%, and 10.87%, respectively. Among the three types of errors examined, label smoothing had the greatest impact on reducing Multi-hop Reasoning errors, which decreased by **10.87%**. Overall, these results suggest that label smoothing is a effective technique to consider when training a multi-hop question answering model.

Model	Incomplete	Superfluous	Multi-hop RC
C2FM	729	644	828
C2FM-F1	669	581	738

Table 8: Error analysis on three types of errors: Incompleteness, Superfluity, and Multi-hop Reasoning errors.

6 Conclusion

In this paper, we present C2FM, a simple architecture for the HotpotQA dataset, and systematically analyze the effect of label smoothing on various modules of multi-hop question answering (MHQA). We also propose F1 smoothing, a novel label smoothing technique specifically designed for machine reading comprehension (MRC) tasks. Our experiments on the HotpotQA dataset demonstrate that C2FM with label smoothing outperforms several strong baselines, highlighting the effectiveness of label smoothing in MHQA. However, our results also show that the choice of smoothing strategy is critical for achieving optimal performance.

Overall, our work contributes to a deeper understanding of the role of label smoothing in MHQA and introduces a new label smoothing technique that can be applied to improve the performance of MRC models. We believe our findings will be valuable for researchers and practitioners working on MRC and MHQA tasks and hope they will inspire further research in this area.

Limitations

There are three main limitations to our study: the limited scope of our model architecture, the assumptions we made in proposing F1 Smoothing, and the computational complexity of our experiments.

Limited scope Due to the specificity of our model architecture design, our study is only applicable to the classical multi-hop question answering (MHQA) dataset HotpotQA. Future work will involve designing a more general MHQA model architecture and studying a broader range of MHQA datasets.

Assumptions We proposed F1 Smoothing under the assumption that the span of answers is not unique. However, this assumption may not hold for all datasets, such as SQuAD (Rajpurkar et al., 2018), which may require the development of new smoothing methods.

Computational complexity Finally, our experiments on large versions of pre-trained language models (PLMs) to investigate the effectiveness of our smoothing method are computationally complex. As mentioned in Section 5.2, reproducing our results may require enough computational resources."

Ethics Statement

In conducting this study, we considered the ethical implications of our work and ensured that this work complied with ACL's ethical policies. We used the publicly available HotpotQA dataset for our experiments, which does not raise potential ethical issues.

References

Jan Chorowski and Navdeep Jaitly. 2017. Towards better decoding and language model integration in sequence to sequence models. In *INTERSPEECH*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.

Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. 2020. Towards a better understanding of label smoothing in neural machine translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 212–223, Suzhou, China. Association for Computational Linguistics.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52, Florence, Italy. Association for Computational Linguistics.

Dirk Groeneveld, Tushar Khot, Mausam, and Ashish Sabharwal. 2020. A simple yet strong pipeline for HotpotQA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8839–8845, Online. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020a. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv preprint*, abs/2006.03654.

Xingxin He, Leyuan Fang, Hossein Rabbani, Xiangdong Chen, and Zhimin Liu. 2020b. Retinal optical coherence tomography image classification with label smoothing generative adversarial network. *Neurocomputing*, 405:37–47.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Ronghan Li, Lifang Wang, Shengli Wang, and Zejun Jiang. 2021. Asynchronous multi-grained graph network for interpretable multi-hop reading comprehension. In *IJCAI*, pages 3857–3863.

Xin-Yi Li, Wei-Jun Lei, and Yu-Bin Yang. 2022a. From easy to hard: Two-stage selector and reader for multi-hop question answering. *ArXiv preprint*, abs/2205.11729.

Xin-Yi Li, Weixian Lei, and Yubin Yang. 2022b. From easy to hard: Two-stage selector and reader for multi-hop question answering. *ArXiv preprint*, abs/2205.11729.

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*, abs/1907.11692.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. 2020a. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. PMLR.

Michal Lukasik, Himanshu Jain, Aditya Menon, Seungyeon Kim, Srinadh Bhojanapalli, Felix Yu, and Sanjiv Kumar. 2020b. Semantic label smoothing for sequence to sequence problems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4992–4998, Online. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4696–4705.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy. Association for Computational Linguistics.

Gustavo Penha and Claudia Hauff. 2021. Weakly supervised label smoothing. In *European Conference on Information Retrieval*, pages 334–341. Springer.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, Florence, Italy. Association for Computational Linguistics.

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.

Sougata Saha, Souvik Das, and Rohini Srihari. 2021. Similarity based label smoothing for dialogue generation. *ArXiv preprint*, abs/2107.11481.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Is Graph Structure Necessary for Multi-hop Question Answering? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7187–7192, Online. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Bohong Wu, Zhuosheng Zhang, and Hai Zhao. 2021a. Graph-free multi-hop reading comprehension: A select-to-guide strategy.

Bohong Wu, Zhuosheng Zhang, and Hai Zhao. 2021b. Graph-free multi-hop reading comprehension: A select-to-guide strategy. *ArXiv preprint*, abs/2107.11823.

Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, and Rong Jin. 2020. Towards understanding label smoothing. *ArXiv preprint*, abs/2006.11653.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhenyu Zhao, Shuangzhi Wu, Muyun Yang, Kehai Chen, and Tiejun Zhao. 2020a. Robust machine

reading comprehension by learning soft labels. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2754–2759, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhenyu Zhao, Shuangzhi Wu, Muyun Yang, Kehai Chen, and Tiejun Zhao. 2020b. Robust machine reading comprehension by learning soft labels. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2754–2759, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Enwei Zhu and Jinpeng Li. 2022. Boundary smoothing for named entity recognition. In *ACL*.

A Appendix A

Table 9 presents the results of our experiments on label smoothing for the fine retrieval module. Our findings indicate that the use of label smoothing, regardless of the strategy employed, negatively impacts model performance. Based on these results, we do not recommend implementing label smoothing in this module. This conclusion aligns with our analysis in Section 4.1.

Setting	F1	EM
Constant	97.77±.06	95.91±.08
w/o $u(x)$	97.91±.09	95.93±.05
TSLA	97.79±.04	95.89±.09
Linear Decay	97.78±.02	95.90±.06

Table 9: An analysis of label smoothing on Fine Retrieval Module.

B Appendix B

We use $L_a = e^* - s^* + 1$ and $L_p = e - s + 1$ to denote respectively the length of gold answer and predicted answer. As mentioned in 4.2,

$$q_s(t|x) = \sum_{\xi=t}^{L-1} \text{F1}\left((t,\xi), a_{gold}\right) \tag{17}$$

If $t < s^*$, the distribution is

$$q_s(t|x) = \sum_{\xi=s^*}^{e^*} \frac{2(\xi - s^* + 1)}{L_p + L_a} + \sum_{\xi=e^*+1}^{L-1} \frac{2L_a}{L_p + L_a}, \tag{18}$$

else if $s^* \leq t \leq e^*$, we have the following distribution

$$q_s(t|x) = \sum_{\xi=s}^{e^*} \frac{2L_p}{L_p + L_a} + \sum_{\xi=e^*+1}^{L-1} \frac{2(e^* - s + 1)}{L_p + L_a}. \tag{19}$$

In equation 18 and 19, $L_p = e - i + 1$.

We can get $q_e(t|x)$ similarly. If $t > e^*$,

$$q_e(t|x) = \sum_{\xi=s^*}^{e^*} \frac{2(e^* - \xi + 1)}{L_p + L_a} + \sum_{\xi=0}^{s^*-1} \frac{2L_a}{L_p + L_a}, \tag{20}$$

else if $s^* \leq t \leq e^*$,

$$q_e(t|x) = \sum_{\xi=s^*}^{e} \frac{2L_p}{L_p + L_a} + \sum_{\xi=0}^{s^*-1} \frac{2(e - s^* + 1)}{L_p + L_a}. \tag{21}$$

In equation 20 and 21, $L_p = i - s + 1$.